

9.0

Managed File Transfer

IBM

附註

使用本資訊及其支援的產品之前，請先閱讀第 821 頁的『[注意事項](#)』中的資訊。

除非新版中另有指示，否則此版本適用於 IBM® MQ 9.0.0 版及所有後續版本與修訂版。

當您將資訊傳送至 IBM 時，您授與 IBM 非專屬權利，以任何其認為適當的方式使用或散佈資訊，而無需對您負責。

© Copyright International Business Machines Corporation 2007, 2023.

目錄

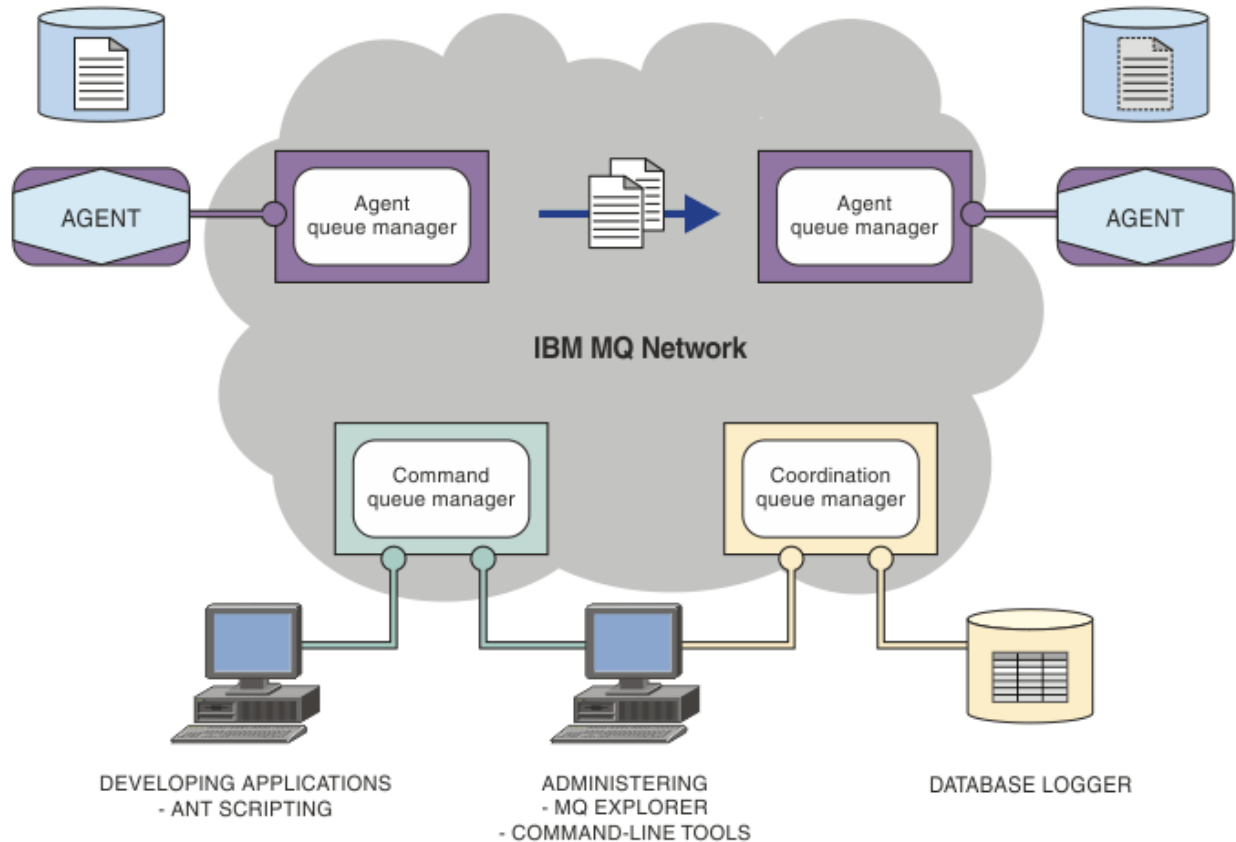
Managed File Transfer.....	5
MFT 概觀.....	6
MFT 簡介.....	7
MFT 如何與 IBM MQ 配合運作?	8
MFT 拓撲概觀.....	9
MFT REST API 概觀.....	10
IBM MQ 9.0 中 MFT 的新增功能及變更功能?.....	10
MFT 的移轉考量.....	12
MFT 實務練習概觀.....	25
MFT 一般拓撲.....	25
配置基本伺服器.....	28
保護 MFT 安全.....	35
MFT 及 IBM MQ 連線鑑別.....	36
MFT 沙盤推演.....	39
配置 MFT 的 SSL 或 TLS 加密.....	44
在用戶端模式下使用通道鑑別連接至佇列管理程式.....	45
在 Connect:Direct 橋接器代理程式與 Connect:Direct 節點之間配置 SSL 或 TLS.....	46
配置 MFT.....	48
多平台上的 MFT 配置選項.....	49
z/OS 上的 MFT 配置選項.....	50
配置 Redistributable Managed File Transfer Agent.....	50
建立 MFT 代理程式或日誌程式指令資料集.....	54
配置 Managed File Transfer for z/OS.....	55
在 IBM i 上配置 MFT.....	79
初次使用時配置 MFT.....	80
配置 MFT 日誌程式.....	89
配置 Connect:Direct 橋接器.....	118
使用 MSCS 配置 MFT 代理程式.....	126
管理 MFT.....	128
啟動 MFT 代理程式.....	128
啟動新的檔案傳送.....	133
建立排定的檔案傳送.....	135
處理擱置中檔案傳送.....	136
觸發檔案傳送.....	137
監視正在進行的檔案傳送.....	138
在傳送日誌中檢視檔案傳送的狀態.....	139
監視 MFT 資源.....	141
使用檔案傳送範本.....	162
將資料從檔案傳送至訊息.....	164
將資料從訊息傳送至檔案.....	171
列出 MFT 代理程式.....	176
停止 MFT 代理程式.....	177
通訊協定橋接器.....	178
Connect:Direct 橋接器.....	192
從 IBM Integration Bus 使用 MFT.....	205
MFT 回復及重新啟動.....	206
回復中檔案傳送的逾時選項.....	206
為 MFT 開發應用程式.....	209
指定要使用 MFT 執行的程式.....	209
搭配使用 Apache Ant 與 MFT.....	211
利用使用者結束程式自訂 MFT.....	214
將訊息放置在代理程式指令佇列上以控制 MFT.....	227

疑難排解 MFT.....	227
MFT 一般疑難排解.....	227
Connect:Direct 橋接器疑難排解.....	289
MFT 參照.....	294
MFT 安全參照.....	294
哪些 MFT 指令和程序連接至哪個佇列管理程式.....	307
MFT 指令.....	313
MFT 配置參照.....	452
MFT 管理參照.....	525
MFT 開發應用程式參照.....	728
MFT 診斷訊息.....	819
注意事項.....	821
程式設計介面資訊.....	822
商標.....	822

Managed File Transfer

不論檔案大小或使用的作業系統為何，Managed File Transfer 皆會以受管理且可審核的方式在系統之間傳送檔案。

您可以使用 Managed File Transfer 來建置自訂、可擴充及自動化的解決方案，讓您能夠管理、信任及維護檔案傳送的安全。Managed File Transfer 刪除了高成本的備用項目、降低維護成本，並使現有的 IT 投資獲得最大的效益。



此圖表顯示簡易 Managed File Transfer 拓撲。在 IBM MQ 網路中有兩個代理程式，各自連接到專屬的代理程式佇列管理程式。檔案從圖表一側的代理程式，經由 IBM MQ 網路，傳送至圖表另一側的代理程式。此外，在 IBM MQ 網路中，還有協調佇列管理程式及指令佇列管理程式。應用程式及工具會連接至這些佇列管理程式，以在 IBM MQ 網路中配置、管理、操作及記載 Managed File Transfer 活動。

視您的作業系統及整體設定而定，Managed File Transfer 可安裝為四個不同的選項。這些選項是 Managed File Transfer Agent、Managed File Transfer Logger、Managed File Transfer Service 或 Managed File Transfer Tools。如需相關資訊，請參閱 [Managed File Transfer 產品選項](#)。

您可以使用 Managed File Transfer 來執行下列作業：

- 建立 Managed File Transfer
 - 在 Linux® 或 Windows 平台上，從 IBM MQ Explorer 建立新的檔案傳送。
 - 從所有受支援平台的指令行建立新的檔案傳送。
 - 將檔案傳送功能整合到 Apache Ant 工具中。
 - 撰寫應用程式，以透過將訊息放置在代理程式指令佇列上來對 Managed File Transfer 進行控制。

- 將檔案傳送排定於稍後進行。您也可以根據一系列檔案系統事件（例如建立新檔案）來觸發排定的檔案傳送。
- 持續監視資源，例如目錄，然後當該資源的內容符合一些預先定義的條件時，啟動作業。此作業可以是檔案傳送、Ant Script 或 JCL 工作。
- 與 IBM MQ 佇列來回傳送檔案。
- 與 FTP、FTPS 或 SFTP 伺服器來回傳送檔案。
- 與 Connect:Direct 節點來回傳送檔案。
- 傳送文字及二進位檔。會在來源與目的地系統的字碼頁及行尾使用慣例之間自動轉換文字檔。
- 可使用以 Secure Socket Layer (SSL) 基礎連線的業界標準來維護傳送安全。
- 檢視進行中的傳送，並記載網路中所有傳送的相關資訊
 - 在 Linux 或 Windows 平台上，從 IBM MQ Explorer 檢視進行中傳送的狀態。
 - 在 Linux 或 Windows 平台上使用 IBM MQ Explorer 來檢查已完成傳送的狀態。
 - 使用 Managed File Transfer 資料庫日誌程式特性，將日誌訊息儲存至 Db2 或 Oracle 資料庫。

Managed File Transfer 建置在 IBM MQ 上，其提供應用程式之間保證唯一一次的訊息遞送。您可以充分運用 IBM MQ 的各種特性。例如，您可以使用通道壓縮來壓縮您在代理程式之間透過 IBM MQ 通道傳送的資料，以及使用 SSL 通道來維護在代理程式之間傳送的資料的安全。可靠的檔案傳送，並可容忍執行檔案傳送的基礎架構失敗。如果您發生網路中斷，當連線功能還原時，檔案傳送會從它停止的位置重新啟動。

合併檔案傳送與現有的 IBM MQ 網路，您可以避免浪費維護兩個不同基礎架構所需的資源。如果您還不是 IBM MQ 客戶，則可以透過建立 IBM MQ 網路來支援 Managed File Transfer，以建置未來 SOA 實作的骨幹。如果您已經是 IBM MQ 客戶，Managed File Transfer 可以充分運用您現有的 IBM MQ 基礎架構，包括 IBM MQ Internet Pass-Thru 及 IBM Integration Bus。

整合 Managed File Transfer 與一些其他 IBM 產品：

IBM Integration Bus

IBM Integration Bus 流程的一部分，由 Managed File Transfer 傳輸程序檔案。如需相關資訊，請參閱第 205 頁的『[從 IBM Integration Bus 使用 MFT](#)』。

IBM Sterling Connect:Direct

使用 Managed File Transfer Connect:Direct 橋接器，與現有的 Connect:Direct 網路來回傳送檔案。如需相關資訊，請參閱第 192 頁的『[Connect:Direct 橋接器](#)』。

IBM Tivoli Composite Application Manager

IBM Tivoli Composite Application Manager 提供的代理程式可讓您用來監視已發佈至協調佇列管理程式的資訊。

相關概念

第 9 頁的『[MFT 拓撲概觀](#)』

有關 Managed File Transfer 代理程式如何連接 IBM MQ 網路中的協調佇列管理程式的概觀。

第 8 頁的『[MFT 如何與 IBM MQ 配合運作?](#)』

Managed File Transfer 以數種方式與 IBM MQ 進行互動。

相關資訊

[Managed File Transfer 產品選項](#)

MFT 概觀

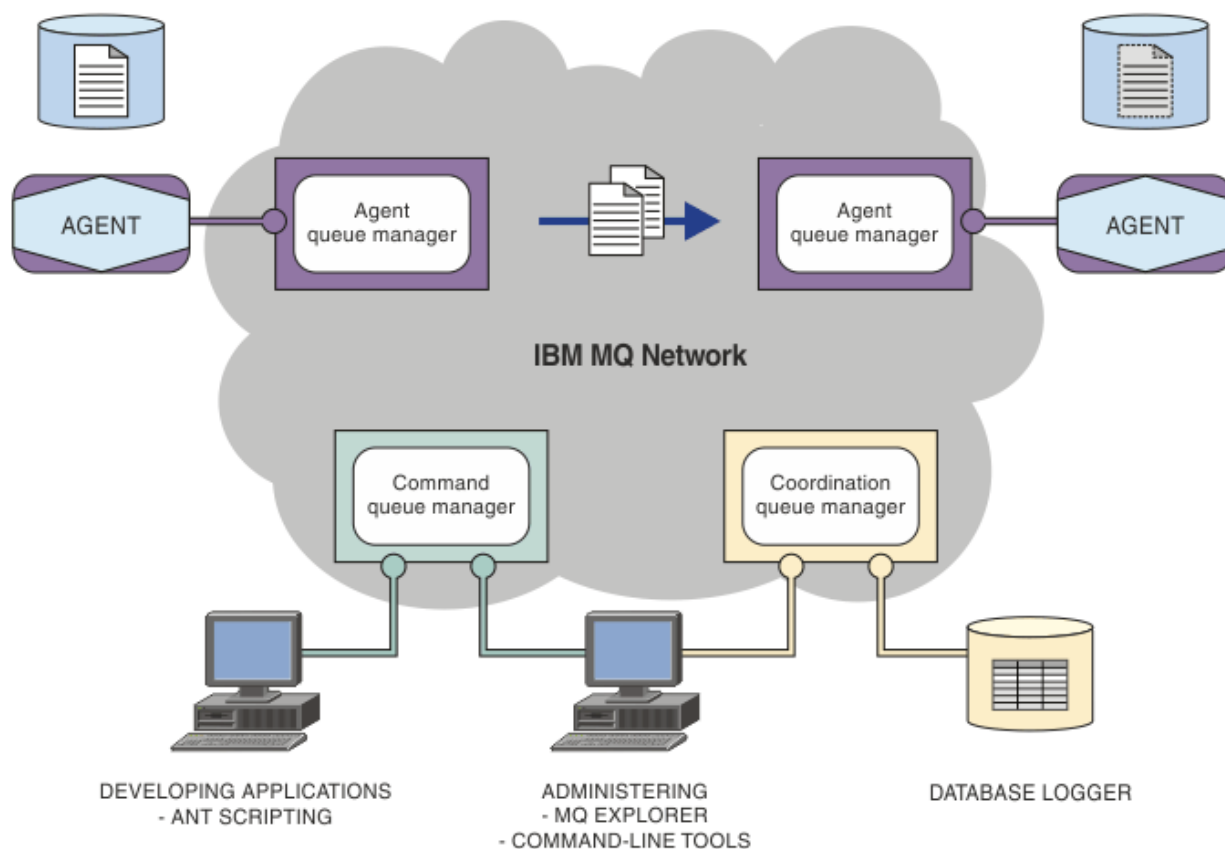
本節提供 Managed File Transfer 入門的介紹資訊。

- [第 5 頁的『Managed File Transfer』](#)
- [Managed File Transfer 產品選項](#)
- [第 9 頁的『MFT 拓撲概觀』](#)
- [第 10 頁的『IBM MQ 9.0 中 MFT 的新增功能及變更功能?』](#)

Managed File Transfer

不論檔案大小或使用的作業系統為何，Managed File Transfer 皆會以受管理且可審核的方式在系統之間傳送檔案。

您可以使用 Managed File Transfer 來建置自訂、可擴充及自動化的解決方案，讓您能夠管理、信任及維護檔案傳送的安全。Managed File Transfer 刪除了高成本的備用項目、降低維護成本，並使現有的 IT 投資獲得最大的效益。



此圖表顯示簡易 Managed File Transfer 拓撲。在 IBM MQ 網路中有兩個代理程式，各自連接到專屬的代理程式佇列管理程式。檔案從圖表一側的代理程式，經由 IBM MQ 網路，傳送至圖表另一側的代理程式。此外，在 IBM MQ 網路中，還有協調佇列管理程式及指令佇列管理程式。應用程式及工具會連接至這些佇列管理程式，以在 IBM MQ 網路中配置、管理、操作及記載 Managed File Transfer 活動。

視您的作業系統及整體設定而定，Managed File Transfer 可安裝為四個不同的選項。這些選項是 Managed File Transfer Agent、Managed File Transfer Logger、Managed File Transfer Service 或 Managed File Transfer Tools。如需相關資訊，請參閱 [Managed File Transfer 產品選項](#)。

您可以使用 Managed File Transfer 來執行下列作業：

- 建立 Managed File Transfer
 - 在 Linux 或 Windows 平台上，從 IBM MQ Explorer 建立新的檔案傳送。
 - 從所有受支援平台的指令行建立新的檔案傳送。
 - 將檔案傳送功能整合到 Apache Ant 工具中。
 - 撰寫應用程式，以透過將訊息放置在代理程式指令佇列上來對 Managed File Transfer 進行控制。
 - 將檔案傳送排定於稍後進行。您也可以根據一系列檔案系統事件（例如建立新檔案）來觸發排定的檔案傳送。
 - 持續監視資源，例如目錄，然後當該資源的內容符合一些預先定義的條件時，啟動作業。此作業可以是檔案傳送、Ant Script 或 JCL 工作。

- 與 IBM MQ 佇列來回傳送檔案。
- 與 FTP、FTPS 或 SFTP 伺服器來回傳送檔案。
- 與 Connect:Direct 節點來回傳送檔案。
- 傳送文字及二進位檔。會在來源與目的地系統的字碼頁及行尾使用慣例之間自動轉換文字檔。
- 可使用以 Secure Socket Layer (SSL) 基礎連線的業界標準來維護傳送安全。
- 檢視進行中的傳送，並記載網路中所有傳送的相關資訊
 - 在 Linux 或 Windows 平台上，從 IBM MQ Explorer 檢視進行中傳送的狀態。
 - 在 Linux 或 Windows 平台上使用 IBM MQ Explorer 來檢查已完成傳送的狀態。
 - 使用 Managed File Transfer 資料庫日誌程式特性，將日誌訊息儲存至 Db2 或 Oracle 資料庫。

Managed File Transfer 建置在 IBM MQ 上，其提供應用程式之間保證唯一一次的訊息遞送。您可以充分運用 IBM MQ 的各種特性。例如，您可以使用通道壓縮來壓縮您在代理程式之間透過 IBM MQ 通道傳送的資料，以及使用 SSL 通道來維護在代理程式之間傳送的資料的安全。可靠的檔案傳送，並可容忍執行檔案傳送的基礎架構失敗。如果您發生網路中斷，當連線功能還原時，檔案傳送會從它停止的位置重新啟動。

合併檔案傳送與現有的 IBM MQ 網路，您可以避免浪費維護兩個不同基礎架構所需的資源。如果您還不是 IBM MQ 客戶，則可以透過建立 IBM MQ 網路來支援 Managed File Transfer，以建置未來 SOA 實作的骨幹。如果您已經是 IBM MQ 客戶，Managed File Transfer 可以充分運用您現有的 IBM MQ 基礎架構，包括 IBM MQ Internet Pass-Thru 及 IBM Integration Bus。

整合 Managed File Transfer 與一些其他 IBM 產品：

IBM Integration Bus

IBM Integration Bus 流程的一部分，由 Managed File Transfer 傳輸程序檔案。如需相關資訊，請參閱第 205 頁的『從 IBM Integration Bus 使用 MFT』。

IBM Sterling Connect:Direct

使用 Managed File Transfer Connect:Direct 橋接器，與現有的 Connect:Direct 網路來回傳送檔案。如需相關資訊，請參閱第 192 頁的『Connect:Direct 橋接器』。

IBM Tivoli Composite Application Manager

IBM Tivoli Composite Application Manager 提供的代理程式可讓您用來監視已發佈至協調佇列管理程式的資訊。

相關概念

第 9 頁的『MFT 拓撲概觀』

有關 Managed File Transfer 代理程式如何連接 IBM MQ 網路中的協調佇列管理程式的概觀。

第 8 頁的『MFT 如何與 IBM MQ 配合運作？』

Managed File Transfer 以數種方式與 IBM MQ 進行互動。

相關資訊

Managed File Transfer [產品選項](#)

MFT 如何與 IBM MQ 配合運作？

Managed File Transfer 以數種方式與 IBM MQ 進行互動。

- Managed File Transfer 透過以下方式在代理程式程序之間傳送檔案：將每個檔案分割為一則以上的訊息，然後透過 IBM MQ 網路傳輸訊息。
- 代理程式程序使用非持續性訊息來移動檔案資料，以最大程度降低對 IBM MQ 日誌的影響。透過彼此間進行通訊，代理程式程序可以限定包含檔案資料的訊息流。如此可避免包含檔案資料的訊息聚集在 IBM MQ 傳輸佇列，並確保如果有任何非持續性訊息未遞送，則再次傳送檔案資料。
- Managed File Transfer 代理程式使用數個 IBM MQ 佇列。如需相關資訊，請參閱第 523 頁的『MFT 系統佇列及系統主題』。
- 雖然其中部分佇列嚴格限於內部使用，但代理程式可以接受傳送至它可讀取的特定佇列，且為特殊格式化指令訊息形式的要求。指令行指令及「IBM MQ Explorer」外掛程式，都可以將 IBM MQ 訊息傳送至代理

程式，以指示代理程式執行所需的動作。您可以撰寫以這種方式與代理程式進行互動的 IBM MQ 應用程式。如需相關資訊，請參閱第 227 頁的『將訊息放置在代理程式指令佇列上以控制 MFT』。

- Managed File Transfer 代理程式會將其傳送狀態、進度及結果的相關資訊，傳送至已指定為協調佇列管理程式的 MQ 佇列管理程式。此類資訊由協調佇列管理程式發佈，可由想要監視傳送進度或記錄已執行傳送的應用程式訂閱。指令行指令及「IBM MQ Explorer」外掛程式，都可以使用所發佈的資訊。您可以撰寫使用這項資訊的 IBM MQ 應用程式。如需向其發佈此類資訊的主題的相關資訊，請參閱第 521 頁的『SYSTEM.FTE 主題』。
- Managed File Transfer 的主要元件可利用 IBM MQ 佇列管理程式的功能，來儲存及轉遞訊息。這意味著如果發生停電，基礎架構中未受影響的部分可以繼續傳送檔案。此情況適用於協調佇列管理程式，其中結合使用儲存及轉遞與可延續訂閱可讓協調佇列管理程式容忍無法使用情況，且不會遺失已執行檔案傳送的主要資訊。

MFT 拓撲概觀

有關 Managed File Transfer 代理程式如何連接 IBM MQ 網路中的協調佇列管理程式的概觀。

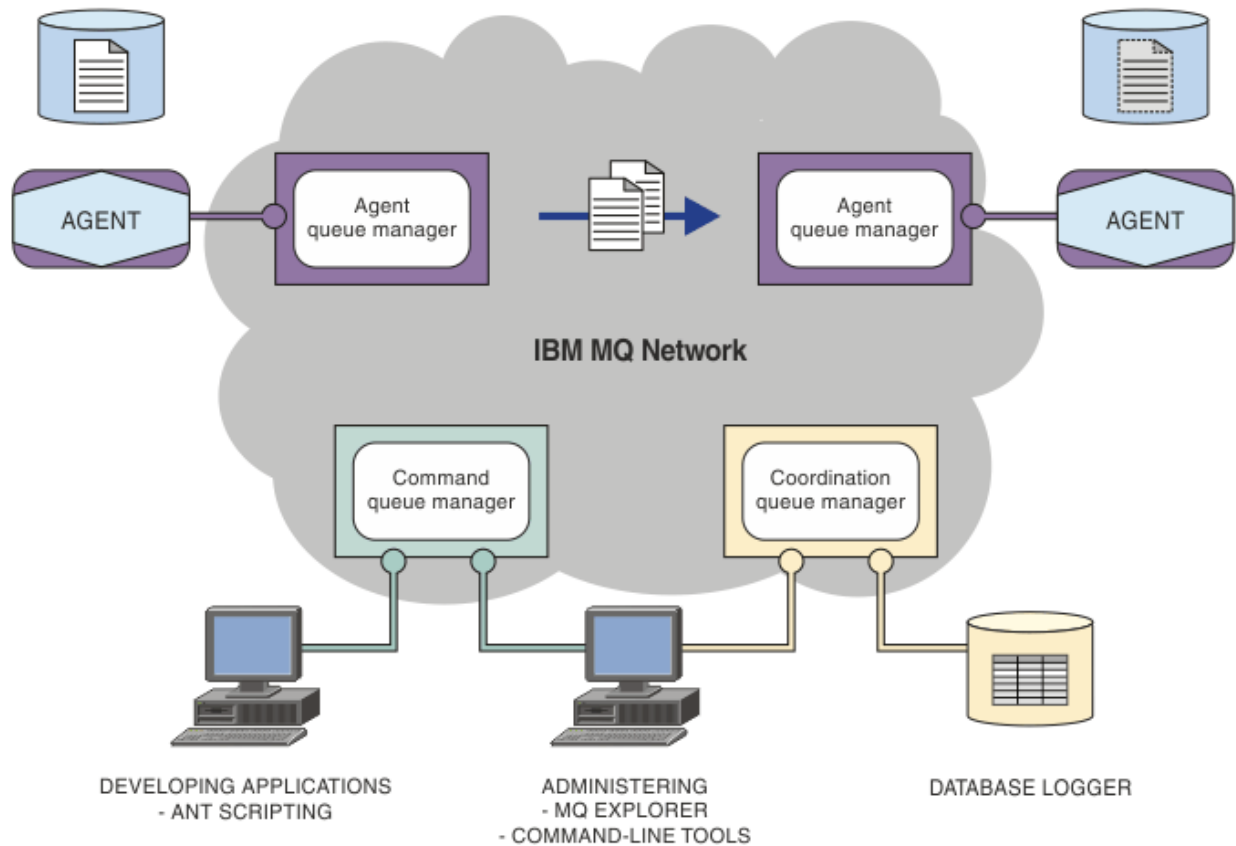
Managed File Transfer 代理程式會傳送及接收已傳送的檔案。每一個代理程式在其關聯的佇列管理程式上都有其專屬的一組佇列，且代理程式是以連結或用戶端模式附加至其佇列管理程式。代理程式也可以使用協調佇列管理程式作為其佇列管理程式。

協調佇列管理程式會播送審核及檔案傳送資訊。協調佇列管理程式代表收集代理程式、傳送狀態及傳送審核資訊的單一點。進行傳送時，不一定要有協調佇列管理程式。如果協調佇列管理程式暫時變成無法使用，傳送會像平常一樣繼續進行。審核及狀態訊息會儲存在代理程式佇列管理程式中，直到協調佇列管理程式變成可用，然後就可以正常地進行處理。

代理程式會向協調佇列管理程式登錄，並將其詳細資料發佈給該佇列管理程式。Managed File Transfer 外掛程式會使用此代理程式資訊，開始從「IBM MQ Explorer」進行傳送。這些指令也會使用在協調佇列管理程式上收集的代理程式資訊，以顯示代理程式資訊及代理程式狀態。

傳送狀態及傳送審核資訊會發佈在協調佇列管理程式上。Managed File Transfer 外掛程式使用傳送狀態及傳送審核資訊，以監視從「IBM MQ Explorer」傳送的進度。可保留儲存在協調佇列管理程式上的傳送審核資訊，以提供審核性。

指令佇列管理程式可以用來連接至 IBM MQ 網路，這也是您發出 Managed File Transfer 指令時所連接的佇列管理程式。



相關概念

第 5 頁的『[Managed File Transfer](#)』

不論檔案大小或使用的作業系統為何，Managed File Transfer 皆會以受管理且可審核的方式在系統之間傳送檔案。

第 8 頁的『[MFT 如何與 IBM MQ 配合運作?](#)』

Managed File Transfer 以數種方式與 IBM MQ 進行互動。

第 25 頁的『[MFT 實務練習概觀](#)』

這個小節列出一般 Managed File Transfer 拓撲，以及一個設定系統和傳送測試訊息的實務練習。

V 9.0.5 MFT REST API 概觀

Managed File Transfer 的 REST API 加強功能概觀。

從 IBM MQ 9.0.5 起，REST API 新增對特定 Managed File Transfer 指令的支援，其中包括列出傳送及檔案傳送代理程式相關詳細資料。

如需相關資訊，請參閱 [REST API 的進一步加強功能](#)。

IBM MQ 9.0 中 MFT 的新增功能及變更功能?

鏈結至 IBM MQ 9.0 中主要新增及已變更 Managed File Transfer 功能的相關資訊。

IBM MQ 9.0.0

> V 9.0.0

- [> V 9.0.0](#) [Managed File Transfer 通訊協定橋接器代理程式記載加強功能](#)
- [> V 9.0.0](#) [移除 Managed File Transfer Web 閘道支援](#)

IBM MQ 9.0.x Continuous Delivery 版本

CD

- Windows V 9.0.1 Linux Managed File Transfer 對已停止傳送的可配置回復逾時
- Windows V 9.0.1 Linux Managed File Transfer Agent 可轉散發套件
- Windows V 9.0.1 Linux 對遠端 Managed File Transfer 代理程式的 **DISPLAY CHSTATUS** 輸出值的變更
- Windows V 9.0.1 Linux 對 Managed File Transfer 中檔案傳送之訊息的變更
- V 9.0.1 新的 MFT 代理程式內容 `additionalWildcardSandboxChecking`
- Windows V 9.0.2 Linux 已改進對 IBM MQ Explorer MFT 外掛程式所建立訂閱的命名
- Windows V 9.0.2 Linux 改良在前次報告狀態為 UNKNOWN 時 MFT 代理程式的相關資訊
- V 9.0.2 z/OS MQ Adv. VUE 新的代理程式內容 `adminGroup`，與 z/OS 上的 MFT 代理程式搭配使用
- Windows z/OS V 9.0.3 Linux Managed File Transfer 資源監視器事件記載
- z/OS 產品 ID (PID) 記錄的變更
- z/OS V 9.0.3 MQ Adv. VUE 代理程式與 IBM MQ Advanced for z/OS, Value Unit Edition 的遠端連線
- Windows z/OS Linux Managed File Transfer `fteCleanAgent` 指令的行為變更
- Windows Linux 已更正 Managed File Transfer 安裝內容中 `messagePublicationFormat=mixed` 的行為
- Windows z/OS V 9.0.3 Linux 對 Managed File Transfer 啟用 MQCSP 鑑別
- Windows V 9.0.4 Linux AIX 以用戶端模式執行 Managed File Transfer Logger 的能力
- V 9.0.5 Multi 第 10 頁的『MFT REST API 概觀』

IBM MQ 9.0.0.x Long Term Support

LTS

- V 9.0.0.1 `fteMigrateAgent` 指令變更
- V 9.0.0.1 新的 MFT 代理程式內容 `addCommandPathToSandbox`
- V 9.0.0.1 新的 MFT 代理程式內容 `additionalWildcardSandboxChecking`
- z/OS V 9.0.0.1 在 z/OS 上與 MFT 代理程式搭配使用的新一代代理程式內容 `adminGroup`
- V 9.0.0.2 Managed File Transfer `fteCleanAgent` 指令的行為變更
- V 9.0.0.2 在 Managed File Transfer 中啟用 MQCSP 鑑別
- V 9.0.0.2 Managed File Transfer `installation.properties` 檔案的新 `messagePublicationFormat` 內容
- V 9.0.0.2 針對訊息轉為檔案傳送而還原的傳送 ID 與 `groupId` 屬性值之比較檢查
- V 9.0.5 在 `fteSetAgentTraceLevel` 及 `fteSetLoggerTracelevel` 指令中使用以冒號區隔的清單之能力

相關資訊

V 9.0.0 IBM MQ 9.0.0 的新增功能與變更內容

CD IBM MQ 9.0.x Continuous Delivery 中的新增功能及變更內容

LTS IBM MQ 9.0.x 長期支援中的變更內容

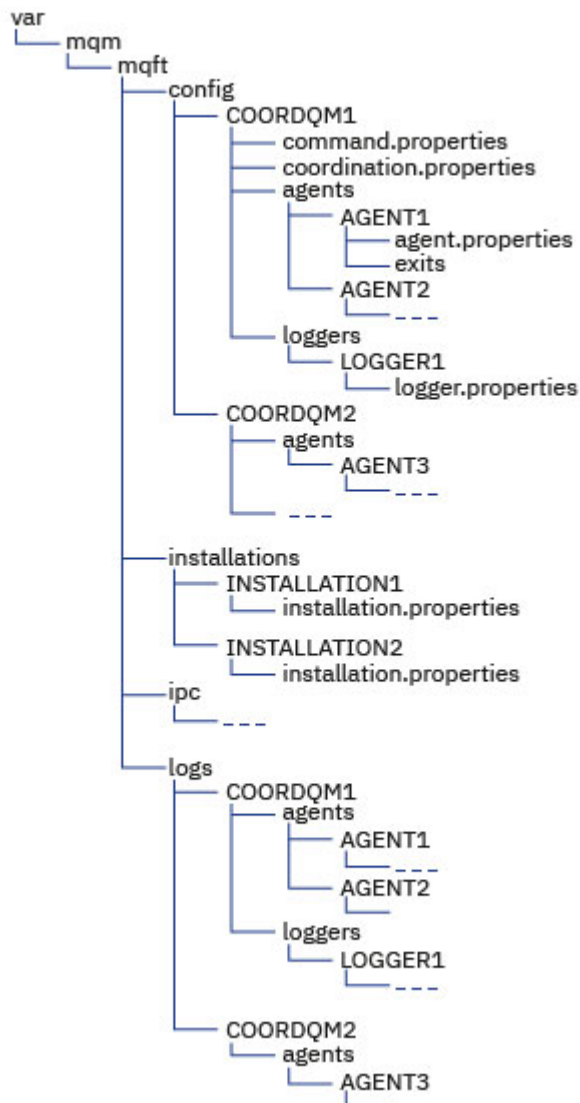
MFT 的移轉考量

如果您計劃移轉 Managed File Transfer，請檢閱下列資訊。

在 MFT 中安裝之後直接配置佈置

此圖顯示安裝之後直接在 IBM MQ 中的配置佈置。

IBM MQ Managed File Transfer configuration



預設配置目錄:

資訊分割成四個個別子目錄: config、installations、ipc 及 logs。

預設產品根目錄 (MQ_DATA_PATH) 如下:

- UNIX 系統: /var/mqm

- Linux 系統: /var/mqm
- Windows: 配置目錄的位置視主要 IBM MQ 安裝的位置而定。主要安裝的預設位置如下所示:
 - 32 位元: C:\程式檔案 (x86) \IBM\WebSphere MQ
 - 64 位元: C:\Program Files\IBM\MQ

配置子目錄如下:

- `MQ_DATA_PATH/mqft/config` 目錄包含對 Managed File Transfer 程序而言是唯讀的配置部分。例如, `agent.properties` 及 `command.properties`。
- `MQ_DATA_PATH/mqft/installations` 目錄包含每一個安裝的配置資訊。此目錄的內容與 `wmqfte.properties` 檔的內容相等。
- `MQ_DATA_PATH/mqft/ipc` 目錄包含內部使用供 Managed File Transfer 元件相互通訊的 IPC 資源。僅適用於 UNIX 及 Linux 系統。
- `MQ_DATA_PATH/mqft/logs` 目錄包含 Managed File Transfer 程序所寫入的配置部分。例如, 追蹤資訊及日誌檔。

installation.properties 檔案

Linux **UNIX** 在 UNIX and Linux 系統上, 預設位置是 `MQ_DATA_PATH/mqft/installations/installation_name`

Windows 在 Windows 上, 預設位置是 `MQ_DATA_PATH\mqft\installations\installation_name`

logger.properties 檔案

此檔案包含獨立式檔案日誌程式、獨立式資料庫日誌程式及 Java EE 資料庫日誌程式的內容資訊。預設位置是 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name`。

安全變更

對於 IBM WebSphere MQ 7.5 或更新版本, 只有身分為管理者 (mqm 群組的成員) 的使用者才能執行下列 **fte** 指令清單:

- [第 321 頁的『fteChangeDefaultConfigurationOptions』](#)
- [第 326 頁的『fteCreateAgent \(建立 MFT 代理程式\)』](#)
- [第 330 頁的『fteCreateBridgeAgent \(建立及配置 MFT 通訊協定橋接器代理程式\)』](#)
- [第 339 頁的『fteCreateCDAgent \(建立 Connect:Direct 橋接器代理程式\)』](#)
- [第 343 頁的『fteCreate 日誌程式 \(建立 MFT 檔案或資料庫日誌程式\)』](#)
- [第 389 頁的『fteDeleteAgent』](#)
- [第 391 頁的『fteDeleteLogger』](#)
- [第 407 頁的『fteMigrate 代理程式: 將 FTE 7.0 代理程式移轉至 MQ 7.5 或更新版本』](#)
- [第 409 頁的『fteMigrateConfigurationOptions: 將 FTE 7.0 配置移轉至 MQ 7.5 或更新版本』](#)
- [第 411 頁的『fteMigrate 日誌程式: 將 FTE 7.0 資料庫日誌程式移轉至 MQ 7.5 或更新版本』](#)
- [第 413 頁的『fteModify 代理程式 \(以 Windows 服務方式執行 MFT 代理程式\)』](#)
- [第 415 頁的『fteModify 日誌程式 \(以 Windows 服務方式執行 MFT 日誌程式\)』](#)
- [第 432 頁的『fteSetupCommands: 建立 MFT command.properties 檔』](#)
- [第 433 頁的『fteSetupCoordination』](#)

Multi 在 Multiplatforms 上使用 IBM WebSphere MQ 7.5 或更新版本時, 只有執行代理程式處理程序的使用者可以執行 **fteSetAgentTraceLevel** 指令。

V 9.0.2 **z/OS** 若為 z/OS, 則從 Continuous Delivery 中的 IBM MQ 9.0.2 開始, 下列使用者 ID 或成員可以執行 **fteSetAgentTraceLevel** 指令:

- 以其身分執行代理程式程序的使用者 ID。
- 代理程式內容 **adminGroup** 所指定之群組的成員。

V 9.0.0.1 這也適用於 IBM MQ 9.0.0 Fix Pack 1 中的 Long Term Support 版本。

V 9.0.2 **z/OS** 若為 z/OS，則從 Continuous Delivery 中的 IBM MQ 9.0.2 開始，如果執行 **fteShowAgentDetails** 指令以顯示本端代理程式的相關資訊，則下列使用者 ID 或成員可以執行此指令：

- 以其身分執行代理程式程序的使用者 ID。
- 代理程式內容 **adminGroup** 所指定之群組的成員。

V 9.0.0.1 這也適用於 IBM MQ 9.0.0 Fix Pack 1 中的 Long Term Support 版本。

如需相關資訊，請參閱第 463 頁的『[MFT agent.properties 檔案](#)』中的 **adminGroup** 內容。

IBM MQ 8.0 中的安全變更

如果您在 IBM WebSphere MQ 7.0 上執行 Managed File Transfer，並移轉至 IBM MQ 8.0，則 MQMFTCredentials.xml 檔中的使用者 ID 資訊會傳遞至佇列管理程式，但不會處理。

這是因為僅在 IBM MQ 8.0 中才支援傳遞使用者 ID 和密碼資訊。

commandPath 及代理程式沙盤推演

對於 IBM MQ 8.0 以及更新版本，如果代理程式配置了代理程式沙盤推演，且已設定代理程式內容 **commandPath**，則當代理程式啟動時，**commandPath** 指定的目錄會自動新增至被拒絕路徑。如果在代理程式上設定了未配置代理程式沙盤推演的 **commandPath** 內容，則會自動設定新的沙盤推演，而且當代理程式啟動時，**commandPath** 指定的目錄會新增至被拒絕目錄。

如果在代理程式上設定了未配置代理程式沙盤推演的 **commandPath** 內容，則會自動設定新的沙盤推演，而且當代理程式啟動時，**commandPath** 指定的目錄會新增至被拒絕目錄。

如需 **commandPath** 內容的相關資訊，請參閱第 306 頁的『[commandPath MFT 內容](#)』及第 463 頁的『[MFT agent.properties 檔案](#)』。

commandPath 及使用者沙盤推演

對於 IBM MQ 8.0 以及更新版本，如果代理程式已配置一個以上的使用者沙盤推演，且已設定代理程式內容 **commandPath**，則當代理程式啟動時，**commandPath** 指定的目錄（以及其所有子目錄）會自動新增為每一個使用者沙盤推演的 **<read>** 及 **<write>** 元素的 **<exclude>** 元素。

如需 **commandPath** 內容的相關資訊，請參閱第 306 頁的『[commandPath MFT 內容](#)』及第 463 頁的『[MFT agent.properties 檔案](#)』。

將作為 Windows 服務執行的 MFT 代理程式從 IBM WebSphere MQ 7.5 移轉至 IBM MQ 9.0

在 IBM WebSphere MQ 7.5、IBM MQ 8.0 和 IBM MQ 9.0 之間，Windows 平台上的預設 IBM MQ 安裝路徑已變更。

如果佇列管理程式正在從 IBM WebSphere MQ 7.5 移轉至 IBM MQ 8.0 或 IBM MQ 9.0，則必須重新配置與佇列管理程式在同一系統上執行的所有應用程式，以從新的安裝位置載入 IBM MQ 程式庫。這包括以 Windows 服務方式執行的任何 IBM MQ Managed File Transfer 代理程式。

執行下列程序：

1. 停止與 IBM WebSphere MQ 7.5 佇列管理程式相關聯的代理程式。如需相關資訊，請參閱[停止 MFT 代理程式](#)。例如，發出下列指令：

```
fteStopAgent <agent_name>
```

2. 修改代理程式以移除代理程式的服務定義。如需相關資訊，請參閱 [以 Windows 服務執行 MFT 代理程式](#)。

例如，發出下列指令：

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

3. 接下來，使用 **setmqm** 指令將代理程式佇列管理程式從 IBM WebSphere MQ 7.5 移轉至更新版本。如需 **setmqm** 指令的相關資訊，請參閱 [將佇列管理程式與安裝相關聯](#)。
4. 最後，使用 **fteModifyAgent** 指令修改代理程式，以將代理程式重新配置為再次以 Windows 服務執行。

例如：

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

將 IBM WebSphere MQ File Transfer Edition 7.0 代理程式移轉至 IBM WebSphere MQ 7.5 或更新版本

使用 **fteMigrateAgent** 指令，將 IBM WebSphere MQ File Transfer Edition 7.0 代理程式移轉至 IBM WebSphere MQ 7.5 或更新版本。如果代理程式已配置成以 Windows 服務方式執行，則必須完成此作業中的選用步驟。

開始之前

V 9.0.0 從 IBM MQ 9.0 開始，Managed File Transfer 不支援 Web 代理程式。如果您嘗試使用 **fteMigrateAgent** 指令，將 Web 代理程式從舊版次移轉至 IBM MQ 9.0，則會顯示錯誤訊息，說明不支援移轉 Web 代理程式。

識別作為 IBM WebSphere MQ File Transfer Edition 的一部分而安裝的 IBM WebSphere MQ File Transfer Edition 代理程式實例。

IBM i 本主題中的資訊不適用於 IBM i。如果您使用 IBM i，在 IBM MQ 8.0 中使用 Managed File Transfer 之前，請先將佇列管理程式移轉至 IBM MQ 8.0。

關於這項作業

若要移轉至 IBM WebSphere MQ 7.5 版或更新版本，請先停止代理程式，然後將佇列管理程式安裝移轉至 IBM WebSphere MQ 7.5 版或更新版本，並另外選取 File Transfer 元件。移轉佇列管理程式之後，您可以使用 **fteMigrateAgent** 指令，將代理程式的配置移轉至 IBM WebSphere MQ 7.5 或更新版本的安裝。請使用 IBM WebSphere MQ 7.5 版或更新版本的佇列管理程式來啟動已移轉的代理程式。

如果代理程式透過 IBM WebSphere MQ 連結模式連接至其佇列管理程式，且佇列管理程式為 IBM WebSphere MQ 7.0.1 Fix Pack 6 版或更新版本，可以選擇使用多重安裝及下列步驟來執行並列移轉。如果您不想要執行並列移轉，請使用上述步驟移轉佇列管理程式及代理程式。

如果代理程式是使用 IBM WebSphere MQ 連結模式來連接至其佇列管理程式，而且此佇列管理程式為 IBM WebSphere MQ 7.0.1 Fix Pack 5 版或更舊版本，則必須將此安裝移轉至 IBM WebSphere MQ 7.0.1 Fix Pack 6 以允許並列移轉，或者必須直接將此安裝移轉至 IBM WebSphere MQ 7.5 版或更新版本。

如果代理程式以 IBM WebSphere MQ 用戶端身分透過網路連接至其佇列管理程式，則只能透過完成步驟 [第 16 頁的『7』](#) 來將代理程式移轉至 IBM WebSphere MQ 7.5 或更新版本。

如果代理程式配置成 Windows 服務，您就必須搭配 **-f** 參數來執行該指令。如需相關資訊，請參閱 [第 407 頁的『fteMigrate 代理程式: 將 FTE 7.0 代理程式移轉至 MQ 7.5 或更新版本』](#)。

程序

1. 隨著現有佇列管理程式安裝一起安裝 IBM WebSphere MQ 7.5 或更新版本，同時選取 MQ Server 和 File Transfer 元件。
2. 停止 IBM WebSphere MQ File Transfer Edition 7.0.x 代理程式。

3. 將佇列管理程式移至 IBM WebSphere MQ 7.5 版或更新版本的安裝。
4. 使用 **fteMigrateAgent** 指令，將代理程式配置從 IBM WebSphere MQ File Transfer Edition 7.0.x 安裝架構移轉至與 IBM WebSphere MQ 7.5 或更新版本整合的代理程式功能。
5. 選用項目: 如果 IBM WebSphere MQ File Transfer Edition 7.0.x 代理程式配置為 Windows 服務，請完成下列選用步驟:
 - a) 使用 **fteModifyAgent** 指令重新配置代理程式，使它不再是 Windows 服務。
 - b) 從 IBM WebSphere MQ 7.5 或更新版本安裝中使用 **fteModifyAgent** 指令，將代理程式的 IBM WebSphere MQ 7.5 或更新版本配置為 Windows 服務。
6. 啟動 IBM WebSphere MQ 7.5 版或更新版本的代理程式。
7. 選用項目: 若要移轉作為用戶端透過網路連接至其佇列管理程式的代理程式，請執行下列動作:
 - a) 將 IBM WebSphere MQ 7.5 版或更新版本安裝到系統上
 - b) 使用 **fteMigrateAgent** 指令，將代理程式配置從 WMQFTE 7.0.x 安裝架構移轉至與 IBM WebSphere MQ 7.5 或更新版本整合的代理程式功能。
 - c) 停止 IBM WebSphere MQ File Transfer Edition 7.0.x 代理程式。
 - d) 如果 IBM WebSphere MQ File Transfer Edition 7.0.x 代理程式配置為 Windows 服務，請從 IBM WebSphere MQ File Transfer Edition 7.0.x 安裝中使用 **fteModifyAgent** 指令重新配置代理程式，使其不會以 Windows 服務方式執行。
 - e) 如果 IBM WebSphere MQ File Transfer Edition 7.0.x 代理程式配置為 Windows 服務，請從 IBM WebSphere MQ 7.5 或更新版本安裝中使用 **fteModifyAgent** 指令，將 IBM WebSphere MQ 7.5 或更新版本代理程式配置為 Windows 服務。
 - f) 啟動 IBM WebSphere MQ 7.5 版或更新版本的代理程式。

將 FTE 7.0 資料庫日誌程式移轉至 MQ 7.5 或更新版本

使用 **fteMigrateLogger** 指令，將獨立式 IBM WebSphere MQ File Transfer Edition 7.0 資料庫日誌程式移轉至 IBM WebSphere MQ 7.5 版或更新版本。如果您已將資料庫日誌程式配置為 Windows 服務，您必須完成額外的移轉步驟。

開始之前

識別獨立式資料庫日誌程式的實例。您必須先更新資料庫日誌程式實例，再更新 Managed File Transfer 網路的其他部分，讓這些實例可以正確處理它們接收的最新版傳送日誌訊息。

關於這項作業

若要直接移轉至 IBM WebSphere MQ 7.5 或更新版本，請先停止日誌程式，然後將此安裝移轉至所需版本。完成此移轉後，請使用 **fteMigrateLogger** 指令，將資料庫日誌程式配置移轉至所需的 IBM MQ 版本。

如果佇列管理程式是 IBM WebSphere MQ File Transfer Edition 7.0.1 Fix Pack 6 版或更新版本，您可以選擇使用下列步驟所述的多重安裝來進行並列移轉。如果您不想要執行並列移轉，請使用上述步驟移轉佇列管理程式及日誌程式。

如果資料庫日誌程式連接至 IBM WebSphere MQ File Transfer Edition 7.0.1 Fix Pack 5 或更舊版本的佇列管理程式，則您必須先將此安裝移轉至 IBM WebSphere MQ File Transfer Edition 7.0.1 Fix Pack 6 以允許並列移轉，否則您必須直接將此安裝移轉至 IBM WebSphere MQ 7.5 或更新版本，並另外選取 File Transfer 元件。

如果資料庫日誌程式配置成 Windows 服務，則您必須執行搭配 **-f** 參數的 **fteMigrateLogger** 指令。如需相關資訊，請參閱第 411 頁的『[fteMigrate 日誌程式: 將 FTE 7.0 資料庫日誌程式移轉至 MQ 7.5 或更新版本](#)』。

程序


1. 隨現有 IBM WebSphere MQ File Transfer Edition 7.0 安裝一起安裝 IBM WebSphere MQ 7.5 或更新版本，並選取 MQ 伺服器及 Managed File Transfer Service 元件。
2. 停止 IBM WebSphere MQ File Transfer Edition 7.0 資料庫日誌程式。

3. 將佇列管理程式移至 IBM WebSphere MQ 7.5 版或更新版本的安裝。
4. 使用 **fteMigrateLogger** 指令，將資料庫日誌程式配置從 IBM WebSphere MQ File Transfer Edition 7.0 的安裝移轉至 IBM WebSphere MQ 7.5 版或更新版本的資料庫日誌程式。
5. 執行 `MQ_INSTALLATION_PATH/mqft/sql` 中的 SQL Script，以建立資料庫日誌程式啟動所需的表格。每提高一個產品層次都需執行一個 Script，請依序執行所有適當的 Script。

您必須依照版本順序執行升級 SQL Script，從 IBM WebSphere MQ File Transfer Edition 或 Managed File Transfer 的現行層次開始著手。可用的 Script（其中 *** 可以是 Db2、Oracle 或 z/OS）如下所示：

- `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，如果您使用 IBM WebSphere MQ File Transfer Edition 7.0.3 並移轉至 IBM MQ 8.0.0.0，請執行 703-704、704-750、750-7502 及 7502-800。

註：  在 z/OS 上，可以使用 `ftelog_tables_zos_704-800.sql` 直接從 704 跳到 800。

直至 `ftelog_tables_***_704-750.sql` 的 Script 包含在 IBM MQ 8.0.0.0 中，並且位於 `MQ_INSTALLATION_PATH/mqft/sql` 目錄中。IBM MQ 8.0.0.0 未隨附 750 以上的升級項目；如果它們不在目錄中，您可以從 [APAR IT01841](#) 中指定的鏈結下載它們。

6. 選用項目：如果 IBM WebSphere MQ File Transfer Edition 7.0 資料庫日誌程式配置為 Windows 服務，請完成下列步驟：
 - a) 使用 `fteModifyDatabaseLogger` 指令重新配置資料庫日誌程式，使其不再是 Windows 服務。
 - b) 使用 **fteModifyLogger** 指令，重新配置 IBM WebSphere MQ 7.5 或更新版本日誌程式中的 Managed File Transfer，使其成為 Windows 服務。
7. 在 IBM WebSphere MQ 7.5 或更新版本資料庫日誌程式中啟動 Managed File Transfer。

結果

資料庫日誌程式現在已從 IBM WebSphere MQ File Transfer Edition 7.0 移轉至 IBM WebSphere MQ 7.5 版或更新版本。

增加 Windows、UNIX 或 Linux 上 Db2 日誌資料庫的分頁大小

如果您的資料庫是 Windows、UNIX 或 Linux 系統上的 Db2，且您已建立分頁大小小於 8 KB 的日誌資料庫，則必須先增加資料庫的分頁大小，然後再移轉至 IBM WebSphere MQ File Transfer Edition 7.0.3 或更新版本的表格。

程序

1. 如果尚未停止資料庫日誌程式，請使用 **fteStopDatabaseLogger** 指令停止資料庫日誌程式。
2. 使用 Db2 提供的工具備份日誌資料庫。
3. 使用 Db2 **export** 指令，將日誌資料庫表格中的資料傳送至磁碟上的檔案。

註：對於包含大型物件的表格，必須指定大型物件檔案。這些表格為 `CALL_RESULT` 及 `METADATA`。

4. 捨棄日誌資料庫表格。
5. 以至少 8KB 的分頁大小，以及至少 8KB 分頁大小的關聯緩衝池，建立表格空間。
指定新表格空間的名稱。例如，`FTE8KSPACE`。
6. 編輯 `ftelog_tables_db2.sql` 檔案，使指令在新表格空間中建立表格。
在 `ftelog_tables_db2.sql` 檔案中，將所有出現的文字 `IN "USERSPACE1"` 變更為 `IN "new_tablespace_name"`。例如，將 `IN "USERSPACE1"` 變更為 `IN "FTE8KSPACE"`。
7. 針對資料庫執行 `ftelog_tables_db2.sql` 檔案中的 SQL 指令。

8. 使用 Db2 **load** 指令，將匯出的資料傳送至新表格。

註：

- 根據在輸入檔中找到的直欄名稱來對映直欄名稱。確保這些已變更其結構的表格中，輸入直欄名稱與目標直欄名稱相符。
- 針對所有表格（MONITOR 及 TRANSFER 除外）的身分直欄，必須指定 IDENTITY OVERRIDE 行為。指定此行為可確保在載入作業期間不會重新產生列 ID。

9. 按照給定的順序對下列表格執行 Db2 **set integrity** 指令（完整性狀態值為 **immediate** 及 **checked**）：

- CALL_ARGUMENT
- MONITOR
- MONITOR_ACTION
- MONITOR_EXIT_RESULT
- MONITOR_METADATA
- SCHEDULE_ACTION
- 排程
- SCHEDULE_ITEM
- TRANSFER
- TRANSFER_CALLS
- TRANSFER_EVENT
- TRANSFER_ITEM
- TRANSFER_STATS
- TRIGGER_CONDITION

10. 在具有產生的 ID 直欄的表格中，將 ID 產生器設定為從高於現有的最高 ID 值的值開始。

下列表格具有產生的 ID 直欄：

- AUTH_EVENT
- CALL
- CALL_ARGUMENT
- CALL_RESULT
- FILE_SPACE_ENTRY
- METADATA
- MONITOR_ACTION
- MONITOR_EXIT_RESULT
- MONITOR_METADATA
- 排程
- 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

若要將這些表格的產生的 ID 設定為正確值，請針對每個表格執行下列步驟：

- a) 決定現有資料中的最大 ID 值。

可以透過執行以下 SQL 陳述式找到此值：

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

從此指令傳回的值即為指定表格中的最大現有 ID。

- b) 變更表格，以將 ID 產生器設定為從新值開始，該值比上一步驟所傳回的值大 1。

可以透過執行以下 SQL 陳述式來設定此值：

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

相關工作

第 117 頁的『[從獨立式資料庫日誌程式移轉至 MFT 的 Java EE 資料庫日誌程式](#)』

您可以從獨立式資料庫日誌程式移轉至 Java EE 資料庫日誌程式。必須停止獨立式資料庫日誌程式，然後再安裝 Java EE 資料庫日誌程式。若要避免日誌項目遺失或重複，您必須在停止獨立式資料庫日誌程式之前，阻止訊息被發佈至 SYSTEM.FTE 主題，然後在 JEE 資料庫日誌程式安裝完成之後再加以重新啟動。移轉之前，請先備份資料庫。

第 19 頁的『[將 Db2 on z/OS 上的資料庫表格移轉至 IBM MQ 8.0](#)』

如果您的資料庫是 z/OS 系統上的 Db2，則必須完成下列步驟，以在 IBM MQ 8.0 中從 IBM WebSphere MQ File Transfer Edition 7.0.3 移轉至 IBM WebSphere MQ File Transfer Edition 7.0.4，以及從 IBM WebSphere MQ File Transfer Edition 7.0.4 移轉至 Managed File Transfer。Db2 表格的結構與舊版不同。例如，某些表格有新的直欄，某些變數字元直欄可能更大，因此必須將舊版中的表格移轉至 Db2 格式。

將 Db2 on z/OS 上的資料庫表格移轉至 IBM MQ 8.0

如果您的資料庫是 z/OS 系統上的 Db2，則必須完成下列步驟，以在 IBM MQ 8.0 中從 IBM WebSphere MQ File Transfer Edition 7.0.3 移轉至 IBM WebSphere MQ File Transfer Edition 7.0.4，以及從 IBM WebSphere MQ File Transfer Edition 7.0.4 移轉至 Managed File Transfer。Db2 表格的結構與舊版不同。例如，某些表格有新的直欄，某些變數字元直欄可能更大，因此必須將舊版中的表格移轉至 Db2 格式。

關於這項作業

IBM MQ 8.0 中的 Managed File Transfer 支援傳送非常大的檔案，其中檔案大小大於可儲存為整數 (32 位元)。有一個定義使用 BIGINT (64 位元) 數字。您可以選擇使用整數定義 (稱為 `ftelog_tables_zos.sql`) 或 BIGINT 定義 (稱為 `ftelog_tables_zos_bigint.sql`)，它們位於 `prod/mqf/sql` 目錄中。

BIGINT 資料類型可以在 IBM WebSphere MQ File Transfer Edition 7.0.4 Fix Pack 3 中使用，或在 IBM WebSphere MQ 7.5.0 Fix Pack 1 或更新版本中使用 Managed File Transfer。

若要啟用 BIGINT 資料類型，您必須使用 Db2 9.1 或更新版本。指定有傳送檔案大小及各項傳送所關聯之表格 ID 的欄位，會使用 INTEGER 資料類型。如果您要記錄超過 2GB 檔案大小的傳送，或將每一個超過 2,147,483,648 的傳送儲存在資料庫中，您必須使用 BIGINT SQL 檔案。

下列清單概述您需要遵循的程序：

1. 您有來自 IBM WebSphere MQ 7.0 的現有表格。這些表格有一個綱目，例如 FTELOG。
2. 使用不同的綱目名稱來建立 IBM MQ 8.0 表格，例如 FTEV8。這可讓您將資料從 `FTELOG.table` 複製到 `FTE8.table`。
3. 將資料複製到新表格。
4. 在新表格中設定產生的 ID 值。
5. 執行 `fteMigrateLogger` 指令，將內容檔移到目錄結構中的新位置。

6. 編輯日誌程式內容檔以指定新綱目 (FTEV8)
7. 編輯現有的 Logger JCL，以使用 IBM MQ 8.0 Managed File Transfer 程式庫。
8. 啟動日誌程式。
9. 日誌程式運作之後，您可以刪除 FTELOG 表格。

在下列說明中，Managed File Transfer 產品是安裝在 USS 的 /HMF8800 目錄中。

程序

1. 如果您尚未停止資料庫日誌程式，請使用 USS 或 P `loggerjob` 中的 `fteStopDatabaseLogger` 指令停止資料庫日誌程式。
2. 發出指令 `ls /HMF8800/mqft/sql`，以列出目錄中的 SQL 檔案。
如果您要使用 BIGINT 數字，請將 `ftelog_tables_zos_bigint.sql` 複製到您的起始目錄，否則請將 `ftelog_tables_zos.sql` 複製到您的起始目錄。
3. 編輯您已移至起始目錄的檔案：
 - a) 將 `ftelog` 變更為新表格的綱目名稱。
 - b) 確保每個索引都具有唯一名稱。
若要這麼做，請在編輯階段作業中執行下列動作：
 - i) 排除所有行。
 - ii) 尋找 `'CREATE UNIQUE INDEX ' ALL`
 - iii) 變更 `_KEY _K8Y ALL NX`
4. 檢查此檔案以確保所有陳述式都在第 71 欄中。
如果陳述式不在第 71 欄中，請分割第 71 欄前面的那一行。
5. 您可以使用此檔案作為 SQL 的輸入，也可以將它複製到 PDS。若要這麼做，請編輯 PDS，然後使用 `copy` 指令，同時指定目錄和檔名。
6. 請洽詢 Db2 管理者以取得定義的相關資訊，因為您可能需要遵循一些網站標準。
7. 執行下列動作：
 - a) 將位於 `/HMF8800/mqft/sql/ftelog_tables` 目錄中的 `_zos_704-800.sql` 檔案複製到您的起始目錄。
 - b) 編輯此檔案。將 `FTESRC` 變更為現有綱目 (FTELOG)，並將 `FTEDEST` 變更為新綱目 (FTEV8)。
 - c) 檢查此檔案以確保所有陳述式都在第 71 欄中。
如果陳述式不在第 71 欄中，請分割第 71 欄前面的那一行。
 - d) 如果將 `DB2 RUNSTATS` JCL 用於 Managed File Transfer 表格，請建立新的工作，同時指定新的綱目和表格。
8. 某些表格包含產生的 ID，以強制每一列具有唯一 ID，而您需要設定這些 ID。
下列表格具有產生的 ID 直欄：
 - AUTH_EVENT
 - CALL
 - CALL_ARGUMENT
 - CALL_RESULT
 - FILE_SPACE_ENTRY
 - METADATA
 - MONITOR_ACTION
 - MONITOR_EXIT_RESULT
 - MONITOR_METADATA
 - 排程
 - 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

若要將這些表格的產生的 ID 設定為正確值，請針對每個表格執行下列步驟：

- a) 決定現有資料中的最大 ID 值。

可以透過執行以下 SQL 陳述式找到此值：

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

從此指令傳回的值即為指定表格中的最大現有 ID。

- b) 變更表格，以將 ID 產生器設定為從新值開始，該值比上一步驟所傳回的值大 1。

可以透過執行以下 SQL 陳述式來設定此值：

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

9. 編輯資料庫內容檔以指定新的綱目名稱：

- a) 如果 Managed File Transfer 配置目錄是 /u/userid/fteconfig，您可以使用 USS 指令 **find /u/userid/fteconfig -name databaselogger.properties** 來尋找日誌程式的內容檔。
- b) 編輯此檔案，並將 `wmqfte.database.schema` 變更為新的綱目值。

10. 在嘗試使用日誌程式之前，請發出下列指令將目錄樹狀結構轉換為 8.0.0 格式：

- a) **fteMigrateConfigurationOptions**
- b) **fteMigrateLogger**

這會將 `databaselogger.properties` 複製到 `logger.properties`。

11. 編輯現有的 Logger JCL，以使用 8.0.0 Managed File Transfer 程式庫。

12. 啟動日誌程式。

日誌程式開始運作後，您即可刪除第 7 版 FTELOG 表格。

將 Managed File Transfer for z/OS 代理程式移轉至更新版本

遵循下列指示，將 IBM MQ Managed File Transfer for z/OS 代理程式移轉至更新版本。本主題使用 IBM MQ 8.0 作為起始版本，並使用 IBM MQ 9.0 作為更新版本。

重要：這些指示假設下列要點：

- 在移轉期間，Managed File Transfer for z/OS (MFT) 拓撲的協調佇列管理程式及指令佇列管理程式不會變更。
- 未移轉代理程式佇列管理程式。在移轉之前及之後，代理程式佇列管理程式會保持在相同的 IBM MQ for z/OS 層次。
- 要移轉的代理程式是使用 JCL 來管理。

步驟 1-起始設定

1. 將 IBM MQ Managed File Transfer for z/OS 9.0 安裝至與 IBM MQ Managed File Transfer for z/OS 8.0 相同的 LPAR。
2. 在 IBM MQ 9.0 中尋找 IBM MQ MFT 的 SBFGCMD5 PDSE。
如需 SBFGCMD5 PDSE 的相關資訊，請參閱 SBFGCMD5 程式庫。
3. 在此資料集內，編輯 BFGCOPY 成員以指定要用來管理代理程式的 PDSE 名稱。
4. 提交 BFGCOPY 成員。這會以步驟 [第 22 頁的『3』](#) 中指定的名稱，將現有的 SBFGCMD5 PDSE 複製到新的 PDSE。

步驟 2-修改新的 PDSE

1. 在您使用 BFGCOPY 成員建立的新 PDSE 內，編輯 BFGCUSTOM 並更新下列變數：

變數	值
++library++	將此變數取代為新 PDSE 的名稱。
armELEMENT	將此值設為 IBM MQ 8.0 代理程式所使用的相同值。
armELEMENTYPE	將此值設為 IBM MQ 8.0 代理程式所使用的相同值。
BFG_DATA	將此值設為 IBM MQ 8.0 代理程式所使用的相同值。
BFG_GROUP_NAME	將此值設為 IBM MQ 8.0 代理程式所使用的相同值。
BFG_JAVA_HOME	這需要設為用來執行代理程式之「Java 執行時期環境」的「UNIX 系統服務 (USS)」路徑。將此值設為 IBM MQ 8.0 代理程式所使用的相同值。
BFG_JVM_PROPERTIES	將此值設為 IBM MQ 8.0 代理程式所使用的相同值。
BFG_PROD	在 IBM MQ 9.0 安裝中，此變數應該設為 Managed File Transfer for z/OS 的 USS 路徑。
BFG_WTO	將此值設為 IBM MQ 8.0 代理程式所使用的相同值。
CLEAN_AGENT_PROPS	將此設為空白。
connectionQMGr	將此值設為 IBM MQ 8.0 代理程式所使用的相同值。
coordinationQMGr	將此值設為 IBM MQ 8.0 代理程式所使用的相同值。
CREDENTIALS_PATH	將此設為空白。
DB_PROPS_PATH	將此值設為 IBM MQ 8.0 代理程式所使用的相同值。
DLQ_HLQ	將此值設為 IBM MQ 8.0 代理程式所使用的相同值。
FTE_CONFIG	將此設為空白。
MQ_HLQ	將此值設為 IBM MQ 8.0 代理程式所使用的相同值。
MQ_PATH	對於與代理程式佇列管理程式相關聯的 IBM MQ 安裝，此變數需要設為 USS 路徑。
名稱	將此設為正在移轉的代理程式名稱。
QM	將此值設為 IBM MQ 8.0 代理程式所使用的相同值。這可確保當代理程式執行 IBM MQ 9.0 時，代理程式所使用的代理程式佇列管理程式與執行 IBM MQ 8.0 時所使用的代理程式佇列管理程式相同。
SERVICE_TYPE	AGENT
TMPDIR	將此值設為 IBM MQ 8.0 代理程式所使用的相同值。

2. 儲存已修改的 BFGCUSTM 成員。
3. 提交 BFGCUSTM。當代理程式使用 IBM MQ 9.0 時，成員會在 PDSE 內建立可用來管理代理程式的新成員。



小心: 成員需要 PDSE 的互斥存取權，因此您必須在 BFGCUSTM 執行時停止使用它。

步驟 3-執行移轉

1. 找出用來管理代理程式的 IBM MQ 8.0 PDSE。
2. 在此 PDSE 內，提交 BFGAGSP 成員以停止代理程式。代理程式完成其所有受管理傳送之後即會停止。
3. 接下來，尋找剛剛為代理程式建立的新 IBM MQ 9.0 PDSE。
4. 在此 PDSE 內，提交 BFGAGST 成員。這會使用 IBM MQ Managed File Transfer for z/OS 9.0 來啟動代理程式

相關概念

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

在 IBM MQ for z/OS 系統上，您可以套用本主題中的指示，作為在企業中使用 Managed File Transfer 的替代方法。

將 MFT 移轉至具有不同作業系統的新機器

順利將 MFT 配置移轉至新系統或平台所需的核心步驟。此作業主要聚焦於 MFT 配置移轉，但也會在適當時討論佇列管理程式移轉。

開始之前

請確定您要移轉的任何代理程式已完成任何進行中或擱置中傳送，且您已取得下列項目的備份：

- 協調佇列管理程式
- 代理程式佇列管理程式
- 代理程式
- 資源監視器
- 傳送範本
- 排程傳送

重要: 除非舊系統和新系統只有一個安裝，或您在 IBM MQ 安裝程序中指定安裝名稱，否則一個系統上的 IBM MQ 安裝名稱不可能符合新系統上的安裝名稱。

關於這項作業

下列移轉程序基於下列實務範例，其中 QMA 既是拓撲的協調佇列管理程式，也是稱為 Agent1 之代理程式的代理程式佇列管理程式。

Agent1 具有監視器、傳送範本及排程傳送。QMA 也會使用其傳送端和接收端通道來進行檔案傳送，以連接至另一個系統上執行的佇列管理程式 QMB。

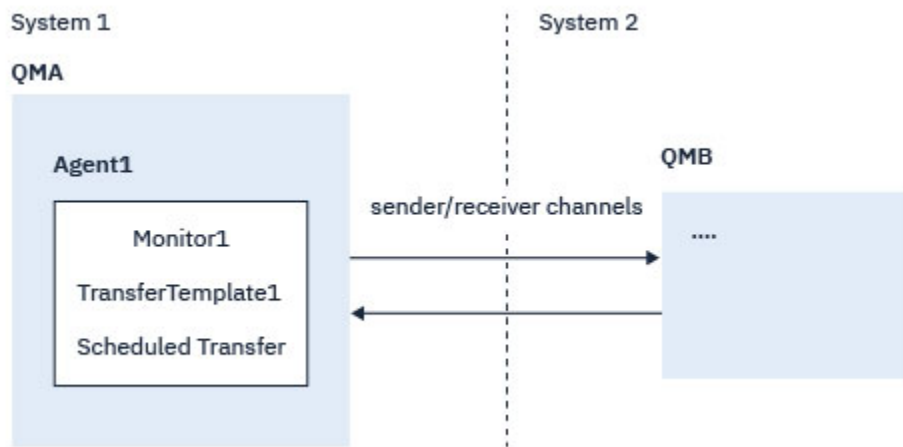


圖 1: 移轉 System 1 上的 MFT 配置



小心: 下列程序僅說明如何備份及還原 MFT 配置。如果您要將 MFT 移轉至具有相同作業系統的新機器，則可以將所有資料檔案從舊系統複製到新系統上的適當目錄，以備份及還原佇列管理程式資料及日誌檔。

不過，如果新機器具有不同的作業系統，則無法移轉資料檔，因為它們是建立在特定平台專用的。

程序

1. 備份程序

- a) 使用 **dmpmqcfig** 指令儲存佇列管理程式配置，稍後再從其定義重建它。

例如：

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

- b) 備份儲存在 IBM MQ 資料目錄 `/MQ_DATA_PATH/mqft` 下之代理程式的配置檔

`mqft` 目錄通常有三個子目錄，分別是 `config`、`installation` 和 `logs`。這些分別包含代理程式安裝資料、配置及資料庫日誌程式檔案。

如果代理程式是「通訊協定橋接器代理程式」，則也需要備份代理程式配置目錄中的 `ProtocolBridgeCredentials.xml` 檔案。此檔案定義通訊協定橋接器代理程式用來向通訊協定伺服器授權其本身的使用者名稱及認證資訊。

- c) 使用 MFT **ftelistMonitors** 指令並搭配 **-ox** 選項，將資源監視器的配置匯出至 XML 檔案。

例如：

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

- d) 使用 MFT **fteListTemplates** 指令搭配 **-x** 及 **-o** 選項，將傳送範本匯出至 XML 檔案。

例如，下列指令會在現行目錄中建立 `TransferTemplate1.xml`：

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

- e) 手動備份排定的傳送定義。

無法將定義匯出至 XML 檔案，但您可以使用 MFT **fteListScheduledTransfers** 指令並手動備份定義，來列出排定的傳送。

2. 重建程序

- a) 在新系統上安裝 IBM MQ 和 MFT 之後 QMA 重建佇列管理程式。

- b) 執行 **runmqsc** 指令，以在步驟 [第 24 頁的『1.a』](#) 中儲存的佇列管理程式配置中進行剖析，以還原 QMA 配置

例如：


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

- c) 在系統 2 上重建連接至 QMB 的傳送端及接收端通道。
- d) 在 QMB 佇列管理程式端，更新連線詳細資料，例如連接至 QMA 之傳送端通道的主機名稱及埠號。
- e) 將所有已備份的代理程式配置檔複製到新系統，以重建 Agent1，然後啟動代理程式。
- f) 使用 MFT `fteCreateMonitor` 指令搭配 **-ix** 及 **-f** 選項，來匯入 Monitor1 的 XML 檔案。
例如：

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

- g) 將訊息內文中包含 TransferTemplate1.xml 內容的訊息發佈至 SYSTEM.FTE 主題。
使用獨立式應用程式，並指定主題字串：

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

其中 `<template_id>` 是可在 TransferTemplate1.xml 檔案內找到的傳送範本 ID。
例如，如果 xml 包含：

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

，主題字串應該為：

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

- h) 使用 MFT `fteCreateTransfers` 指令手動重建排程傳送。

MFT 實務練習概觀

這個小節列出一般 Managed File Transfer 拓撲，以及一個設定系統和傳送測試訊息的實務練習。

- [一般拓撲](#)
- [配置基本伺服器](#)

MFT 一般拓撲

這個小節列出一般 Managed File Transfer 拓撲。每一個圖表中的雙向箭頭都代表佇列管理程式的連線。如需佇列管理程式連線選項的相關資訊，請參閱第 28 頁的『[連線功能考量](#)』。

含有一個佇列管理程式的基本拓撲

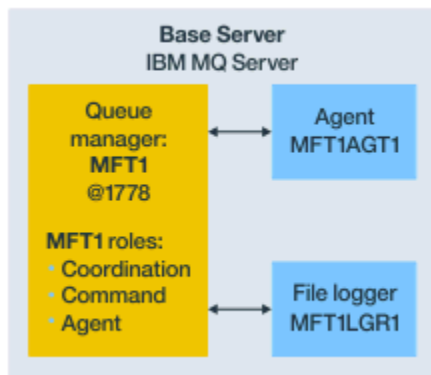


圖 2: 含有一個佇列管理程式的基本拓撲

基本拓撲代表一個完整配置，其中包括協調佇列管理程式。其配置名稱與協調佇列管理程式的名稱相同。如果協調佇列管理程式名稱是 MFT1，則配置名稱就是 MFT1。

基本拓撲是您完成的第一個 Managed File Transfer 配置。完成基本配置之後，來自遠端伺服器的友機代理程式即會新增至基本配置，以交換檔案。

基本拓撲不在基本拓撲伺服器之外交換檔案。不過，基本拓撲可讓您將檔案移至相同伺服器的不同位置，並可供開發用途使用。

含有一個友機代理程式的基本拓撲

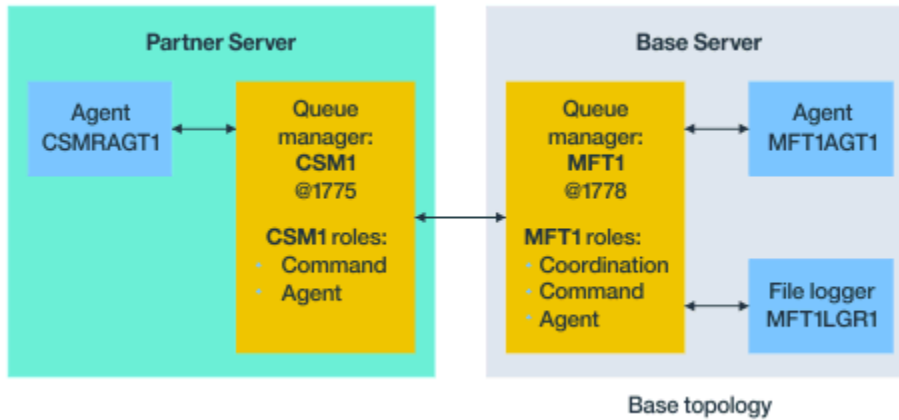


圖 3: 含有一個友機代理程式的基本拓撲

此拓撲可在兩個代理程式之間交換檔案。可以用新增第一個代理程式的類似方式來新增額外的友機代理程式。

您可以對所有這三個 Managed File Transfer 佇列管理程式角色使用單一佇列管理程式，或者，您可以對特定角色使用專用的佇列管理程式。

例如，您可以有一個佇列管理程式專用於協調佇列管理程式角色，而指令及代理程式角色可共用第二個佇列管理程式。

與基本配置不同的伺服器中的遠端代理程式佇列管理程式，與基本配置協調佇列管理程式，這兩者之間的連線必須配置成 IBM MQ 用戶端或 MQI 通道。

協調佇列管理程式的連線由 **fteSetupCoordination** 指令建立。如果協調佇列管理程式連線未配置成 IBM MQ 用戶端通道，則在友機伺服器上，由友機代理程式伺服器發出的 **fteListAgents** 之類的指令會失敗。

含有不同的協調佇列管理程式和一個友機代理程式的基本拓撲

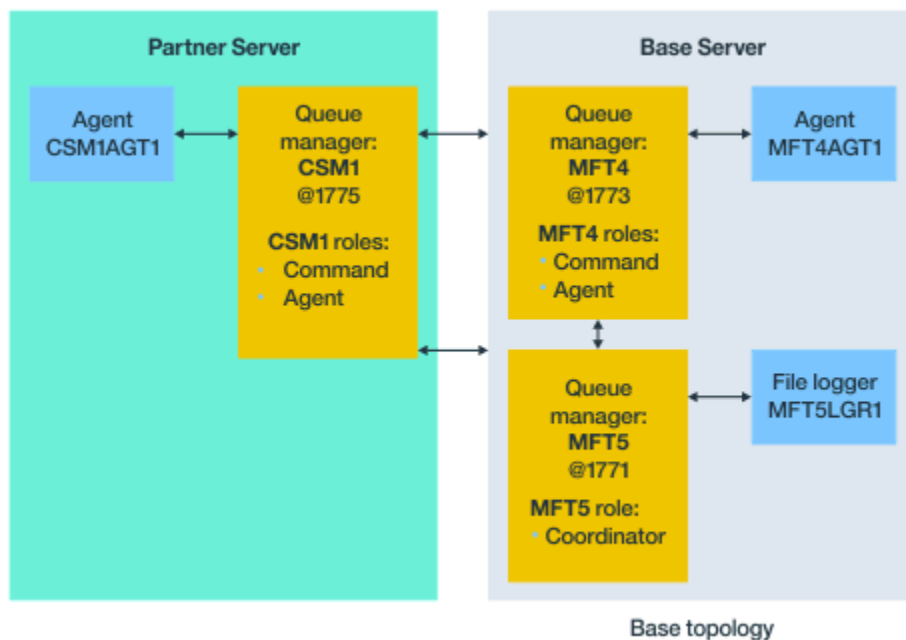


圖 4: 含有不同的協調佇列管理程式和一個友機代理程式的基本拓撲

在圖 3 的基本拓撲中，在基本伺服器上，佇列管理程式 MFT4 由指令與代理程式角色共用，佇列管理程式 MFT5 則為協調佇列管理程式角色專用。

連線功能必須存在於此拓撲中的所有佇列管理程式之間，包括基本拓撲中的佇列管理程式 MFT4 和 MFT5。

在友機伺服器佇列管理程式上，佇列管理程式 CSM1 具有代理程式和指令佇列管理程式的角色。

此拓撲可在兩個代理程式之間交換檔案。每一個友機代理程式都必須連接至佇列管理程式，如此圖所示。可以用新增第一個友機代理程式的類似方式來新增額外的友機代理程式。

含有 Managed File Transfer Agent 友機的基本拓撲

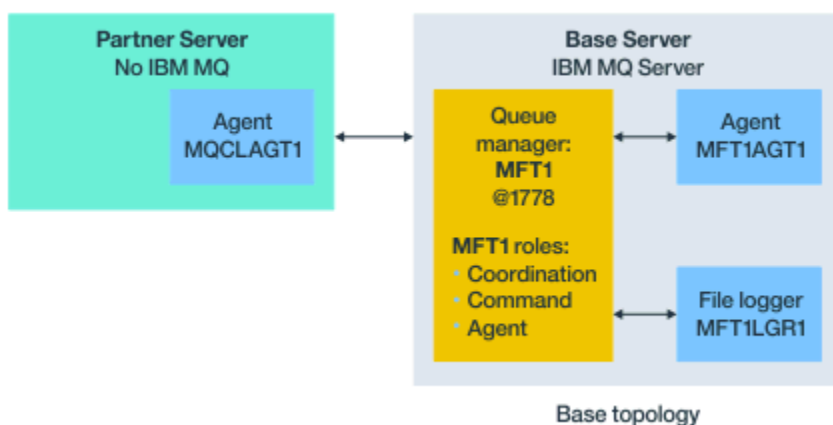


圖 5: 含有 Managed File Transfer Agent 友機的基本拓撲

此拓撲可在兩個代理程式之間交換檔案。

友機代理程式（在圖表中是描述為 MQCLAGT1）中的伺服器未安裝 IBM MQ 伺服器。

友機代理程式是使用與 IBM MQ 已安裝伺服器相同的指令進行配置，但有些例外情況：

- 此友機代理程式與基本佇列管理程式或佇列管理程式之間的配置必須使用 IBM MQ 用戶端連線。

- 不需要執行由友機代理程式伺服器中的配置指令所建立的協調佇列管理程式角色 IBM MQ 定義。協調佇列管理程式定義已存在於基本伺服器中。

但是，您必須：

- 複製在友機伺服器中建立代理程式時所產生的代理程式物件定義
- 將定義檔傳送至基本配置伺服器，以及
- 在識別為基本伺服器中的代理程式佇列管理程式的佇列管理程式中建立定義。

在此情況下，MFT1 負責處理所有這三個角色，而由您在 MFT1 佇列管理程式中建立代理程式 MQCLAGT1 的物件。

除了將物件定義複製到基本伺服器之外，您可以有別的替代方案，亦即在代理程式佇列管理程式所在的基本伺服器上，對 MQCLAGT1 代理程式執行 **fteDefine** 指令。請使用 **fteDefine** 指令產生的定義，在代理程式佇列管理程式上建立所需的代理程式定義。

例如，如圖表中所示，您可以將檔案 MQCLAGT1_create.mqsc 從友機伺服器中的代理程式目錄複製到基本配置伺服器，然後在 MFT1 佇列管理程式中建立所需的代理程式定義。

您在友機代理程式伺服器上完成的配置會建立 Managed File Transfer 配置目錄及所需的內容檔。

V 9.0.1 在友機伺服器上，從 IBM MQ 9.0.1 開始，您可以從 Fix Central 安裝 Managed File Transfer 可重新配送的用戶端。

註：MQMFT 可重新配送的用戶端與 IBM MQ 可重新配送的用戶端的不同之處在於，它是套裝的，不需要使用 **genmqpkg** 公用程式。如需相關資訊，請參閱 [可重新配送的用戶端](#)。

連線功能考量

在之前的圖表中，代理程式與佇列管理程式之間的每一條線都代表佇列管理程式的連線。

此連線可能是：

- 本端連線
- 連結或訊息通道連線，或者
- IBM MQ 用戶端或 MQI 連線。

您在配置中選取的連線類型視您指定的參數而定

- 當您指定佇列管理程式名稱參數而沒有其他連線參數時，請指定連結連線。

如果所使用的佇列管理程式是 Managed File Transfer 配置的本端，它也代表使用於基本配置伺服器中的本端連線。

- 如果您指定佇列管理程式名稱參數以及對應的主機、埠及通道名稱等參數，請指定 IBM MQ 用戶端連線。

當代理程式位於與代理程式佇列管理程式相同的主機上時，指定產生本端連線的連結類型將更有效率。

配置基本伺服器

如何設定含有不同的配置佇列管理程式的基本伺服器。

開始之前

下列範例假設您：

- 已檢閱過第 28 頁的『連線功能考量』小節，並瞭解如何影響配置中的佇列管理程式的連線類型。
- 有運作中的 IBM MQ 基礎架構。如需設定佇列管理程式的相關資訊，請參閱 [配置 IBM MQ 佇列管理程式](#)。
- IBM MQ 安全作業已完成。

所有系統資源，例如存取檔案，都已配置足夠的安全性。

如需 Managed File Transfer 安全配置，請參閱 [Managed File Transfer 的安全概觀](#) 以及 [Managed File Transfer 動作的使用者權限](#)。

- 不論是否透過以下哪一種方式來配置，在配置 IBM MQ 之後，所有 IBM MQ 連線都已經過測試：使用範例程式來傳送和接收訊息，或是使用 **amqscnxc** 範例來測試 IBM MQ 用戶端類型的連線。

amqscnxc 範例是透過在範例程式碼中定義通道連線來連接到佇列管理程式，此方式類似 Managed File Transfer 使用 MQI 或 IBM MQ 用戶端類型連線來進行連接的方式。

- 這些指示假設您用於基本配置的伺服器上已安裝一個 IBM MQ 版本。如果您在基本伺服器中有多個 IBM MQ 安裝，則必須對於您要使用的 IBM MQ 版本小心使用正確的檔案路徑。
- 這些指示中所使用的佇列管理程式不需要連線鑑別。

雖然不需要連線鑑別可以更容易完成第一次配置，但如果您的企業需要立即使用連線鑑別，請參閱 [Managed File Transfer](#) 和 [IBM MQ 連線鑑別](#)，以取得如何配置 `MQMFTCredentials.xml` 認證檔的指示

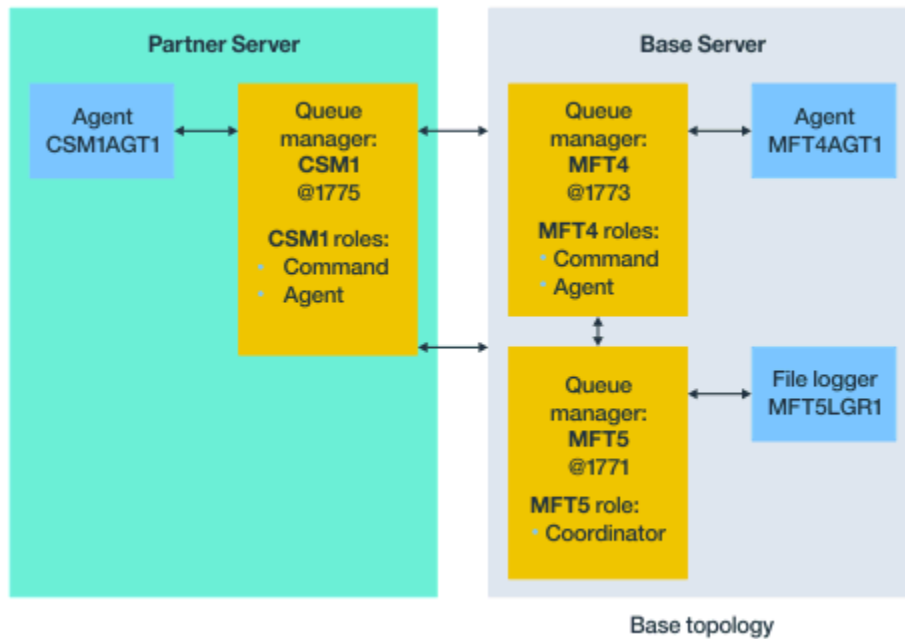


圖 6: 含有不同的協調佇列管理程式和一個友機代理程式的基本拓撲

關於這項作業

範例配置的佇列管理程式角色是：

- 基本伺服器
 - 佇列管理程式 MFT5 是協調佇列管理程式
 - 佇列管理程式 MFT4 是作為代理程式 MFT4AGT1 的代理程式佇列管理程式，並同時作為基本伺服器上的 MFT5 配置的指令佇列管理程式。
- 友機伺服器
 - 佇列管理程式 CSM1 兼作代理程式 CSM1AGT1 的代理程式佇列管理程式，以及友機伺服器上的 MFT5 配置的指令佇列管理程式。
 - 基本伺服器上的佇列管理程式 MFT5 是協調佇列管理程式。

程序

- 配置協調佇列管理程式
- 配置指令佇列管理程式
- 設定代理程式
- 設定日誌程式
- 配置友機伺服器

下一步

設定具有 MQMFT 的 MQExplorer，以測試您的範例設定。

配置協調佇列管理程式

如何配置協調佇列管理程式來協調檔案傳送。

開始之前

確保您對此實務練習設定的佇列管理程式之間有完整連線功能。

關於這項作業

這項作業設定協調佇列管理程式 MFT5，而且此章節中的指示假設您使用一個 IBM MQ 安裝。

如果您有多個安裝架構，則必須先使用 `setmqenv` 指令將 IBM MQ 路徑設定為所需的 IBM MQ 版本，然後再啟動任何配置作業。

程序

1. 以 Managed File Transfer 管理者身分登入。
2. 發出下列指令，以識別協調佇列管理程式並設定配置目錄結構：

```
fteSetupCoordination -coordinationQMgr MFT5
```

協調佇列管理程式目錄

```
C:\data\mqft\config\MFT5
```

coordination.properties 檔案

```
C:\data\mqft\config\MFT5\coordination.properties
```

此指令也會產生您必須對協調佇列管理程式 C:\data\mqft\config\MFT5\MFT5.mqsc 執行的 MQSC 指令檔：

3. 切換至 C:\data\mqft\config\MFT5 目錄。
4. 執行下列指令來配置佇列管理程式，以作為協調佇列管理程式。
您需要提供在步驟 第 30 頁的『2』中發出之指令所產生的 MQSC 指令檔：

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

5. 使用您偏好的編輯器來開啟 mft5.txt 結果檔案。同時確保已順利建立定義。

下一步

設定指令佇列管理程式。

配置指令佇列管理程式

如何配置指令佇列管理程式。

開始之前

請確定您已配置協調佇列管理程式。如需相關資訊，請參閱第 30 頁的『配置協調佇列管理程式』。

關於這項作業

這項作業可識別指令佇列管理程式。

程序

發出下列指令：

```
fteSetupCommands -connectionQMgr MFT4
```

您會取得下列訊息 BFGCL0245I：已順利建立 C:\data\mqft\config\MFT4\command.properties 檔案。

指令佇列管理程式不需要額外的 IBM MQ 定義。執行 **fteSetupCommands** 之後，會在 MFT5 配置目錄中建立 command.properties 檔案。

下一步

設定代理程式。

設定代理程式

如何準備檔案傳送代理程式 MFT4AGT1，包括您必須執行的 MQSC Script。

開始之前

您應該已設定指令佇列管理程式。如需相關資訊，請參閱第 30 頁的『配置指令佇列管理程式』。

關於這項作業

這項作業會準備 Windows 檔案傳送代理程式 MFT4AGT1。

程序

1. 發出下列指令：

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

當您使用 **fteCreateAgent** 指令建立代理程式之後，agents 目錄及代理程式 MFT4AGT1 的子目錄就會新增至 MFT5 目錄中。

您可以在 data\MFT5\agents\MFT4AGT1 目錄中找到下列項目：

- agent.properties 檔案
 - MFT4AGT1_create.mqsc 檔案，其中包含代理程式所需的 IBM MQ 定義。
2. 切換至 data\MFT5\agents\MFT4AGT1 目錄，並透過發出下列指令來建立所需的代理程式佇列管理程式定義：

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

3. 使用您偏好的編輯器來開啟 mft4.txt 結果檔案，並確定已順利建立這些定義。
4. 鍵入下列指令來啟動代理程式：**fteStartAgent** MFT4AGT1。
5. 鍵入下列指令來顯示代理程式：**fteListAgents**。

您應該會看到類似下列內容的輸出：

```
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
```

註：如果您尚未在 Managed File Transfer 環境中啟用連線鑑別，您可以忽略 BFGPR0127W 訊息。

如果您發出 **ftelistAgents** 指令並收到下列訊息: BFGCL0014W: 不存在符合現行選取準則的代理程式。 , 如需進一步資訊, 請參閱第 253 頁的『[ftelistAgents 指令未列出您的 MFT 代理程式時應該如何處理](#)』。

下一步

設定日誌程式。

設定日誌程式

需要有檔案或資料庫日誌程式, 才能保留配置的傳送活動的歷程和審核相關資訊。在此範例中, 您會建立檔案日誌程式。

開始之前

您必須已設定:

- 配置佇列管理程式
- 指令佇列管理程式
- 代理程式

程序

1. 發出下列指令:

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

執行 **fteCreateLogger** 指令之後, 會建立 `data\mqft\config\MFT5\loggers` 目錄以及 `MFT5LGR1` 子目錄。

`MFT5LGR1` 子目錄保留 `logger.properties` 檔案。另外, 在此目錄中還有一個稱為 `MFT5LGR1_create.mqsc` 的檔案, 它含有日誌程式所需的 IBM MQ 定義。

2. 切換至 `data\mqft\config\MFT5\loggers\MFT5LGR1` 目錄。
3. 執行關聯的 MQSC 指令檔。

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

以建立日誌程式所需的定義。

a) 檢閱物件定義的結果, 確認已順利建立所需的物件。

4. 發出下列指令來啟動日誌程式: **fteStartLogger** MFT5LGR1。

5. 檢閱 `data\mqft\logs\MFT5\loggers\MFT5LGR1\logs` 中的 `output0.log` 檔案的內容。

在日誌程式的一些相關資訊之後, 最後的陳述應包含此訊息: BFGDB0023I: 日誌程式已完成啟動活動, 現在正在執行中。

有時候, 日誌程式第一次啟動時, 日誌資訊可能不會寫入 `output0.log`。如果 `output0.log` 檔案是空的, 請鍵入 **fteStopLogger** MFT5LGR1, 並按 **Enter** 鍵來重新啟動日誌程式。

鍵入 **fteStartLogger** MFT5LGR1, 並按 **Enter** 鍵來重新啟動日誌程式。 `output0.log` 檔案現在會顯示資料。

第一次啟動代理程式時, 相同的行為會延伸至 `output0.log` 檔案的代理程式版本。

使用 **fteStopAgent** 和 **fteStartAgent** 指令來停止及啟動代理程式。然後, 您會看到日誌資料寫入至代理程式 `output0.log` 檔案。

結果

您已配置基本伺服器, 其中包括此配置的協調佇列管理程式。

下一步

現在，您可以對友機伺服器（其包含遠端代理程式）執行類似工作。

配置友機伺服器

當基本伺服器有不同的協調佇列管理程式時該如何配置友機伺服器

開始之前

確保您已全部完成設定基本伺服器（包含配置佇列管理程式）的所有作業。

關於這項作業

對於 IBM MQ 採用相同的假設，同時安全配置以及 IBM MQ 路徑也會套用至友機伺服器。

開始時，先設定 MFT5 配置目錄，並使用 **fteSetupCoordination** 指令來識別協調佇列管理程式。

程序

1. 發出下列指令來建立友機伺服器配置目錄：

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

附註：

- a. 當協調佇列管理程式位於與友機伺服器不同的伺服器上時，基本伺服器協調佇列管理程式的連線必須定義為用戶端連線。

在友機伺服器上，若未能將協調佇列管理程式連線定義為 IBM MQ 用戶端連線，將導致任何連接至該協調佇列管理程式的 Managed File Transfer 指令失敗。

連接至協調佇列管理程式的指令範例：**fteListAgents**。

- b. 您不需要建立 IBM MQ 定義，因為當您配置基本伺服器時，就已完成協調佇列管理程式所需要的定義。
2. 發出下列指令來識別指令佇列管理程式：

```
fteSetupCommands -connectionQMGr CSM1
```

指令佇列管理程式不需要任何額外的 IBM MQ 定義。

3. 發出下列指令，以識別友機代理程式佇列管理程式並且建立友機代理程式佇列管理程式：

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

4. 切換至 CSM1AGT1 目錄。

5. 發出下列指令來建立代理程式所需要的 IBM MQ 定義：

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

- a) 使用您偏好的編輯器來開啟 **csm1.txt** 檔案，確認已順利建立所有代理程式需要的定義。

6. 發出下列指令來啟動代理程式：

```
fteStartAgent CSM1AGT1
```

7. 鍵入 **fteListAgents** 來顯示代理程式

您應該會看到類以下列內容的輸出：

```
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
```

註: 如果您尚未在 Managed File Transfer 環境中啟用連線鑑別, 您可以忽略 BFGPR0127W 訊息。

如果您發出 **ftelistAgents** 指令並收到下列訊息: BFGCL0014W: 不存在符合現行選取準則的代理程式。 , 如需進一步資訊, 請參閱第 253 頁的『[ftelistAgents 指令未列出您的 MFT 代理程式時應該如何處理](#)』。

如果其中一個代理程式的狀態是 UNREACHABLE, 請參閱 [第 255 頁的『代理程式顯示為處於 UNKNOWN 狀態時要執行的動作』](#), 以取得進一步資訊。

使用 MFT 來設定 IBM MQ Explorer

此作業協助您將「IBM MQ Explorer」連接至 Managed File Transfer 配置。

程序

1. 啟動「IBM MQ Explorer」。
2. 在左 Navigator 畫面中, 向下捲動並展開資料夾: Managed File Transfer。
您會看到協調佇列管理程式的項目: MFT5
3. 在 MFT5 上按一下滑鼠右鍵, 然後選取**連接**。
 - a) 在出現的下拉功能表中選取 Agents, 確保代理程式 MFT4AGT1 及 CSMAGT1 都處於 Ready 狀態。

下一步

使用「[IBM MQ Explorer](#)」來測試您的範例設定。

使用 IBM MQ Explorer 來測試檔案傳送

這項作業提供一個範例, 說明在您依照前一個主題所說明來設定 IBM MQ Explorer 之後, 如何搭配使用 IBM MQ Explorer 與 Managed File Transfer 來測試檔案傳送。

開始之前

確保您有工作中的系統, 代理程式是 READY 且 IBM MQ Explorer 在工作中。如需相關資訊, 請參閱第 34 頁的『[使用 MFT 來設定 IBM MQ Explorer](#)』。


關於這項作業

決定要用來測試傳送的檔案, 以及要將它複製到哪一個目錄。就此範例而言, 假設使用 C:\temp\mft 目錄中的 test-file.txt 檔案。

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

程序

1. 在 Windows 中啟動 IBM MQ Explorer
2. 在左側導覽器畫面中, 展開此資料夾: Managed File Transfer。
您會看到協調佇列管理程式的項目: MFT5
3. 在 MFT5 上按一下滑鼠右鍵, 然後選取**連接**。
4. 連接之後, 在 MFT5 上按一下滑鼠右鍵, 然後選取**新傳送**。

- a) 使用下拉功能表，以選取代表來源代理程式的 MFT4AGT1 及代表目的地代理程式的 CSMAGT1。
 - b) 按下一步。
 - c) 在下一個視窗上，按一下**新增**。
即會出現一個很大的對話框。左側用於 Source，右側用於 Destination。
5. 在 Source 畫面上：
- a) 選取**文字傳送**，因為檔案是文字檔。
 - b) 選取**瀏覽**，以尋找檔案。
在此案例中，檔案是 C:\temp\mft\test-file.txt。
-  **小心:** 請勿按一下 **確定**，因為您需要完成 Destination 畫面。
6. 在 Destination 畫面上：
- a) 輸入您要提供給目的地上檔案的名稱，例如，test-file.txt。
支援使用相對路徑。完整路徑的上半部是啟動目的地代理程式的使用者 ID 的起始目錄。
 - b) 如果您需要此選項，請選取 **Overwrite files if present**。
 - c) 按一下**確定**。
您選取的檔案即會出現在「**新傳送**」畫面中。
7. 如果已關閉 MFT5 配置功能表，並顯示 +MFT5，請按一下 + 符號來展開此功能表。
8. 保持已選取的 MFT 配置。
接下來，請執行下列程序來檢查傳送的狀態。
9. 在協調佇列管理程式 MFT5 下，按一下**傳送日誌**。
10. 查看 Managed File Transfer - Current Transfer progress panel 中的狀態，緊接在 **傳送日誌** 頂端畫面下方，並等待傳送完成。
如果傳送顯示成功，並出現綠色背景，即表示您已順利完成配置的測試。
如果傳送失敗，並出現紅色背景，則表示發生錯誤。
在大部分情況下，您可以使用上半部**傳送日誌**畫面下方的捲軸來檢視失敗原因摘要。
- a) 如果您無法判定傳送失敗的原因，請在上半部**傳送日誌**畫面中按兩下代表此傳送的項目。
 - b) 在出現的蹦現畫面的左窗格中，選取 XML。
 - c) 捲動全部資訊，以判斷此錯誤的原因。
 - d) 進行必要的更正，然後重新測試傳送。

保護 MFT 安全

Managed File Transfer 在剛安裝且未經修改時具有一種安全層次，適合在受保護的環境下進行測試或評估用途。但是，在正式作業環境中，您必須考量適當地控制誰可以啟動檔案傳送、誰可以讀取及寫入傳送的檔案，以及如何保護檔案的完整性。

相關概念

[第 39 頁的『MFT 沙盤推演』](#)

您可以限制代理程式在傳送過程中可存取的檔案系統區域。代理程式受限的區域稱為沙盤推演。您可以將限制套用至代理程式或要求傳送的使用者。

相關工作

[第 44 頁的『配置 MFT 的 SSL 或 TLS 加密』](#)

您可以搭配使用 SSL 或 TLS 與 IBM MQ Managed File Transfer，以保護代理程式與其代理程式佇列管理程式、指令及其所連接的佇列管理程式，以及拓撲內各種佇列管理程式至佇列管理程式連線之間的通訊安全。

相關參考

[第 296 頁的『限制 MFT 特定資源的群組權限』](#)

您可以為管理 Managed File Transfer 存取控制目的建立兩個安全群組：FTEUSER 及 FTEAGENT，而非針對可能涉及之各種不同物件將權限授與個別使用者。IBM MQ 管理者負責建立及移入這些群組。管理者可以選擇延伸或修改這裡所說明的提議配置。

第 295 頁的『管理 MFT 特定資源的權限』

對於任何檔案傳送要求，Managed File Transfer 代理程式程序都需要對其本端檔案系統具有某種層次的存取權。此外，與代理程式處理程序相關聯的使用者 ID 以及與執行檔案傳送作業的使用者相關聯的使用者 ID，兩者都必須有使用特定 IBM MQ 物件的權限。

第 305 頁的『MFT 存取檔案系統的權限』

對於任何檔案傳送要求，Managed File Transfer 代理程式程序都需要對其本端檔案系統具有某種層次的存取權。

第 36 頁的『MFT 及 IBM MQ 連線鑑別』

連線鑑別可讓佇列管理程式配置成使用提供的使用者 ID 和密碼來鑑別應用程式。如果相關聯的佇列管理程式已啟用安全，且需要認證詳細資料 (使用者 ID 和密碼)，則必須先啟用連線鑑別功能，才能順利建立與佇列管理程式的連線。連線鑑別可以在相容模式或 MQCSP 鑑別模式中執行。

第 306 頁的『commandPath MFT 內容』

使用 commandPath 內容來指定 Managed File Transfer 可以從中執行指令的位置。在設定此內容時請特別小心，因為可以從遠端用戶端系統有效地呼叫其中一個指定的 commandPaths 中的任何指令，從而可以將指令傳送至代理程式。

第 305 頁的『發佈 MFT 代理程式日誌及狀態訊息的權限』

Managed File Transfer 代理程式會發出各種可在協調佇列管理程式上發佈的日誌、進度及狀態訊息。這些訊息的發佈會隨著 IBM MQ 安全模型而有不同，在某些情況下，您可能必須執行進一步的配置才能啟用發佈。

MFT 及 IBM MQ 連線鑑別

連線鑑別可讓佇列管理程式配置成使用提供的使用者 ID 和密碼來鑑別應用程式。如果相關聯的佇列管理程式已啟用安全，且需要認證詳細資料 (使用者 ID 和密碼)，則必須先啟用連線鑑別功能，才能順利建立與佇列管理程式的連線。連線鑑別可以在相容模式或 MQCSP 鑑別模式中執行。

提供認證詳細資料的方法

許多 Managed File Transfer 指令支援下列方法來提供認證詳細資料：

指令行引數所提供的詳細資料。

可以使用 **-mquserid** 和 **-mqpassword** 參數來指定認證詳細資料。如果未提供 **-mqpassword**，則會要求使用者提供未顯示輸入的密碼。

從認證檔提供的詳細資料：**MQMFTCredentials.xml**。

可以在 MQMFTCredentials.xml 檔案中將認證詳細資料預先定義為明碼或模糊文字。

如需在 IBM MQ for Multiplatforms 上設定 MQMFTCredentials.xml 檔案的相關資訊，請參閱 [第 37 頁的『在多平台上配置 MQMFTCredentials.xml』](#)。

如需在 IBM MQ for z/OS 上設定 MQMFTCredentials.xml 檔案的相關資訊，請參閱 [在 z/OS 上配置 MQMFTCredentials.xml](#)。

優先順序

判定認證詳細資料的優先順序如下：

1. 指令行引數。
2. 依相關聯佇列管理程式及執行指令之使用者的 MQMFTCredentials.xml 索引。
3. 依相關聯佇列管理程式的 MQMFTCredentials.xml 索引。
4. 未提供認證詳細資料以容許與舊版 IBM MQ 或 IBM WebSphere MQ 相容的預設舊版相容模式

附註：

- **fteStartAgent** 及 **fteStartLogger** 指令不支援指令行引數 **-mquserid** 或 **-mqpassword**，因此認證詳細資料只能使用 MQMFTCredentials.xml 檔案予以指定。

z/OS

在 z/OS 上，即使使用者的密碼使用小寫字母，密碼也必須為大寫。例如，若使用者的密碼是 "password"，便須輸入為 "PASSWORD"。

相關參考

各個 MFT 指令對應連接的佇列管理程式

Managed File Transfer 拓撲由一些不同元件組成。

MFT 認證檔案格式

MQMFTCredentials.xml 檔案包含機密的使用者 ID 及密碼資訊。MQMFTCredentials.xml 檔案中的元素必須符合 MQMFTCredentials.xsd 綱目。認證檔的安全由使用者負責維護。

在多平台上配置 MQMFTCredentials.xml

如果在已啟用安全的情況下配置 Managed File Transfer (MFT)，則連線鑑別需要所有與佇列管理程式連接的 MFT 指令提供使用者 ID 和密碼認證。同樣地，當連接至資料庫時，可能需要 MFT 日誌程式來指定使用者 ID 和密碼。此認證資訊可以儲存在 MFT 認證檔中。

關於這項作業

MQMFTCredentials.xml 檔案中的元素必須符合 MQMFTCredentials.xsd 綱目。如需 MQMFTCredentials.xml 格式的相關資訊，請參閱 [MFT 認證檔案格式](#)。

您可以在 MQ_INSTALLATION_PATH/mqft/samples/credentials 目錄中找到範例認證檔。

您的協調佇列管理程式、指令佇列管理程式、每個代理程式和每個日誌程式，可分別有一個 MFT 認證檔。或者，您也可以有一個檔案，供拓撲中的所有項目使用。

MFT 認證檔的預設位置如下：

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

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

如果認證檔儲存在不同位置，則您可以使用下列內容來指定指令應該尋找它的位置：

指令類型	內容檔	內容名稱
連接至協調佇列管理程式的指令	coordination.properties	coordinationQMgrAuthenticationCredentialsFile
連接至指令佇列管理程式的指令	connection.properties	connectionQMgrAuthenticationCredentialsFile
連接至代理程式處理程序的指令	agent.properties	agentQMgrAuthenticationCredentialsFile
連接至日誌程式處理程序的指令	logger.properties	loggerQMgrAuthenticationCredentialsFile

指令類型	內容檔	內容名稱
MFT 代理程式	agent.properties	agentQMgrAuthenticationCredentialsFile
MFT 日誌程式	logger.properties	loggerQMgrAuthenticationCredentialsFile

如需哪些指令及處理程序連接至哪個佇列管理程式的詳細資料，請參閱 [哪些 MFT 指令及處理程序連接至哪個佇列管理程式](#)。

因為認證檔包含使用者 ID 和密碼資訊，它需要特殊許可權來防止未獲授權的存取：

Linux → UNIX and Linux

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

Windows → Windows

請確定未啟用繼承，然後移除所有使用者 ID，但執行代理程式或日誌程式且將使用認證檔的使用者 ID 除外。

在的 IBM MQ Explorer Managed File Transfer 外掛程式中，用來連接至 MFT 協調佇列管理程式的認證詳細資料取決於配置類型：

廣域（本端磁碟上的配置）

廣域配置會使用協調和指令內容中指定的認證檔。

本端（於「IBM MQ Explorer」內定義）：

本端配置會使用「IBM MQ Explorer」中相關聯佇列管理程式的連線詳細資料內容。

相關工作

第 38 頁的『啟用 MQCSP 鑑別』

從 IBM MQ 9.0.3 for Continuous Delivery 及 IBM MQ 9.0.0 Fix Pack 2 for Long Term Support，您可以啟用 MQCSP 鑑別模式，以進行與協調佇列管理程式或指令佇列管理程式連接之 IBM MQ Explorer MFT 外掛程式的連線鑑別。還可以對與協調佇列管理程式或指令佇列管理程式進行連線的 Managed File Transfer 代理程式的連線鑑別啟用 MQCSP 鑑別模式。

相關參考

[MFT 認證檔案格式](#)

MQMFTCredentials.xml 檔案包含機密的使用者 ID 及密碼資訊。MQMFTCredentials.xml 檔案中的元素必須符合 MQMFTCredentials.xsd 綱目。認證檔的安全由使用者負責維護。

相關資訊

fte0bfuscate：[加密機密資料](#)

V 9.0.0.2 → V 9.0.3 啟用 MQCSP 鑑別

從 IBM MQ 9.0.3 for Continuous Delivery 及 IBM MQ 9.0.0 Fix Pack 2 for Long Term Support，您可以啟用 MQCSP 鑑別模式，以進行與協調佇列管理程式或指令佇列管理程式連接之 IBM MQ Explorer MFT 外掛程式的連線鑑別。還可以對與協調佇列管理程式或指令佇列管理程式進行連線的 Managed File Transfer 代理程式的連線鑑別啟用 MQCSP 鑑別模式。

關於這項作業

如果使用「IBM MQ Explorer」Managed File Transfer 外掛程式，或者具有的 Managed File Transfer 代理程式是使用 CLIENT 傳送來連接至佇列管理程式並指定密碼，則當指定的密碼長度大於 12 個字元時，該代理程式不會向佇列管理程式鑑別。這是因為程式碼未使用 MQCSP 鑑別，並使用相容模式進行鑑別，這將密碼長度限制為 12 個字元。

從 IBM MQ 9.0.3 和 IBM MQ 9.0.0 Fix Pack 2，您可以停用預設相容模式，並啟用 MQCSP 鑑別模式。

程序

- 若要在「IBM MQ Explorer」中針對協調佇列管理程式或指令佇列管理程式停用相容模式並啟用 MQCSP 鑑別，請完成下列步驟：
 - a) 選取要連接的佇列管理程式。
 - b) 按一下滑鼠右鍵並從蹦現功能表中選取 **連線詳細資料 -> 內容**。

- c) 按一下**使用者 ID** 標籤。
- d) 請確保選取**啟用使用者識別**，並清除**使用者識別相容模式**勾選框。
- 若要針對 Managed File Transfer 代理程式停用相容模式並啟用 MQCSP 鑑別，請將參數 **useMQCSPAAuthentication** 新增至相關使用者的 MQMFTCredentials.xml 檔案，並將它設為 true。
此參數必須設為 true。如果未指定參數，則預設為 false 並且使用相容模式向佇列管理程式鑑別使用者。

下列範例說明如何在 MQMFTCredentials.xml 檔案中設定 **useMQCSPAAuthentication** 參數：

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

相關參考

第 36 頁的『[MFT 及 IBM MQ 連線鑑別](#)』

連線鑑別可讓佇列管理程式配置成使用提供的使用者 ID 和密碼來鑑別應用程式。如果相關聯的佇列管理程式已啟用安全，且需要認證詳細資料 (使用者 ID 和密碼)，則必須先啟用連線鑑別功能，才能順利建立與佇列管理程式的連線。連線鑑別可以在相容模式或 MQCSP 鑑別模式中執行。

第 696 頁的『[MFT 認證檔格式](#)』

MQMFTCredentials.xml 檔案包含機密的使用者 ID 及密碼資訊。MQMFTCredentials.xml 檔案中的元素必須符合 MQMFTCredentials.xsd 綱目。認證檔的安全由使用者負責維護。

相關資訊

[MQCSP 密碼保護](#)

MFT 沙盤推演

您可以限制代理程式在傳送過程中可存取的檔案系統區域。代理程式受限的區域稱為沙盤推演。您可以將限制套用至代理程式或要求傳送的使用者。

代理程式為通訊協定橋接器代理程式或 Connect:Direct 橋接器代理程式時，不支援沙盤推演。對於必須傳送至 IBM MQ 佇列或從該處傳送的代理程式，您無法使用代理程式沙盤推演。

相關參考

第 39 頁的『[使用 MFT 代理程式沙盤推演](#)』

若要對 Managed File Transfer 增加其他安全等級，您可以限制代理程式可存取的檔案系統區域。

第 40 頁的『[使用 MFT 使用者沙盤推演](#)』

您可以根據要求傳送的 MQMD 使用者名稱，限制可來回傳送檔案的檔案系統區域。

使用 MFT 代理程式沙盤推演

若要對 Managed File Transfer 增加其他安全等級，您可以限制代理程式可存取的檔案系統區域。

對於傳送至 IBM MQ 佇列或從該處傳送的代理程式，您無法使用代理程式沙盤推演。以沙盤推演限制 IBM MQ 佇列的存取，可改為透過使用者沙盤推演加以實作，建議所有沙盤推演需求皆利用此解決方案。如需使用者沙盤推演的相關資訊，請參閱第 40 頁的『[使用 MFT 使用者沙盤推演](#)』。

若要啟用代理程式沙盤推演作業，請針對您想要限制的代理程式，將下列內容新增至 agent.properties 檔中：

```
sandboxRoot=[!]restricted_directory_nameseparator...separator[!]restricted_directory_name
```

其中：

- *restricted_directory_name* 是要允許或拒絕的目錄路徑。
- **!** 是選用項目，並指定拒絕 (排除) *restricted_directory_name* 的下列值。如果未指定 **!**，則 *restricted_directory_name* 是容許 (併入) 的路徑。
- *separator* 是平台專用分隔字元。

比方說，例如想要限制 AGENT1 只能存取 /tmp 目錄，但不允許存取子目錄 private，請在屬於 AGENT1 的 agent.properties 檔中設定如下內容：sandboxRoot=/tmp:!/tmp/private。

sandboxRoot 內容詳述於進階代理程式內容中。

通訊協定橋接器代理程式或 Connect:Direct 橋接器代理程式上，皆不支援代理程式及使用者沙盤推演。

在 UNIX、Linux 及 Windows 平台上使用沙盤推演

UNIX 在 UNIX、Linux 及 Windows 平台上，沙盤推演作業會限制 Managed File Transfer Agent 可讀取及寫入的目錄。啟動沙盤推演作業時，Managed File Transfer Agent 可讀取及寫入指定為允許的目錄，以及所指定目錄包含的任何子目錄，除非子目錄在 sandboxRoot 中指定為拒絕。Managed File Transfer 沙盤推演作業的優先順序不高於作業系統安全。啟動 Managed File Transfer Agent 的使用者對任何目錄必須具有適當的作業系統層次存取權，才能夠讀取或寫入該目錄。如果鏈結的目錄位於指定的 sandboxRoot 目錄（及子目錄）外，則不會遵循目錄的符號鏈結。

在 z/OS 上使用沙盤推演

z/OS 在 z/OS 上，沙盤推演作業會限制 Managed File Transfer Agent 可讀取及寫入的資料集名稱限定元。啟動 Managed File Transfer Agent 的使用者對任何相關資料集必須具有正確的作業系統權限。如果以雙引號括住 sandboxRoot 資料集名稱限定元的值，則值遵循一般 z/OS 慣例，且視為完整路徑。如果省略雙引號，則 sandboxRoot 會以現行使用者 ID 為字首。例如，如果您設定 sandboxRoot 內容如下：sandboxRoot=//test，則代理程式可存取下列資料集（採用標準 z/OS 表示法）//username.test.** 在執行時期，如果完整解析的資料集名稱的起始層次不符合 sandboxRoot，則會拒絕傳送要求。

在 IBM i 系統上使用沙盤推演

IBM i 若為 IBM i 系統的整合檔案系統中的檔案，沙盤推演作業會限制 Managed File Transfer Agent 可讀取及寫入的目錄。啟動沙盤推演作業時，Managed File Transfer Agent 可讀取及寫入指定為允許的目錄，以及所指定目錄包含的任何子目錄，除非子目錄在 sandboxRoot 中指定為拒絕。Managed File Transfer 沙盤推演作業的優先順序不高於作業系統安全。啟動 Managed File Transfer Agent 的使用者對任何目錄必須具有適當的作業系統層次存取權，才能夠讀取或寫入該目錄。如果鏈結的目錄位於指定的 sandboxRoot 目錄（及子目錄）外，則不會遵循目錄的符號鏈結。

相關參考

第 43 頁的『對萬用字元傳送進行其他檢查』

從 Continuous Delivery 中的 IBM MQ 9.0.1，以及 Long Term Support 中的 IBM MQ 9.0.0 Fix Pack 1，如果已使用使用者或代理程式沙盤推演來配置代理程式，以限制代理程式可以來回傳送檔案的位置，您可以指定對該代理程式的萬用字元傳送進行其他檢查。

第 39 頁的『使用 MFT 代理程式沙盤推演』

若要對 Managed File Transfer 增加其他安全等級，您可以限制代理程式可存取的檔案系統區域。

第 463 頁的『MFT agent.properties 檔案』

每一個 Managed File Transfer Agent 都有自己的內容檔 agent.properties，其中必須包含代理程式用來連接至其佇列管理程式的資訊。agent.properties 檔案也可以包含變更代理程式行為的內容。

使用 MFT 使用者沙盤推演

您可以根據要求傳送的 MQMD 使用者名稱，限制可來回傳送檔案的檔案系統區域。

當代理程式是通訊協定橋接器代理程式或 Connect:Direct 橋接器代理程式時，不支援使用者沙盤推演。

若要啟用使用者沙盤推演，請針對您想要限制的代理程式，將下列內容新增至 agent.properties 檔案中：

```
userSandboxes=true
```

有了這項內容並設為 true 時，代理程式就會使用 MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name/UserSandboxes.xml 檔案中的資訊，來決定要求傳送的使用者可以存取檔案系統的哪些部分。

UserSandboxes.xml XML 由包含零個以上 <sandbox> 元素的 <agent> 元素組成。這些元素說明哪些規則適用於哪些使用者。<sandbox> 元素的 user 屬性是一種型樣，用來比對要求的 MQMD 使用者。

代理程式會定期重新載入 UserSandboxes.xml 檔案，因此對該檔案所做的任何有效變更，都會影響代理程式的行為。預設重新載入間隔是 30 秒。透過指定 agent.properties 檔案中的代理程式內容 xmlConfigReloadInterval，即可變更此間隔。

如果指定 userPattern="regex" 屬性或值，則 user 屬性會解譯為 Java 正規表示式。如需相關資訊，請參閱第 554 頁的『MFT 所使用的正規表示式』。

如果您未指定 userPattern="regex" 屬性或值，則 user 屬性會解譯為具有下列萬用字元的型樣：

- 星號 (*)，代表零個以上字元
- 問號 (?)，正好代表一個字元

會依照該檔案中列出 <sandbox> 元素的順序來執行比對。只會使用第一個相符項，而忽略該檔案中所有後續可能的相符項。如果該檔案中指定的 <sandbox> 元素全部都不符合與傳送要求訊息相關聯的 MQMD 使用者，則傳送時將無法存取檔案系統。在 MQMD 使用者名稱與 user 屬性之間找到相符項之後，此相符項將識別 <sandbox> 元素內套用至傳送的一組規則。這組規則用來決定傳送過程中可以讀取或寫入的檔案或資料集。

每一組規則可指定 <read> 元素來識別可讀取的檔案，以及指定 <write> 元素來識別可寫入的檔案。如果您在一組規則中省略了 <read> 或 <write> 元素，即假設與該組規則相關聯的使用者不得執行任何讀取或寫入。

註：在 UserSandboxes.xml 檔中，<read> 元素必須在 <write> 元素之前，<include> 元素必須在 <exclude> 元素之前。

每一個 <read> 或 <write> 元素包含一個以上型樣，用來決定沙盤推演中是否存在檔案以及是否可以進行傳送。請使用 <include> 及 <exclude> 元素來指定這些型樣。<include> 或 <exclude> 元素的 name 屬性指定要比對的型樣。選用的 type 屬性指定名稱值是檔案或佇列型樣。如果未指定 type 屬性，代理程式會將此型樣視為檔案或目錄路徑型樣。例如：

```
<tns:read>
  <tns:include name="/home/user/**"/>
  <tns:include name="USER.**" type="queue"/>
  <tns:exclude name="/home/user/private/**"/>
</tns:read>
```

代理程式使用 <include> 及 <exclude> name 型樣，來決定是否可以讀取或寫入檔案、資料集或佇列。如果標準檔案路徑、資料集或佇列名稱符合至少其中一個包括的型樣，而排除的型樣一個都不符合，則容許作業。使用 <include> 及 <exclude> 元素的 name 屬性指定的型樣，會使用適用於代理程式執行所在平台的路徑分隔字元及使用慣例。如果您指定相對檔案路徑，則解析的路徑將相對於代理程式的 transferRoot 內容。

指定佇列限制時，支援 QUEUE@QUEUEMANAGER 的語法，且規則如下：

- 如果項目中遺漏 at 字元 (@)，則將該型樣視為可以在任何佇列管理程式上存取的佇列名稱。比方說，例如型樣是 name，則將它視為與 name@** 相同。
- 如果 at 字元 (@) 是項目中的第一個字元，則將該型樣視為佇列管理程式名稱，並可存取該佇列管理程式上的所有佇列。比方說，例如型樣是 @name，則將它視為與 **@name 相同。

當您指定下列萬用字元作為 <include> 及 <exclude> 元素的 name 屬性的一部分時，它們具有特殊意義：


在目錄名稱中，或在資料集名稱或佇列名稱的限定元中，單一星號符合零或多個字元。

?

問號符合目錄名稱中的正好一個字元，或符合資料集名稱或佇列名稱的限定元。

兩個星號字元符合零或多個目錄名稱，或資料集名稱或佇列名稱中零或多個限定元。同時，結尾是路徑分隔字元的路徑有隱含的 "***" 新增至路徑結尾。因此，/home/user/ 與 /home/user/** 相同。

例如：

- `/**/test/**` 符合其路徑中有 `test` 目錄的任何檔案
- `/test/file?` 符合 `/test` 目錄內以字串 `file` 開頭且後面接著任何單一字元的任何檔案
- `c:\test*.txt` 符合 `c:\test` 目錄內具有 `.txt` 副檔名的任何檔案
- `c:\test***.txt` 符合 `c:\test` 目錄內任何檔案，或其中一個子目錄具有 `.txt` 副檔名
-  `// 'TEST.*.DATA'` 符合其第一個限定元為 `TEST`、有任何第二個限定元，且第三個限定元為 `DATA` 的任何資料集。
- `*@QM1` 符合佇列管理程式 `QM1` 中具有單一限定元的任何佇列。
- `TEST.*.QUEUE@QM1` 符合佇列管理程式 `QM1` 上其第一個限定元為 `TEST`、有任何第二個限定元，且第三個限定元為 `QUEUE` 的任何佇列。
- `**@QM1` 符合佇列管理程式 `QM1` 上的任何佇列。

符號鏈結

您必須完整解析 `UserSandboxes.xml` 檔案中檔案路徑所使用的任何符號鏈結，方法是在 `<include>` 及 `<exclude>` 元素中指定固定鏈結。例如，如果您具有 `/var` 對映至 `/SYSTEM/var` 的符號鏈結，則必須將此路徑指定為 `<tns:include name="/SYSTEM/var"/>`，否則預期的傳送會因使用者沙盤推演安全錯誤而失敗。

範例

此範例顯示如何容許具有 MQMD 使用者名稱 `guest` 的使用者，透過將下列 `<sandbox>` 元素新增至 `AGENT_JUPITER` 配置目錄中的 `UserSandboxes.xml` 檔案，從代理程式 `AGENT_JUPITER` 執行所在系統上的 `/home/user/public` 目錄或其任何子目錄中傳送任何檔案：

```
<?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>
```

範例

此範例說明如何容許具有 MQMD 使用者名稱 `account` 且後面跟著單一數字的任何使用者（例如 `account4`）完成下列動作：

- 傳送來自 `/home/account` 目錄或其任何子目錄的任何檔案（不包括代理程式 `AGENT_SATURN` 執行所在系統上的 `/home/account/private` 目錄）
- 將任何檔案傳送至代理程式 `AGENT_SATURN` 執行所在系統上的 `/home/account/output` 目錄或其任何子目錄
- 從本端佇列管理程式上以字首 `ACCOUNT.` 開頭的佇列中讀取訊息，除非它以 `ACCOUNT.PRIVATE.` 開頭（即在第二層具有 `PRIVATE`）。
- 將資料傳送至任何佇列管理程式上開頭為字首 `ACCOUNT.OUTPUT.` 的佇列上。

若要容許具有 MQMD 使用者名稱 `account` 的使用者完成這些動作，請將下列 `<sandbox>` 元素新增至 `AGENT_SATURN` 配置目錄中的 `UserSandboxes.xml` 檔案：

```
<?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>

```

相關參考

第 43 頁的『對萬用字元傳送進行其他檢查』

從 Continuous Delivery 中的 IBM MQ 9.0.1，以及 Long Term Support 中的 IBM MQ 9.0.0 Fix Pack 1，如果已使用使用者或代理程式沙盤推演來配置代理程式，以限制代理程式可以來回傳送檔案的位置，您可以指定對該代理程式的萬用字元傳送進行其他檢查。

第 40 頁的『使用 MFT 使用者沙盤推演』

您可以根據要求傳送的 MQMD 使用者名稱，限制可來回傳送檔案的檔案系統區域。

第 463 頁的『MFT agent.properties 檔案』

每一個 Managed File Transfer Agent 都有自己的內容檔 agent.properties，其中必須包含代理程式用來連接至其佇列管理程式的資訊。agent.properties 檔案也可以包含變更代理程式行為的內容。

V 9.0.1 V 9.0.0.1 對萬用字元傳送進行其他檢查

從 Continuous Delivery 中的 IBM MQ 9.0.1，以及 Long Term Support 中的 IBM MQ 9.0.0 Fix Pack 1，如果已使用使用者或代理程式沙盤推演來配置代理程式，以限制代理程式可以來回傳送檔案的位置，您可以指定對該代理程式的萬用字元傳送進行其他檢查。

additionalWildcardSandboxChecking 內容

若要對萬用字元傳送啟用其他檢查，請針對您想要檢查的代理程式，將下列內容新增至 agent.properties 檔案中：

```
additionalWildcardSandboxChecking=true
```

如果將此內容設為 true，而且代理程式所提出的傳送要求，嘗試讀取的位置與萬用字元相符，但位於定義給檔案的沙盤推演外部，則傳送會失敗。如果在一個傳送要求中進行多個傳送，而且其中一個要求由於嘗試讀取沙盤推演外部的位址而失敗，則整個傳送都會失敗。如果檢查失敗，則錯誤訊息會提供失敗原因。

如果 additionalWildcardSandboxChecking 內容已從代理程式的 agent.properties 檔案中省略或者設為 false，則不會對該代理程式的萬用字元傳送進行其他檢查。

萬用字元檢查的錯誤訊息

從 Continuous Delivery 中的 IBM MQ 9.0.1，以及 Long Term Support 中的 IBM MQ 9.0.0 Fix Pack 1，當對所配置沙盤推演位置以外的位址發出萬用字元傳送要求時所報告的訊息已變更。

如果傳送要求中的萬用字元檔案路徑位於所要求沙盤推演外部，則會出現下列訊息：

```
BFGSS0077E: 嘗試讀取檔案路徑 path 遭拒絕。
此檔案路徑位於限制性傳送沙盤推演外部。
```

如果多重傳送要求中的一項傳送包含一個萬用字元傳送要求，但其中的路徑位於限制性沙盤推演外部，則會出現下列訊息：

```
BFGSS0078E: Attempt to read file path: path has being ignored as another transfer
受管理傳送中的另一個傳送項目嘗試在限制性傳送沙盤推演外部進行讀取。
```

如果檔案位於限制性沙盤推演外部，則會出現下列訊息：

BFGSS0079E: 嘗試讀取檔案 *file path* 遭拒絕。
該檔案位於受限傳送沙盤推演外。

在多重傳送要求中，如果一個萬用字元傳送要求由於其他萬用字元傳送要求而遭忽略，則會出現下列訊息：

BFGSS0080E: 嘗試讀取檔案: *檔案路徑* 已被忽略，因為另一個傳送受管理傳送中的另一個傳送項目嘗試在限制性傳送沙盤推演外部進行讀取。

如果單一檔案傳送不包含萬用字元，則當傳送涉及的檔案位於沙盤推演外部時，報告的訊息與舊版相同：

由於下列訊息而失敗: BFGI00056E: 嘗試讀取檔案 "*FILE*" 遭拒絕。
該檔案位於受限傳送沙盤推演外。

相關參考

第 40 頁的『使用 MFT 使用者沙盤推演』

您可以根據要求傳送的 MQMD 使用者名稱，限制可來回傳送檔案的檔案系統區域。

第 39 頁的『使用 MFT 代理程式沙盤推演』

若要對 Managed File Transfer 增加其他安全等級，您可以限制代理程式可存取的檔案系統區域。

第 463 頁的『MFT agent.properties 檔案』

每一個 Managed File Transfer Agent 都有自己的內容檔 `agent.properties`，其中必須包含代理程式用來連接至其佇列管理程式的資訊。`agent.properties` 檔案也可以包含變更代理程式行為的內容。

配置 MFT 的 SSL 或 TLS 加密

您可以搭配使用 SSL 或 TLS 與 IBM MQ Managed File Transfer，以保護代理程式與其代理程式佇列管理程式、指令及其所連接的佇列管理程式，以及拓撲內各種佇列管理程式至佇列管理程式連線之間的通訊安全。

開始之前

您可以使用 SSL 或 TLS 加密來加密流經 IBM MQ Managed File Transfer 拓撲的訊息。其中包括：

- 在代理程式與其代理程式佇列管理程式之間傳遞的訊息。
- 指令及其所連接之佇列管理程式的訊息。
- 在拓撲內的代理程式佇列管理程式、指令佇列管理程式及協調佇列管理程式之間流動的內部訊息。

關於這項作業

如需搭配使用 SSL 與 IBM MQ 的一般資訊，請參閱[使用 SSL 或 TLS](#)。就 IBM MQ 而言，Managed File Transfer 是標準 Java 用戶端應用程式。

遵循這些步驟對 Managed File Transfer 使用 SSL：

程序

1. 建立信任儲存庫檔案及（選擇性地）金鑰儲存庫檔（這些檔案可以是相同檔案）。如果不需要用戶端鑑別（亦即，通道上的 `SSLCAUTH=OPTIONAL`），則不需要提供金鑰儲存庫。您只需要信任儲存庫來鑑別佇列管理程式的憑證。
用於為信任儲存庫和金鑰儲存庫建立憑證的金鑰演算法必須是 RSA，才能使用 IBM MQ。
2. 設定 IBM MQ 佇列管理程式來使用 SSL。
如需利用「IBM MQ Explorer」設定佇列管理程式以使用 SSL 的相關資訊，請參閱在佇列管理程式上配置 SSL。
3. 將信任儲存庫檔案及金鑰儲存庫檔（如果有的話）儲存在適當位置。建議位置是 `config_directory/coordination_qmgr/agents/agent_name` 目錄。
4. 在適當的 Managed File Transfer 內容檔中設定每一個已啟用 SSL 佇列管理程式所需的 SSL 內容。每一組內容各自參照個別的佇列管理程式（代理程式、協調及指令），但一個佇列管理程式可能扮演其中兩個以上的角色。

需要 **CipherSpec** 或 **CipherSuite** 內容其中一個，否則用戶端會嘗試不使用 SSL 而直接連接。同時提供 **CipherSpec** 或 **CipherSuite** 內容，因為 IBM MQ 與 Java 之間的術語差異。Managed File Transfer 會接受任一內容並執行必要的轉換，所以您不需要同時設定這兩個內容。如果同時指定 **CipherSpec** 或 **CipherSuite** 內容，則優先採用 **CipherSpec**。

PeerName 是選用內容。此內容可設為您想要連接的佇列管理程式的「識別名稱」。Managed File Transfer 會拒絕連線至「識別名稱」不符的不正確 SSL 伺服器。

將 **SslTrustStore** 及 **SslKeyStore** 內容設為指向信任儲存庫及金鑰儲存庫檔的檔案名稱。如果是針對已執行的代理程式設定這些內容，請停止並重新啟動代理程式，並以 SSL 模式重新連接。

內容檔包含純文字密碼，請考慮設定適當的檔案系統權限。

如需 SSL 內容的相關資訊，請參閱第 508 頁的『MFT 的 SSL 內容』。

5. 如果代理程式佇列管理程式使用 SSL，則建立代理程式時您無法提供必要的詳細資料。請使用下列步驟來建立代理程式：
 - a) 使用 **fteCreateAgent** 指令建立代理程式。您會收到警告，表示無法將代理程式存在的事實發佈至代理程式協調佇列管理程式。
 - b) 編輯前一個步驟所建立的 `agent.properties` 檔來新增 SSL 資訊。代理程式順利啟動後，即再次嘗試發佈。
6. 如果在變更 `agent.properties` 檔案或 `coordination.properties` 檔案中的 SSL 內容時，正在執行「IBM MQ 探險家」的代理程式或實例，則必須重新啟動代理程式或 IBM MQ Explorer。

相關參考

第 463 頁的『MFT `agent.properties` 檔案』

每一個 Managed File Transfer Agent 都有自己的內容檔 `agent.properties`，其中必須包含代理程式用來連接至其佇列管理程式的資訊。`agent.properties` 檔案也可以包含變更代理程式行為的內容。

在用戶端模式下使用通道鑑別連接至佇列管理程式

IBM WebSphere MQ 7.1 推出了通道鑑別記錄功能，可更精確地控制通道層次上的存取。這項行為變更意味著新建立的 IBM WebSphere MQ 7.1 版或更新版本佇列管理程式，依預設會拒絕來自 Managed File Transfer 元件的用戶端連線。

如需通道鑑別的相關資訊，請參閱[通道鑑別記錄](#)。

如果 Managed File Transfer 所使用的 SVRCONN 的通道鑑別配置指定了非特許 MCAUSER ID，您必須授與佇列管理程式、佇列及主題的特定權限記錄，以讓 Managed File Transfer Agent 及指令正常運作。請使用 MQSC 指令 SET CHLAUTH 或 PCF 指令設定通道鑑別記錄，來建立、修改或移除通道鑑別記錄。對於您要連接至 IBM WebSphere MQ 7.1 或更新版本佇列管理程式的所有 Managed File Transfer 代理程式，您可以設定要用於所有代理程式的 MCAUSER ID，或為每一個代理程式設定個別的 MCAUSER ID。

請為每一個 MCAUSER ID 授與下列權限：

- 佇列管理程式所需的權限記錄：

- 連接
- `setid`
- `inq`

- 佇列所需的權限記錄。

對於所有代理程式特定佇列（即下列清單中以 `agent_name` 結尾的佇列名稱），您必須為要使用用戶端連線連接至 IBM WebSphere MQ 7.1 版或更新版本佇列管理程式的每個代理程式，建立這些佇列權限記錄。

- `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)`

- 主題所需的權限記錄：

- `sub, pub (SYSTEM.FTE)`

- 檔案傳送所需的權限記錄。

如果來源及目的地代理程式具有個別的 MCAUSER ID，請在來源及目的地代理程式佇列中建立權限記錄。

例如，如果來源代理程式的 MCAUSER ID 為 **user1**，目的地代理程式 MCAUSER ID 為 **user2**，請為代理程式使用者設定下列權限：

代理程式使用者	佇列	所需的權限
user1	SYSTEM.FTE.DATA.destination_agent_name	放置
user1	SYSTEM.FTE.COMMAND.destination_agent_name	放置
user2	SYSTEM.FTE.REPLY.source_agent_name	放置
user2	SYSTEM.FTE.COMMAND.source_agent_name	放置

在 Connect:Direct 橋接器代理程式與 Connect:Direct 節點之間配置 SSL 或 TLS

透過建立金鑰儲存庫與信任儲存庫，以及設定 Connect:Direct 橋接器代理程式內容檔中的內容，即可將 Connect:Direct 橋接器代理程式與 Connect:Direct 節點，配置為透過 SSL 通訊協定彼此連接。

關於這項作業

下列步驟包含一些指示來取得透過憑證管理中心簽署的金鑰。如果您不使用憑證管理中心，則可產生自簽憑證。如需產生自簽憑證的相關資訊，請參閱在 [UNIX 和 Windows 系統上使用 SSL 或 TLS](#)。

下列步驟包含為 Connect:Direct 橋接器代理程式，建立新的金鑰儲存庫及信任儲存庫的指示。如果 Connect:Direct 橋接器代理程式，已經具有用於安全連接至 IBM MQ 佇列管理程式的金鑰儲存庫及信任儲存庫，則可在安全連接至 Connect:Direct 節點時使用現有的金鑰儲存庫及信任儲存庫。如需相關資訊，請參閱第 44 頁的『配置 MFT 的 SSL 或 TLS 加密』。

程序

對於 Connect:Direct 節點，請完成下列步驟：

1. 產生用於 Connect:Direct 節點的金鑰及已簽章的憑證。
您可以使用 IBM MQ 隨附的 IBM Key Management 工具來執行此動作。如需相關資訊，請參閱 [使用 SSL 或 TLS](#)。
2. 向憑證管理中心傳送簽署金鑰的要求。您會收到回報的憑證。
3. 建立包含憑證管理中心公開金鑰的文字檔，例如 /test/ssl/certs/CAcert。
4. 在 Connect:Direct 節點上安裝 Secure+ Option。
如果該節點已經存在，您可以透過下列方式來安裝 Secure+ Option：再次執行安裝程式，指定現有安裝的位置，然後選擇僅安裝 Secure+ Option。
5. 建立新的文字檔，例如 /test/ssl/cd/keyCertFile/node_name.txt。
6. 將您從憑證管理中心收到的憑證，以及位於 /test/ssl/cd/privateKeys/node_name.key 中的私密金鑰，複製到該文字檔中。

/test/ssl/cd/keyCertFile/node_name.txt 的內容必須採用下列格式：

```
-----BEGIN CERTIFICATE-----
MIIcZwCCAgigAwIBAgIBGjANBgkqhkiG9w0BAQUFADBeMQswCQYDVQQGEwJHcWJES
MBAGA1UECBMJSjGfTcHN0aXJlMR4wDgYDVQQHEwdIdXJzZbGV5M0QwYyYDVQKKEwNj
Qk0xOjAMBGNVBAStBU1RSVBUMQswCQYDVQQDEwJDTAeFw0xMTAzMDEwMDExNDZa
Fw0yMTAyMjYxNjIwNDZaMFAXCzAJBgNVBAYTAkdCMRIwEAYDVQQIEw1IYW1wc2hp
cmUxDDAKBgNVBAoTA01CTTEOMAwGA1UECxMFTVFVGVUxZzANBgNVBAMTBmJpbmJh
ZzZCBnzANBgkqhkiG9w0BAQEFAA0BjQAwYkCgYEAvgP1QIk1U9yppSKD1Xo0Do1yk
EyMFxB0UpZRiDVxjoSEC0vtWncJ199e+Vc4UpNybdyBu+Nkd1MNoF4QxeQcLAFj
WnhakqCiQ+JIAD5Aurhn1wChe0MV3kjA84GKH/10SVqt1984mu/1dyS819XcfSSn
c00MsK1KbneVSCIV2XECaWAAa7MHkwCQYDVR0TBAlwADAsBg1ghkgBhvCAQ0E
HxYdTB3l1b1NTTCBHZW51cmF0ZWQgQ2VydG1maWNhdGUwHQYDVR00BBYEFNXMIpSc
csBXUniW4A3UrzNCRsv3MB8GA1UdIwQYMBaAFDXY81mj41Vz5+FVAoQb++cns+B4
MA0GCSqGSIB3DQEBBQUAA4GBAFc7k1Xa4pGKYgwcxKpE3ZF6FNwy4vBXS216/ja
8h/v18+iv010CL8t0Z0KSU95fyZLz0PKnCH7v+ItFSE3CIiEk9D1z2U6W091ICwn
```

```

17PL72Tdfal3kabwHYVf17IVcuL+VZsZ3HjLggP2qH09ZuJPspeT9+AxFVMLiaAb
8eHw
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,64A02DA15B6B6EF9

57kqxL0J/gRU0IQ6hVK2YN13B4E1jAi1gSme0I5ZpEIG8CHXISKB7/0cke2FTqsV
lvI99QyCsDwoMnt5fj51v7aPmVeS60b0m+U1Gre8B/Zel8JvJ204K2Uh72rDCXE
5e6eFxsDUM207sQDy20euBVELJtM2k0kL1R0doQOS1U3XQNgJw/t3ZIx5hPXWEQT
rjRQ064BEhb+PzzxPF8uwzZ9IrkUK9BJ/UUnqC60dBR87IeA4pnJD1Jvb2ML7EN9Z
5Y+50hTKI80GvBvWX04fHyvIX5aslwhBoArXIS1AtNTTrptPvoaP1zyIAeZ60Cvo/
SFo+A2UhmTEJe0JaZG2XZ3H495fAw/EHmjehzIACwukQ9nSIETgu4A1+CV64RJED
aYBCM8UjaAkbZDH5gn7+eBov0ssXAXWdyJBVhU0jXjvAj/e1h+kcSF1hax5D//AI
66nRMZzboSxNqkjcVd8wfDwP+bEjDzUaaarJTS71IFeLLw7eJ8MNAkMGicDkycL0
EPBU9X5QnHKLK0fYHN/1WgUk8qt3UytFXXfzTXGF3EbsWbBupkT5e5+1YcX80VZ6
sHFPN1H1uCNy/riUcBy9iviVeodX8Iom0chSy05DK18bwZNjYtUP+CtYHNFU5BaD
I+1uU0AeJ+wjQYKT1WaeIGZ3VxuNITJu18y5qDTXXfX7vxM50oWxa6U5+AYuGUMg
/itPZmUmNrhjT7ghT6i1IQ0aBowXXKJB1Mmq/6BQXN2IhkD9ys2qrvM1hdi5nAf
egmdiG50l0LnBRqWbFR+DykpAhK4SaDi2F52Uxovw3Lhiw8dQP71zQ==
-----END RSA PRIVATE KEY-----

```

7. 啟動 Secure+ Admin Tool。

- 在 Linux 或 UNIX 系統上，執行 **spadmin.sh** 指令。
- 在 Windows 系統上，按一下**開始 > 程式集 > Sterling Commerce Connect:Direct > CD Secure+ Admin Tool**

CD Secure+ Admin Tool 即會啟動。

8. 在 CD Secure+ Admin Tool 中，按兩下 **.Local** 行以編輯主要 SSL 或 TLS 設定。

- 視您使用的通訊協定而定，選取**啟用 SSL 通訊協定**或**啟用 TLS 通訊協定**。
- 選取**停用置換**。
- 至少選取一個密碼組合。
- 如果您想要雙向鑑別，請將 **啟用用戶端鑑別** 的值變更為 Yes。
- 在**授信主要憑證**欄位中，輸入憑證管理中心的公開憑證檔路徑 (/test/ssl/certs/CAcert)。
- 在**金鑰憑證檔**欄位中，輸入您所建立的檔案路徑 (/test/ssl/cd/keyCertFile/node_name.txt)。

9. 按兩下 **.Client** 行以編輯主要 SSL 或 TLS 設定。

- 視您使用的通訊協定而定，選取**啟用 SSL 通訊協定**或**啟用 TLS 通訊協定**。
- 選取**停用置換**。

對於 Connect:Direct 橋接器代理程式，請執行下列步驟：

10. 建立信任儲存庫。您可以透過先建立後刪除虛擬金鑰來執行此作業。

您可以使用下列指令：

```
keytool -genkey -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
```

```
keytool -delete -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
```

11. 將憑證管理中心的公開憑證匯入信任儲存庫。

您可以使用下列指令：

```
keytool -import -trustcacerts -alias myCA
-file /test/ssl/certs/CAcert
-keystore /test/ssl/fte/stores/truststore.jks
```

12. 編輯 Connect:Direct 橋接器代理程式內容檔。

將下列行併入檔案中的任何位置：

```
cdNodeProtocol=protocol
```

```
cdNodeTruststore=/test/ssl/fte/stores/truststore.jks
cdNodeTruststorePassword=password
```

在此步驟的範例中，*protocol* 是您使用的通訊協定（SSL 或 TLS），*password* 是您在建立信任儲存庫時指定的密碼。

13. 如果要進行雙向鑑別，請為 Connect:Direct 橋接器代理程式建立金鑰及憑證。

a) 建立金鑰儲存庫及金鑰。

您可以使用下列指令：

```
keytool -genkey -keyalg RSA -alias agent_name
-keystore /test/ssl/fte/stores/keystore.jks
-storepass password -validity 365
```

b) 產生簽署要求。

您可以使用下列指令：

```
keytool -certreq -v -alias agent_name
-keystore /test/ssl/fte/stores/keystore.jks -storepass password
-file /test/ssl/fte/requests/agent_name.request
```

c) 將您從之前步驟中收到的憑證匯入金鑰儲存庫。此憑證必須採用 x.509 格式。

您可以使用下列指令：

```
keytool -import -keystore /test/ssl/fte/stores/keystore.jks
-storepass password -file certificate_file_path
```

d) 編輯 Connect:Direct 橋接器代理程式內容檔。

將下列行併入檔案中的任何位置：

```
cdNodeKeystore=/test/ssl/fte/stores/keystore.jks
cdNodeKeystorePassword=password
```

在此步驟的範例中，*password* 是您在建立金鑰儲存庫時指定的密碼。

相關工作

[第 118 頁的『配置 Connect:Direct 橋接器』](#)

配置 Connect:Direct 橋接器，以在 Managed File Transfer 網路與 Connect:Direct 網路之間傳送檔案。

Connect:Direct 橋接器的元件為 Connect:Direct 節點，以及專門用來與該節點通訊的 Managed File Transfer 代理程式。此代理程式稱為 Connect:Direct 橋接器代理程式。

配置 MFT

您可以在安裝之後配置 Managed File Transfer 的特性。

相關概念

[第 214 頁的『利用使用者結束程式自訂 MFT』](#)

您可以使用專屬程式（即所謂的使用者結束常式）來自訂 Managed File Transfer 的特性。

[第 206 頁的『MFT 回復及重新啟動』](#)

如果代理程式或佇列管理程式因故無法使用，例如因為電源或網路故障，在這些情況下，Managed File Transfer 會回復，如下所示：

[第 229 頁的『使用 MFT 的提示及要訣』](#)

這裡提供了一些建議，可協助您充分使用 Managed File Transfer。

[第 128 頁的『管理 MFT』](#)

使用 Managed File Transfer 指令來管理 Managed File Transfer。您也可以使用「IBM MQ Explorer」來執行部分管理作業。

相關工作

[第 141 頁的『監視 MFT 資源』](#)

您可以監視 Managed File Transfer 資源；例如，佇列或目錄。滿足此資源的條件時，資源監視器會啟動作業，例如檔案傳送。您可以在 IBM MQ Explorer 的 Managed File Transfer 外掛程式中使用 **fteCreateMonitor** 指令或「監視器」視圖來建立資源監視器。

第 37 頁的『在多平台上配置 MQMFTCredentials.xml』

如果在已啟用安全的情況下配置 Managed File Transfer (MFT)，則連線鑑別需要所有與佇列管理程式連接的 MFT 指令提供使用者 ID 和密碼認證。同樣地，當連接至資料庫時，可能需要 MFT 日誌程式來指定使用者 ID 和密碼。此認證資訊可以儲存在 MFT 認證檔中。

相關參考

第 313 頁的『MFT 指令』

列出了所有 Managed File Transfer 指令，及其詳細說明的鏈結。

第 35 頁的『保護 MFT 安全』

Managed File Transfer 在剛安裝且未經修改時具有一種安全層次，適合在受保護的環境下進行測試或評估用途。但是，在正式作業環境中，您必須考量適當地控制誰可以啟動檔案傳送、誰可以讀取及寫入傳送的檔案，以及如何保護檔案的完整性。

第 463 頁的『MFT agent.properties 檔案』

每一個 Managed File Transfer Agent 都有自己的內容檔 `agent.properties`，其中必須包含代理程式用來連接至其佇列管理程式的資訊。`agent.properties` 檔案也可以包含變更代理程式行為的內容。

第 211 頁的『搭配使用 Apache Ant 與 MFT』

您可以利用 Managed File Transfer 提供的作業，將檔案傳送功能整合到 Apache Ant 工具中。

第 227 頁的『疑難排解 MFT』

使用下列參照資訊來協助您診斷 Managed File Transfer 中的錯誤：

多平台上的 MFT 配置選項

Managed File Transfer 提供了一組內容檔，這些檔案包含有關您的設定的金鑰資訊，是作業所必要的。這些內容檔位於安裝產品時所定義的配置目錄中。

您可以具有多組配置選項，每一組配置選項包含一組目錄及內容檔。這些內容檔中定義的值用作所有 Managed File Transfer 指令的預設參數，除非您明確在指令行上指定不同的值。

若要變更您正在使用的預設配置選項集，您可以使用 **fteChangeDefaultConfigurationOptions** 指令。若要變更您正在用於個別指令的配置選項集，您可以將 **-p** 參數與任何 Managed File Transfer 指令搭配使用。

配置選項集的名稱即是協調佇列管理程式的名稱，建議不要變更。您仍可變更配置選項集的名稱，但必須變更 `config` 和 `logs` 目錄的名稱。在下列範例中，配置選項集的名稱會以 `coordination_qmgr_name` 顯示。

配置選項目錄結構

當您配置產品時，會在配置目錄的下列結構中建立目錄及內容檔。您也可以使用下列指令來變更這些目錄及內容檔：**fteSetupCoordination**、**fteSetupCommands**、**fteChangeDefaultConfiguration** 及 **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` 目錄是配置選項目錄。配置目錄中可能存在多個配置選項目錄。`agent_name` 目錄是代理程式目錄。除了包含 `agent.properties` 檔案外，此目錄包含 `exits` 目錄（使用者結束程式常式及 `fteCreateBridgeAgent` 與 `fteCreateCDAgent` 指令產生的各種 XML 檔案的預設位置）。配置選項集的 `agents` 目錄中可能具有多個代理程式目錄。

內容檔

installation.properties

`installation.properties` 檔案指定預設配置選項集的名稱。此項目將 Managed File Transfer 指向一組結構化目錄以及包含要使用之配置的內容檔。一組配置選項的名稱通常是關聯的協調佇列管理程式的名稱。如需 `installation.properties` 檔案的相關資訊，請參閱 [第 453 頁的『MFT installation.properties 檔』](#)。

coordination.properties

`coordination.properties` 檔案指定協調佇列管理程式的連線詳細資料。因為數個 Managed File Transfer 安裝可能共用相同的協調佇列管理程式，所以您可以使用共用磁碟機上一般 `coordination.properties` 檔案的符號鏈結。如需 `coordination.properties` 檔案的相關資訊，請參閱 [第 456 頁的『MFT coordination.properties 檔』](#)。

command.properties

`command.properties` 檔指定當您發出指令時要連接的指令佇列管理程式，以及 Managed File Transfer 聯絡該佇列管理程式所需的資訊。如需 `command.properties` 檔案的相關資訊，請參閱 [第 460 頁的『MFT command.properties 檔』](#)。

agent.properties

每一個 Managed File Transfer Agent 都有自己的內容檔 `agent.properties`，其中必須包含代理程式用來連接至其佇列管理程式的資訊。`agent.properties` 檔案也可以包含變更代理程式行為的內容。如需 `agent.properties` 檔案的相關資訊，請參閱 [第 463 頁的『MFT agent.properties 檔案』](#)。

logger.properties

`logger.properties` 檔案指定日誌程式的配置內容。如需 `logger.properties` 檔案的相關資訊，請參閱 [第 99 頁的『MFT 日誌程式配置內容』](#)。

內容檔及字碼頁

由於 Java 的限制，所有 Managed File Transfer 內容檔的內容必須維持為美式英文。如果在非美式英文系統上編輯內容檔，您必須使用 Unicode ESC 序列。

z/OS 上的 MFT 配置選項

z/OS 上的 Managed File Transfer 配置選項與分散式平台的選項相同。

如需 [Multiplatforms](#) 上配置選項的相關資訊，請參閱 [第 49 頁的『多平台上的 MFT 配置選項』](#)。

在 z/OS 上，配置位置由環境變數 `BFG_DATA` 定義。如果 `BFG_DATA` 所參照的「UNIX 系統服務」目錄下尚無配置存在，則會由 MFT 指令 PDSE 程式庫資料集的 `BFGCUSTOM JCL Script` 產生建立配置所需的工作。然後，配置將會在您執行這些產生的工作時建立。建立配置時，需使用對可存取的現有目錄進行參照的 `BFG_DATA`。

您也可以使用在多平台及 z/OS 上皆可使用的相同 `fte` 指令，來建立及維護配置。如需 `fte` 指令清單，請參閱 [第 313 頁的『MFT 指令』](#)。

V 9.0.1 配置 Redistributable Managed File Transfer Agent

配置選擇性提供的 Redistributable Managed File Transfer Agent，以連接到現有的 IBM MQ 基礎架構，讓使用者能夠在沒有安裝 IBM MQ 的情況下傳送檔案，以取得 Managed File Transfer 功能。

開始之前

Redistributable Managed File Transfer Agent 提供 Managed File Transfer 的功能，但下列情況例外：

- 不支援將模式連線連結到協調、指令及代理程式佇列管理程式，您必須使用用戶端模式連線。當您發出指令時，當您使用安裝為 IBM MQ 一部分的 Managed File Transfer 時，必須提供選用參數：佇列管理程式主機、埠、名稱及通道名稱。
- 不包括下列指令：

- fteCreateCDAgent.cmd
- fteCreateLogger.cmd
- fteDeleteLogger.cmd
- fteMigrateLogger.cmd
- fteSetLoggerTraceLevel.cmd
- fteShowLoggerDetails.cmd
- fteStartLogger.cmd
- fteStopLogger.cmd

如需可用指令的完整清單，請參閱已安裝的 MFT 指令集。

- 不支援 Managed File Transfer Connect:Direct。
- 不包括「IBM MQ Explorer」。

Windows 您必須在系統上安裝下列 Microsoft 程式庫，才能使用 Redistributable Managed File Transfer Agent：

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

可從 Microsoft 取得這些程式庫。請參閱[最新支援的 Visual C++ 下載項目](#)。

關於這項作業

您希望允許使用者在區域環境與現有的 IBM MQ 基礎架構之間傳送檔案，且不需要安裝 IBM MQ 就可以這麼做。請遵循下列步驟來下載 Redistributable Managed File Transfer Agent，並進行解壓縮：

程序

1. 從 [IBM Fix Central](#) 下載 Redistributable Managed File Transfer Agent 套件。

a) 為您的作業系統選擇套件：

- **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) 識別您要在其中解壓縮套件的目錄，例如：

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

2. 將所下載套件的內容解壓縮：

- **Windows** 使用 Windows 「檔案總管」工具來解壓縮。
- **Linux** 如下所示進行解壓縮：

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

接著，

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

會建立下列目錄：

- **Windows** > **Linux** bin: 包含所有必要的 MFT 指令
- **Windows** bin64: 包含 Windows 64 位元 OS 支援所需的必要程式庫
- **Windows** > **Linux** java: 包含 IBM JRE 和 IBM MQ 程式庫
- **Windows** > **Linux** licenses: 包含授權檔
- **Windows** > **Linux** mqft: 包含 Ant 支援及核心 MFT 功能支援所需的 ant 和 lib 目錄
- **Windows** > **Linux** swtag: 包含 swidtag 檔案，授權管理程式需要此檔案來識別機器上的安裝

下一步

您已準備好配置 MFT 代理程式。如需後續步驟，請參閱下列主題：[第 52 頁的『建立起始配置』](#)。

相關參考

[第 270 頁的『配置可重新配送的 MFT 代理程式時可能發生的錯誤』](#)
配置 Redistributable Managed File Transfer Agent 時的錯誤訊息

相關資訊

[視訊：如何安裝及配置 Redistributable Managed File Transfer Agent \(YouTube\)](#)

建立起始配置

您可以配置 Managed File Transfer Agent 以連接至現有 IBM MQ 配置。

開始之前

V 9.0.1 請務必下載 Redistributable Managed File Transfer Agent 套件的內容，並進行解壓縮。如需相關資訊，請參閱 [第 50 頁的『配置 Redistributable Managed File Transfer Agent』](#)。

關於這項作業

請建立環境，以便能夠使用指令來設定與 IBM MQ 伺服器上所執行之佇列管理程式的連線功能。

程序

1. 為 Redistributable Managed File Transfer Agent 建立環境。

當您執行 **fteCreateEnvironment** 指令時，會建立 MFT 資料目錄，其中包含 MFT 代理程式的配置資訊。確保您所在的 bin 目錄是在您將所下載的 MFT 代理程式可重新配送元件解壓縮時建立的目錄。請執行下列指令：

- **Windows**

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

- **Linux**

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

此指令採用一個選用的參數：

-d

在其中建立、儲存及維護 MFT 配置之資料路徑的位置。如果使用者執行 **fteCreateEnvironment** 而未指定資料位置，則會在解壓縮 Redistributable Managed File Transfer Agent 的位置建立 mftdata 目錄。

註: 如果可轉散發的代理程式將以 Windows 服務形式來執行，則需要在系統環境中設定 **BFG_DATA** 環境變數，服務才能運作。

您還可以設定環境變數 **BFG_DATA** 來搭配資料路徑位置：

```
BFG_DATA=Datapath location
```

在建立、啟動及停止代理程式或其他任何指令之前，您必須確保 **BFG_DATA** 變數是設為正確的資料路徑位置。

2. 設定 IBM MQ 連線功能。

a) 使用 **fteSetupCoordination** 指令來設定協調佇列管理程式。

fteSetupCoordination 指令會建立協調佇列管理程式所需的設定，以及進一步配置所需的目錄。Redistributable Managed File Transfer Agent 是以用戶端模式運作，因此您必須在此指令中提供額外參數來避免錯誤，因為連結模式不受支援。

```
fteSetupCoordination -coordinationQMGr PRMFTDEM02  
-coordinationQMGrHost 9.121.59.233 -coordinationQMGrPort 3002  
-coordinationQMGrChannel SYSTEM.DEF.SVRCONN
```

如需使用 **fteSetupCoordination** 指令的詳細資料和步驟，請參閱第 433 頁的『[fteSetupCoordination](#)』。如需如何配置協調佇列管理程式的相關資訊，請參閱第 82 頁的『[配置 MFT 的協調佇列管理程式](#)』。

b) 建立及設定指令佇列管理程式：

```
fteSetupCommands -p PRMFTDEM02 -connectionQMGrHost 9.121.59.233  
-connectionQMGrPort 3002 -connectionQMGrChannel SYSTEM.DEF.SVRCONN  
-connectionQMGr PRMFTDEM02 -f
```

如需使用 **fteSetupCommands** 指令的詳細資料和步驟，請參閱第 432 頁的『[fteSetupCommands: 建立 MFT command.properties 檔](#)』。

3. 為端點建立 MFT 代理程式定義。

```
fteCreateAgent -p PRMFTDEM02 -agentQMGrHost 9.121.59.233  
-agentQMGrPort 3002 -agentQMGrChannel SYSTEM.DEF.SVRCONN  
-agentName AGENT.TRI.BANK -agentQMGr PRMFTDEM02 -f
```

如需使用 **fteCreateAgent** 指令配置代理程式與代理程式佇列管理程式的相關資訊，請參閱第 326 頁的『[fteCreateAgent \(建立 MFT 代理程式\)](#)』。

在每一個代理程式的 [第 53 頁的『2』](#) 和 [第 53 頁的『3』](#) 步驟中，您會在代理程式佇列管理程式上建立佇列和主題定義。

4. 啟動代理程式，然後您就可以開始傳送檔案。

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

您可以透過執行下列指令來驗證代理程式的狀態：

```
fteListAgents
```

如需使用 **fteListAgents** 指令的詳細資料，請參閱第 398 頁的『[fteListAgents: 列出協調佇列管理程式的 MFT 代理程式](#)』。

相關概念

[第 48 頁的『配置 MFT』](#)

您可以在安裝之後配置 Managed File Transfer 的特性。

[第 49 頁的『多平台上的 MFT 配置選項』](#)

Managed File Transfer 提供了一組內容檔，這些檔案包含有關您的設定的金鑰資訊，是作業所必要的。這些內容檔位於安裝產品時所定義的配置目錄中。

相關參考

[第 367 頁的『fteCreateTransfer: 啟動新的檔案傳送』](#)

fteCreateTransfer 指令從指令行建立並啟動新的檔案傳送。此指令可立即啟動檔案傳送、將檔案傳送排定在未來的時間和日期、將排定的傳送重複一次以上，以及根據特定條件觸發檔案傳送。

z/OS 建立 MFT 代理程式或日誌程式指令資料集

您可以從 Managed File Transfer 指令範本資料集，為特定的 Managed File Transfer Agent 或 Managed File Transfer Logger 建立指令的 PDSE 資料集，以進行特定協調。

關於這項作業

請完成下列步驟：

程序

1. 複製 MFT 指令範本 PDSE 程式庫資料集 SBFGCMD5。

z/OS SBFGCMD5 必須複製到新的程式庫，例如 *prefix.agent.JCL_*。進行下列取代作業時，您可以使用經過更新的 SBFGCMD5(BFGCOPY) 成員：

- 將 *++supplied-library++* 取代為 SBFGCMD5 PDSE 的完整名稱。
 - **z/OS** 將 *++service-library++* 取代為新的 MFT 指令 PDSE 程式庫資料集的完整名稱。 *+service-library+* 是建立的代理程式或日誌程式服務的輸出資料集。
2. 針對新的 MFT 指令 PDSE 程式庫資料集，編輯成員 BFGCUSTM，這是用來為代理程式或日誌程式自訂指令的 JCL Script。每一個變數都會以 *++variable name++* 的格式指定，您必須將其取代為必要值。如需各個 JCL 變數的說明，請參閱第 66 頁的『z/OS JCL 變數』。BFGSTDIN DD 陳述式會定義三個種類的變數：Variables、Properties 及 Environment。此陳述式採用下列格式：

```
[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
```

變數會定義每一個指令所需的一組設定及環境變數。

內容定義會置換 MFT 配置內容。您可以視需要新增代理程式或日誌程式內容，為您的環境自訂代理程式或日誌程式。如需所有內容的清單，請參閱第 75 頁的『配置內容檔』。提供這項機制，是為了免去您存取 MFT 配置內容檔（以「UNIX 系統服務」檔案形式進行維護）的需要。

Environment 會定義所有額外需要的自訂環境變數。

3. 提交新的 MFT 指令 PDSE 程式庫資料集的工作 BFGCUSTM。此工作會產生適用於代理程式或日誌程式的 JCL 指令集，作為 PDSE 的新成員。如需指令的完整清單，請參閱第 69 頁的『z/OS 代理程式及日誌程式指令 JCL Script』。

工作 BFGCUSTM 會對包括 DISP=OLD 之 DD 陳述式的 JCL 所在的程式庫進行更新。您必須在提交之後結束編輯器，讓工作執行。

請檢查輸出工作日誌，以確認 JCL Script 已順利執行。如果執行失敗，請予以更正，然後重新提交 BFGCUSTM 工作。

BFGCUSTM JCL Script 也會在必要時更新「UNIX 系統服務」MFT 配置內容檔，使檔案保持一致。如果 CoordinationQMGr 內容定義的配置不存在，則會輸出警告訊息，而且您必須執行產生的 BFGCFGR 和 BFGCMCR 工作，才能建立配置內容檔。必須針對代理程式執行 BFGAGCR，而針對日誌程式編輯執行

BFGLGCRS。如果指定的配置已存在，該配置將會以 BFTCUSTOM JCL Script 中定義的任何內容進行更新。

相關概念

第 50 頁的『z/OS 上的 MFT 配置選項』

z/OS 上的 Managed File Transfer 配置選項與分散式平台的選項相同。

相關工作

第 65 頁的『在 z/OS 上更新現有 MFT 代理程式或日誌程式指令資料集』

您可以更新從 Managed File Transfer 指令範本資料集建立的 Managed File Transfer 指令 PDSE 程式庫資料集。

z/OS 配置 Managed File Transfer for z/OS

Managed File Transfer for z/OS 需要經過自訂，元件才能正常運作。

關於這項作業

您需要：

1. 編輯 PDSE 成員，以指定配置資料
2. 定義協調佇列管理程式。
3. 定義指令佇列管理程式
4. 配置一或多個代理程式
5. 選用項目：配置日誌程式作業，以將資料儲存在 Db2 中

下列主題詳細說明所需執行之作業的順序。

相關概念

第 55 頁的『檢閱 MFT 配置』

在開始之前，您需要先檢閱您的系統配置。

相關資訊

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

z/OS 檢閱 MFT 配置

在開始之前，您需要先檢閱您的系統配置。

Managed File Transfer (MFT) 需要一或多個佇列管理程式，才能為每個已定義的 MFT 配置扮演下列角色：

- 協調佇列管理程式，用於維護配置中各個代理程式狀態的相關資訊，這些資訊發佈至協調程式的相關主題中。
- 一或多個指令或連線佇列管理程式，作為 MFT 指令的 IBM MQ 網路進入點。
- 一或多個代理程式佇列管理程式，其負責在 MFT 代理程式與 IBM MQ 網路之間提供通訊。

上述每一個角色可由不同的佇列管理程式執行，您也可以將這些角色結合在一個最簡單的配置中，讓單一佇列管理程式執行所有角色。

如果要在現有的 MFT 環境中新增 z/OS 佇列管理程式，您需要在 z/OS 佇列管理程式與此配置的其他佇列管理程式之間定義連線功能。您可以透過手動定義傳輸佇列或利用叢集作業來達到此目的。

每個 MFT 代理程式與單一佇列管理程式進行通訊。如果有多個代理程式與同一個佇列管理程式通訊，則代理程式佇列管理程式將為每個代理程式定義多個佇列：

- 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

請注意，您可以定義通用安全設定檔，在其中使用諸如 SYSTEM.FTE.COMMAND.* 之類的設定檔，也可以為每個代理程式定義特定的設定檔。

相關概念

第 56 頁的『開始之前』

Managed File Transfer (MFT) 配置會使用 UNIX 系統服務 (USS) 和 PDSE 資料集中的檔案。

開始之前

Managed File Transfer (MFT) 配置會使用 UNIX 系統服務 (USS) 和 PDSE 資料集中的檔案。

大部分的配置和作業都可使用 PDSE 中的 JCL 來完成，而您必須熟悉 USS 環境中的運作方式。

您可以透過 ISPF 或在您的工作站上透過指令使用 Telnet 類型階段作業（例如 Telnet Putty 或 SSH）來存取 OMVS。

如果您透過 ISPF 使用 OMVS，則可以使用標準 ISPF 編輯器，並瀏覽指令 **oedit** 和 **obrowse**。

您必須熟悉下列 USS 指令

指令	函數
ls -ltr path	列出路徑中之檔案的相關資訊
ls -ltrd directory	列出指定目錄（而不是目錄中的檔案）的相關資訊。
find path -name xxx	在路徑目錄中搜尋名為 xxx 的檔案。xxx 區分大小寫，且可以是 *zzz
chmod xxx path	變更檔案存取權限
df -k path	報告檔案系統中剩餘的可用空間。-k 會以 KB 為單位報告可用空間。
du -kt path	報告路徑下的目錄大小。報告的大小以 KB 為單位
oedit filename	在 OMVS 中編輯檔案
obrowse filename	瀏覽檔名

檢閱下表中的項目，並以您的企業適用的項目完成表格。您在編輯成員 **BFGCUSTM** 時將需要這些值。

姓名	範例資料	Comments
BFG_JAVA_HOME	/java/java71_bit64_GA/J7.1_64/	
BFG_GROUP_NAME	MQM	
LIBRARY	SCEN.FTE.JCL	MFT PDSE 的名稱。您需要每個代理程式或日誌程式作業的副本。
TMPDIR	/tmp	讀取和寫入暫存檔可存取的 USS 路徑。

表 4: 成員 BFGCUSTM 所需的參數 (繼續)		
姓名	範例資料	Comments
SERVICE_TYPE	AGENT 或 LOGGER	
名稱	AGENT1	
BFG_PROD	/var/ibm/wmqmft	
BFG_DATA		視需要填寫
BFG_JVM_PROPERTIES		視需要填寫
QMGR	MQPV	
MQ_PATH	/mqm/V8R0M0	
MQ_LANG	E	
Db2_HLQ	SYS2.Db2.V10	
FTE_CONFIG		用於移轉
CREDENTIAL_PATH		用於移轉
DB_PROPS_PATH		用於移轉
BFG_WTO	YES	用以取得 syslog 上的 MFT 訊息。
ADMIN_JOB1		工作卡。所有工作皆以相同的 JCL 卡產生。
OUTPUT_CLASS	*	
JOBCARD1		這是長時間執行作業、代理程式和日誌程式的工作卡。
PATH	bin:/usr/bin:/usr/sbin	
armELEMTYPE	如果使用 ARM, 請使用 ARM 原則中指定的 ARM ELEMTYPE。例如, 為代理程式使用 armELEMTYPE=SYSBFGAG, 或為日誌程式使用 armELEMTYPE=SYSBFGLG。如果未使用 ARM, 請將此參數設為空白; 例如 armELEMTYPE=	
armELEMENT	如果使用 ARM, 請為此代理程式或日誌程式使用 ARM 原則中指定的 ARM ELEMENT 值。如果未使用 ARM, 請將此參數設為空白; 例如 armELEMENT=	
coordinationQMgr	MQPV	必要配置

此外, 您必須檢閱下列變數並提供必要的值:

- coordinationQMgrHost=
- coordinationQMgrPort=
- coordinationQMgrChannel=
- connectionQMgr=
- connectionQMgrHost=
- connectionQMgrPort=

- connectionQMgrChannel=

這些內容是 AGENT 或 LOGGER 通用的。

註：「主機」、「埠」和「通道」是用戶端連線的必要項目，且對於本端機器上的連結連線不應保留為空白。

相關概念

第 58 頁的『要檢查的項目』

確定您有足夠的磁碟空間、用來儲存資料的目錄，且必要的檔案皆存在。

第 62 頁的『編輯成員 BFGCUSTM』

您必須先編輯成員 BFGCUSTM，並為您的企業所使用的參數輸入值，才能執行工作。

z/OS 要檢查的項目

確定您有足夠的磁碟空間、用來儲存資料的目錄，且必要的檔案皆存在。

確認您有足夠的磁碟空間

確認您要用來儲存配置專用檔案的檔案系統上有足夠的磁碟空間可供使用。

如果代理程式追蹤已啟用，依預設將可使用 100 MB 的磁碟空間。

配置檔本身並不大，只有幾 KB 而已。

如果您預計要使用兩個代理程式和一個日誌程式，則至少需要 300 MB。您可以使用指令 **df -k path**，其中，**path** 是安裝專用檔案的位置。這會指出以 KB 為單位的可用空間和空間總計。

300 MB 是 307,200 KB，因此您至少應提供 310,000 KB

建立並檢查用來儲存 Managed File Transfer 資料的目錄

您必須有用來儲存 Managed File Transfer (MFT) 資料的目錄。

確認您在檔案系統 **df -k /var** 中有足夠的空間。此檔案系統至少應有 310,000 KB 可供使用。

如果您尚未建立此檔案系統，請使用 **mkdir** 指令；例如 **mkdir /var/mft**。

使用指令 **ls -ltrd /var/mft** 顯示使用者對此目錄具有的許可權。

如果擁有者或群組不正確，請使用指令 **chown owner:group /var/mft**。

如果群組的權限不正確，請使用下列指令為擁有者和群組提供讀取、寫入和執行權限。請注意，下列指令也會為所有使用者提供讀取和執行權限：**chmod 775 /var/mft**。

確認檔案存取，且您具有其存取權。

對您在自訂期間所將使用的檔案使用 **ls -ltr** 指令。例如：

```
ls -ltrd /java/java71_bit64_GA/J7.1_64/bin
```

會產生

```
drwxr-xr-x  4 SYSTASK TSOUSER   8192 Nov 15 2013 /java/java71_bit64_GA/J7.1_64/bin
```

其中，**drwxr-xr-x** 表示

d

這是目錄。

rwX

擁有者 SYSTASK 具有目錄的讀取、寫入和執行權限。

r-x

群組 TSOUSER 中的人員可讀取和執行目錄中的檔案。

r-x

通用存取，即任何人皆可讀取或執行目錄中的檔案。

檢查以下指定的檔案：

路徑	使用者執行配置所需的存取權
BFG_JAVA_HOME	讀取和執行
/tmp	讀取和寫入
BFG_PROD	讀取
BFG_DATA	寫入
MQ_PATH	讀取

相關概念

第 56 頁的『開始之前』

Managed File Transfer (MFT) 配置會使用 UNIX 系統服務 (USS) 和 PDSE 資料集中的檔案。

第 59 頁的『一般 MFT for z/OS 配置』

不同 Managed File Transfer 配置的概觀

一般 MFT for z/OS 配置

不同 Managed File Transfer 配置的概觀

Managed File Transfer 會使用連結至佇列管理程式的代理程式來傳送資料。

MFT 可以使用多個佇列管理程式：

- 一個以上用於傳送資料的佇列管理程式。
- 一個用於發出要求的指令佇列管理程式。例如，啟動傳送的要求傳送到此佇列管理程式之後，相關聯的指令將遞送至 MFT 代理程式。
- 一個用於管理工作的協調佇列管理程式。

有三個一般 Managed File Transfer (MFT) 配置：

1. 單一佇列管理程式與一個以上使用本端連線的代理程式。這可能是用來將資料集的內容放置到 IBM MQ 佇列中。
2. 單一佇列管理程式與一個在分散式機器上使用用戶端連結的 MFT 用戶端。
3. 由通道連接的兩個佇列管理程式，且在每部機器上各有一個以上的代理程式。這些代理程式可以是用戶端或本端連結。

請注意下列幾點：

1. MFT 是以 Java 撰寫，其中包含用於配置及操作 MFT 的部分 Shell Script 及 JCL。
2. Db2 狀態及活動可以記載並儲存在 Db2 表格中。
3. 配置 MFT 的人員必須熟悉「UNIX 系統服務 (USS)」。
- 例如：
 - 包含名稱類似於 /u/userID/myfile.txt2 之檔案的目錄結構
 - USS 指令，例如：
 - cd** (切換目錄)
 - ls** (列出)
 - chmod** (變更檔案許可權)
 - chown** (變更檔案所有權或可以存取檔案或目錄的群組)
4. 在 USS 中需要下列產品，才能配置及執行 MFT：
 - Java；例如， /java/java71_bit64_GA/J7.1_64/

- IBM MQ 第 800 版，例如 /mqm/V8R0M03。
- Db2 JDBC 程式庫（如果您要使用 Db2 儲存狀態及歷程）；例如 /db2/db2v10/jdbc/lib

您需要協調佇列管理程式。不過，您可以使用相同的佇列管理程式來執行代理程式、處理指令和進行協調。如果您使用多個佇列管理程式，您必須擇一作為協調程式。

檢查您的 IBM MQ 連線功能

如果您具有現有的 MFT 協調佇列管理程式，您在執行配置的佇列管理程式與協調和指令佇列管理程式之間必須有連線。

在 z/OS 上配置 MQMFTCredentials.xml

如果在已啟用安全的情況下配置 Managed File Transfer (MFT)，則連線鑑別需要所有 MFT 代理程式以及連接至佇列管理程式的指令，才能提供使用者 ID 和密碼認證。

同樣地，當連接至資料庫時，可能需要 MFT 日誌程式來指定使用者 ID 和密碼。

此認證資訊可以儲存在 MFT 認證檔中。請注意，認證檔是選用的，不過，在自訂環境之前更容易定義您需要的一或多個檔案。

此外，如果您有認證檔，則會收到較少的警告訊息。警告訊息會通知您 MFT 認為佇列管理程式安全已關閉，因此您不應提供鑑別詳細資料。

您可以在 MQ_INSTALLATION_PATH/mqft/samples/credentials 目錄中找到範例認證檔。

以下是 MQMFTCredentials.xml 檔的範例：

```
<?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>
```

使用者 ID 為 ADMIN 的工作在需要連接到佇列管理程式 MQPH 時，將會傳遞使用者 ID JOHNDOEH，並使用密碼 cXXXX。

如果工作由任何其他使用者 ID 執行，並連接 MQPH，該工作將會傳遞使用者 ID NONEH 和密碼 yXXXX。

MQMFTCredentials.xml 檔的預設位置是使用者在 z/OS Unix 系統服務 (USS) 上的起始目錄。也可以將檔案儲存在 USS 上的不同位置，或儲存在分割資料集內的成員中。

如果認證檔儲存在不同位置，則您可以使用下列內容來指定指令應該尋找它的位置：

表 6：這些內容定義各種指令的 MQMFTCredentials.xml 檔案位置。		
指令類型	內容檔	內容名稱
連接至協調佇列管理程式的指令	coordination.properties	coordinationQMGrAuthenticationCredentialsFile
連接至指令佇列管理程式的指令	connection.properties	connectionQMGrAuthenticationCredentialsFile
連接至代理程式處理程序的指令	agent.properties	agentQMGrAuthenticationCredentialsFile
連接至日誌程式處理程序的指令	logger.properties	loggerQMGrAuthenticationCredentialsFile

表 7: : 定義代理程式及日誌程式處理程序的 *MQMFTCredentials.xml* 檔案位置的內容。

指令類型	內容檔	內容名稱
MFT 代理程式	agent.properties	agentQMGrAuthenticationCredentialsFile
MFT 日誌程式	logger.properties	loggerQMGrAuthenticationCredentialsFile

如需哪些指令及處理程序連接至哪個佇列管理程式的詳細資料，請參閱 [哪些 MFT 指令及處理程序連接至哪個佇列管理程式](#)。

若要在分割的資料集內建立認證檔，請執行下列步驟：

- 使用格式 VB 和邏輯記錄長度 (Lrecl) 200 來建立 PDSE。
- 在資料集內建立成員、記下資料集和成員，然後將下列程式碼新增至成員：

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

您可以使用安全產品 (例如 RACF) 來保護認證檔，但執行 Managed File Transfer 指令及管理代理程式和日誌程式處理程序的使用者 ID 需要此檔案的讀取權。

您可以在成員 BFGCROBS 中使用 JCL，以遮蔽此檔案中的資訊。這會取走檔案，並將 IBM MQ 使用者 ID 和密碼加密。例如，成員 BFGCROBS 取走以下一行

```
<tns:qmgr name="MQPI" user="JOHND0E2" mqUserId="JOHND0E1" mqPassword="yXXXX" />
```

並建立

```
<tns:qmgr mqPasswordCipher="e977c61e9b9c363c" mqUserIdCipher="c394c5887867157c"
name="MQPI" user="JOHND0E2"/>
```

如果您想要將使用者 ID 保留在 IBM MQ 使用者 ID 對映，您可以在檔案中新增註解。例如

```
<!-- name="MQPI" user="ADMIN" mqUserId="JOHND0E1" -->
```

這些註解並不會因為遮蔽程序而變更。

請注意，內容只是遮蔽而已，並未嚴密加密。您應限制哪些使用者 ID 可存取檔案。

相關工作

在 [Multiplatforms](#) 上配置 *MQMFTCredentials.xml*

如果在已啟用安全的情況下配置 Managed File Transfer (MFT)，則連線鑑別需要所有與佇列管理程式連接的 MFT 指令提供使用者 ID 和密碼認證。同樣地，當連接至資料庫時，可能需要 MFT 日誌程式來指定使用者 ID 和密碼。此認證資訊可以儲存在 MFT 認證檔中。

複製 SBFGCMDs 以建立 JCL 程式庫

您必須為每個代理程式和日誌程式建立一個 JCL 程式庫，JCL 包含用來建立和執行代理程式或日誌程式的配置和工作。

編輯並執行 BFGCOPY 成員，為每個代理程式和日誌程式建立 IBM 提供的 SBFGCMDs 程式庫的副本。

此程式庫會用來定義代理程式或日誌程式的配置，而在自訂之後，會包含可用來建立必要 Managed File Transfer 配置和代理程式或日誌程式的工作。

您可以在此程序中建立成員 BFGCUSTM。

註：如果您熟悉 USS 指令，您可以用您在其他平台上使用的相同指令來配置 z/OS。

相關概念

[第 59 頁的『一般 MFT for z/OS 配置』](#)

不同 Managed File Transfer 配置的概觀

第 62 頁的『編輯成員 BFGCUSTM』

您必須先編輯成員 BFGCUSTM，並為您的企業所使用的參數輸入值，才能執行工作。

編輯成員 BFGCUSTM

您必須先編輯成員 BFGCUSTM，並為您的企業所使用的參數輸入值，才能執行工作。

請參閱成員 BFGCUSTM 所需的參數，以取得需要特定值的參數清單。

此外，您必須檢閱下列變數並提供必要的值：

- coordinationQMGrHost=
- coordinationQMGrPort=
- coordinationQMGrChannel=
- connectionQMGr=
- connectionQMGrHost=
- connectionQMGrPort=
- connectionQMGrChannel=

這些內容是 AGENT 或 LOGGER 通用的。

註：「主機」、「埠」和「通道」是用戶端連線的必要項目，且對於本端機器上的連結連線不應保留為空白。

如果這是您 Managed File Transfer 環境中的第一個佇列管理程式，且您要讓協調、指令和執行中的代理程式使用相同的佇列管理程式，請將值設定為本端佇列管理程式名稱。

```
coordinationQMGr=MQPV  
connectionQMGr=MQPV
```

其中，MQPV 是您的本端佇列管理程式名稱。

提交工作，以更新 PDSE，並在指定的路徑下建立目錄結構。

請注意，這項工作需要專用權，因此在工作執行時請停止使用 PSDE。

提示：當您提交工作 BFGCUSTM 時，此工作一律會取代所有的 JCL 檔案。您應將您變更的每個成員重新命名。

相關概念

第 56 頁的『開始之前』

Managed File Transfer (MFT) 配置會使用 UNIX 系統服務 (USS) 和 PDSE 資料集中的檔案。

第 64 頁的『建立代理程式』

您必須複製 PDSE 以建立代理程式專用的 PDSE，例如 *user.MFT.AGENT1*。如果先前的代理程式或日誌程式配置中存在 PDSE，請加以複製。如果這是您的第一個配置，請複製隨附於 MFT 的 PDSE。

定義協調佇列管理程式

Managed File Transfer 要求將佇列管理程式建立為協調佇列管理程式。

根據您所選擇的配置，此佇列管理程式會位於本端 MVS 系統或另一部機器上。如果是前者，其連線將會是連結連線；如果是後者，則是用戶端連線。

在您順利執行配置步驟後，PDSE 中會有經配置的成員。

成員 BFGCFGR 會定義協調佇列管理程式，而這項工作會：

1. 在 Managed File Transfer (MFT) 目錄中建立目錄結構，並建立配置檔。
2. 執行 CSQUTIL 以定義 IBM MQ 資源。

如果協調佇列管理程式位於遠端機器上，則此工作會失敗。

成員 BCFCFCR 會在 USS 中建立檔案，並建立 MQ 定義。這項工作會：

1. 建立 MFT 主題、
2. 建立 MFT 佇列
3. 將 `NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)` 變更為 `NAMES(SYSTEM.BROKER.DEFAULT.STREAM, SYSTEM.BROKER.ADMIN.STREAM, SYSTEM.FTE)`
4. 執行 `ALTER QMGR PSMODE(ENABLED)`

在執行變更之前，會發出 `DISPLAY NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)` 指令。如果您的 `NAMLIST` 不是預設的，您應變更名稱清單，將 `SYSTEM.FTE` 新增至您的名稱清單

以您自己的字首（例如 `CCPCFCR`）重新命名成員 `BCFCFCR`，因為重新自訂此檔案會加以取代。

插入認證檔的名稱，以編輯這個重新命名過的成員。例如：

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

儲存並提交工作。請注意，如果您需要重新提交工作，您必須新增 `-f` 選項。

這項工作執行時，會列出它所建立的 IBM MQ 資源。您必須保護這些資源。

```
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)
```

相關工作

第 63 頁的『定義指令佇列管理程式』

您可以使用與協調和指令佇列管理程式相同的佇列管理程式，或者建立新的指令佇列管理程式。

定義指令佇列管理程式

您可以使用與協調和指令佇列管理程式相同的佇列管理程式，或者建立新的指令佇列管理程式。

關於這項作業

您必須具有指令佇列管理程式，但您的協調和指令佇列管理程式可以使用相同的佇列管理程式。否則，您必須建立新的指令佇列管理程式。它可以位於與協調佇列管理程式相同的機器上，但不一定要如此。

程序

1. 使用專屬字首（例如 `CCPCMCR`）來重新命名成員 `BFGCMCR`。
您必須重新命名 `BFGCMCR`，因為重新自訂此檔案會取代此檔案。
2. 插入認證檔的名稱，以編輯已重新命名的成員。

例如：

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

3. 儲存並提交工作。
請注意，如果您需要重新提交工作，您必須新增 `-f` 選項。

此佇列管理程式會用於 **ftePingAgent** 之類的指令。

4. 檢閱此成員、加以提交，並檢閱輸出。

下一步

如需如何建立代理程式的相關資訊，請參閱第 64 頁的『建立代理程式』。

相關概念

第 62 頁的『定義協調佇列管理程式』

Managed File Transfer 要求將佇列管理程式建立為協調佇列管理程式。

相關工作

第 37 頁的『在多平台上配置 MQMFTCredentials.xml』

如果在已啟用安全的情況下配置 Managed File Transfer (MFT)，則連線鑑別需要所有與佇列管理程式連接的 MFT 指令提供使用者 ID 和密碼認證。同樣地，當連接至資料庫時，可能需要 MFT 日誌程式來指定使用者 ID 和密碼。此認證資訊可以儲存在 MFT 認證檔中。

相關參考

第 696 頁的『MFT 認證檔格式』

MQMFTCredentials.xml 檔案包含機密的使用者 ID 及密碼資訊。MQMFTCredentials.xml 檔案中的元素必須符合 MQMFTCredentials.xsd 綱目。認證檔的安全由使用者負責維護。

z/OS 建立代理程式

您必須複製 PDSE 以建立代理程式專用的 PDSE，例如 *user.MFT.AGENT1*。如果先前的代理程式或日誌程式配置中存在 PDSE，請加以複製。如果這是您的第一個配置，請複製隨附於 MFT 的 PDSE。

檢閱成員 BFGCUSTM，如果您需要使用不同的認證檔，請加以建立。

有許多內容仍與第 62 頁的『編輯成員 BFGCUSTM』中詳述的自訂作業相同。

您必須變更：

- //SYSEXEC DD DSN=SCEN.FTE.JCL.AGENT1
- LIBRARY，以符合代理程式 PDSE
- SERVICE_TYPE=AGENT
- NAME，使其成為代理程式（符合 PDSE）JOB CARD 的名稱
- 變更 BFG_JVM_PROPERTIES="-Xmx1024M"

提交此工作；請切記，此工作需要資料集的互斥存取權。

代理程式的工作全都會有 BFGAG* 格式的名稱

重新命名成員 BFGAGCR。此工作會更新 Managed File Transfer 目錄中的檔案，並使用 CSQUTIL 在本端佇列管理程式中建立代理程式專用佇列。指定認證檔的名稱，例如 -credentialsFile //'SCEN.FTE.JCL.VB(CREDOLD)。如果未指定此名稱，啟動代理程式的工作就不會使用認證檔。

檢查輸出以確定程序順利執行。

提示：將 *agent.properties* 檔案的路徑名稱從工作的輸出複製到代理程式在 PDSE 中的成員。

例如，將 /u/userid/fte/wmqmft/mqft/config/MQPA/agents/AGENT1/agent.properties 複製到成員 AGENT 中。

如果您需要顯示內容檔，並新增 /u/userid/fte/wmqmft/mqft/logs/MQPA/agents/AGENT1/logs 一行，這將會有幫助。

這是追蹤檔案的存放之處。

相關概念

第 62 頁的『定義協調佇列管理程式』

Managed File Transfer 要求將佇列管理程式建立為協調佇列管理程式。

第 65 頁的『使用代理程式』
如何使用各種指令確定代理程式正常運作。

相關工作

第 63 頁的『定義指令佇列管理程式』
您可以使用與協調和指令佇列管理程式相同的佇列管理程式，或者建立新的指令佇列管理程式。

使用代理程式

如何使用各種指令確定代理程式正常運作。

啟動代理程式

重新命名成員 BFGAGST、檢閱成員，然後提交工作。

如果作業成功，您會收到訊息 BFGAG0059I：代理程式已順利啟動。

顯示作用中的代理程式

重新命名成員 BFGAGLI、檢閱成員，然後提交會使用協調佇列管理程式的工作。

您必須解決任何連線問題

Ping 代理程式以確認它可運作

重新命名成員 BFGAGPI、檢閱成員，然後提交會使用指令佇列管理程式的工作。

您必須解決任何連線問題

執行測試傳送

如需進一步資訊，請參閱第 70 頁的『執行驗證傳送』。

停止代理程式

重新命名成員 BFGAGSP、檢閱成員，然後提交工作。

使用成員 BFGAGST 重新啟動代理程式。

相關概念

第 64 頁的『建立代理程式』

您必須複製 PDSE 以建立代理程式專用的 PDSE，例如 *user.MFT.AGENT1*。如果先前的代理程式或日誌程式配置中存在 PDSE，請加以複製。如果這是您的第一個配置，請複製隨附於 MFT 的 PDSE。

在 z/OS 上更新現有 MFT 代理程式或日誌程式指令資料集

您可以更新從 Managed File Transfer 指令範本資料集建立的 Managed File Transfer 指令 PDSE 程式庫資料集。

程序

1. 編輯 BFGCUSTM JCL Script 成員，並更新 BFGSTDIN DD 陳述式中的變數及內容。

如果您要移除先前定義的內容，請將其值設為空值，而非移除項目。當 BFGCUSTM JCL Script 執行時，指定的內容會以更新形式套用至實際的代理程式及日誌程式「UNIX 系統服務」內容檔；將內容設為空值，表示要移除該內容

2. 提交工作 BFGCUSTM。此工作會重新產生適用於代理程式或日誌程式的 JCL 指令集。如需指令的完整清單，請參閱第 69 頁的『z/OS 代理程式及日誌程式指令 JCL Script』。請檢查輸出工作日誌，以確認 JCL Script 已順利執行。如果執行失敗，請予以更正，然後重新提交 BFGCUSTM 工作。

結果

您可以修改產生的 JCL Script，並新增您自己的邏輯。不過，在重新執行 BFGCUSTM 時請多加留意，因為您可能會改寫自訂邏輯。

相關概念

第 50 頁的『z/OS 上的 MFT 配置選項』

z/OS 上的 Managed File Transfer 配置選項與分散式平台的選項相同。

相關工作

第 54 頁的『建立 MFT 代理程式或日誌程式指令資料集』

您可以從 Managed File Transfer 指令範本資料集，為特定的 Managed File Transfer Agent 或 Managed File Transfer Logger 建立指令的 PDSE 資料集，以進行特定協調。

z/OS JCL 變數

您可以在 BFGCUSTM Script 中使用替代值、JCL 變數及配置內容。

下表列出 MFT 指令 PDSE 程式庫資料集中的 BFGCUSTM JCL Script 所適用的替代值。在提交 BFGCUSTM 工作之前，您必須將這些替代值取代為適當的值。

替代變數	值
++library++	包含 MFT 指令 PDSE 程式庫的資料集名稱。
++bfg_java_home++	Java 安裝的位置。
++bfg_prod++	MFT 產品安裝「UNIX 系統服務」根目錄的位置。

下表說明 MFT 指令 PDSE 程式庫資料集中，BFGCUSTM JCL Script 的 BFGSTDIN DD 陳述式所適用的環境變數（位於 [Variables] 區段中）。在提交 BFGCUSTM 工作之前，您必須將所有以替代值（亦即，以兩個加號 ++ 括住的值）指定的變數取代為適當的值。

環境變數	值
LIBRARY	包含 MFT 指令 PDSE 程式庫的資料集名稱。
TMPDIR	暫存檔的「UNIX 系統服務」目錄。
BFG_PROD	MFT 產品安裝「UNIX 系統服務」根目錄的位置。
BFG_DATA	Managed File Transfer for z/OS 的資料目錄位置，此為 DATA_DIR 的路徑。
BFG_JAVA_HOME	Java 安裝的位置。
BFG_JVM_PROPERTIES	選用項目。設定 BFG_JVM_PROPERTIES 環境變數的值。這些內容會傳遞至 Java 虛擬機器。

表 9: 環境變數 (繼續)

環境變數	值
BFG_GROUP_NAME	<p>mqm 檔案群組通常會與 MFT 配置資料檔案和指令相關聯。因此，mqm 群組的所有使用者成員皆可存取及變更 MFT 配置。如需相關資訊，請參閱第 294 頁的『IBM MQ 中 MFT 的檔案系統許可權』。</p> <p>對於 z/OS 系統，檔案群組是 USS 檔案系統實體，因此不一定需要定義 mqm 檔案群組。您可以使用 BFG_GROUP_NAME 環境變數，將 z/OS USS 檔案系統群組與 MFT 配置資料檔案產生關聯。例如，在 USS Shell 提示中，使用：</p> <pre>export BFG_GROUP_NAME=FTEGB</pre> <p>可定義要與後續針對現行 USS 階段作業建立的配置檔相關聯的群組 <i>FTEGB</i>。</p> <p>您可以將 BFG_GROUP_NAME 設為空白值，或將它移除。</p> <p>註: 初次執行 BFGCUSTM 時，若 MFT 配置要供多個使用者 ID 使用，請務必將 BFG_GROUP_NAME 群組設定為所有需要的使用者 ID 皆可存取。第二次執行 BFGCUSTM 時，不得變更 BFG_GROUP_NAME (否則，必須同時變更目錄中 BFG_DATA 參照之所有檔案及目錄的 USS 群組檔案權限，以反映新的 BFG_GROUP_NAME 設定)。</p> <p>V9.0.0.1 如果您在 z/OS 系統上執行 fteMigrateAgent 指令，且 BFG_GROUP_NAME 環境變數設定為非空白值，則該指令會查看使用者是否為 BFG_GROUP_NAME 變數所指名群組的成員。如果使用者不在具名群組中，指令可能會報告錯誤訊息 BFGCL0502E: 您未獲授權執行所要求的作業。無法執行。如需使用者必須符合才能順利執行該指令的準則的詳細資料，請參閱第 407 頁的『fteMigrate 代理程式: 將 FTE 7.0 代理程式移轉至 MQ 7.5 或更新版本』。</p>
BFG_WTO	<p>當 BFG_WTO 設為 YES、ON 或 TRUE 時，會啟用 z/OS 記載。這可控制寫入代理程式事件日誌的訊息是否也會寫入 z/OS 操作員日誌機能，它可讓您在從 JCL 執行代理程式時更容易存取自動化產品。遞送程式碼是 Programmer Information (11)，描述子碼是 Informational (12)。</p>
SERVICE_TYPE	<p>指定 MFT 指令程式庫是用於代理程式還是日誌程式。有效值為 AGENT 或 LOGGER。</p>
名稱	<p>SERVICE_TYPE 值的代理程式或日誌程式的名稱。</p>
QMGR	<p>與 SERVICE_TYPE 值的代理程式或日誌程式相關聯的本端佇列管理程式名稱。</p>
OUTPUT_CLASS	<p>SYSOUT 資料集的輸出類別。預設為 *，以要求與工作陳述式中的 MSGCLASS 參數相同的輸出類別。</p>
MQ_PATH	<p>用於 BFGPROF 中建立 LIBPATH 環境變數。</p>

表 9: 環境變數 (繼續)	
環境變數	值
MQ_HLQ	IBM MQ 資料集的高階限定元。
MQ_LANG	必要的語言。
DB2_HLQ	選用項目。Db2 資料集的高階限定元。
JOB CARD1	JCL 指令工作的標頭行 1。
JOB CARD2	JCL 指令工作的標頭行 2。
JOB CARD3	JCL 指令工作的標頭行 3。
ADMIN_JOB1	管理工作的標頭行 1。
ADMIN_JOB2	管理工作的標頭行 2。
ADMIN_JOB3	管理工作的標頭行 3。
FTE_CONFIG	用於移轉的現有 WMQFTE 配置。如果不需要移轉，請設為空白值。
CREDENTIAL_PATH	用於移轉的認證檔的路徑，例如 /u/user1/agent3。Managed File Transfer 移轉的認證檔必須位於 IBM WebSphere MQ File Transfer Edition 7.0.4.4 上配置資訊及配置檔的個別檔案中。只有移轉指令 BFGAGMG 和 BFGLGMG JCL Script 才需要此項目。如果不需要移轉，請設為空白值。
DB_PROPS_PATH	指定移轉的資料庫日誌程式內容檔。只有在內容檔未使用下列預設名稱及路徑時，才需使用此選項： config_directory/coordination_qmgr/databaselogger.properties。如果不需要移轉，請設為空白值。

註: MFT 隨附的 IBM MQ Jar 檔位於目錄 *MQMFT product root/java/lib* 中，一律使用且不可配置。

下表說明 MFT 指令 PDSE 程式庫資料集中，BFGCUSTM JCL Script 之 BFGSTDIN DD 陳述式的必要 MFT 配置內容。在提交 BFGCUSTM 工作之前，您必須將以替代值（亦即，以兩個加號 ++ 括住的值）指定的內容取代為適當的非空白值。這些內容會定義 MFT 配置內容的置換。您可以新增代理程式或日誌程式內容，為您的環境自訂代理程式或日誌程式。如需所有內容的清單，請參閱第 75 頁的『配置內容檔』。

表 10: BFGSTDIN DD 陳述式的必要配置內容	
內容	值
coordinationQMGr	與代理程式或日誌程式相關聯配置的協調佇列管理程式名稱。
coordinationQMGrHost	選用項目。協調佇列管理程式執行所在系統的主機名稱。如果將此內容值保留為空白，則會採用連結模式連線。
coordinationQMGrPort	選用項目。協調佇列管理程式用來接聽的埠號。只有在同時為 coordinationQMGrHost 內容指定非空白值時，才使用此參數。
coordinationQMGrChannel	選用項目。用來連接至協調佇列管理程式的通道。只有在同時為 coordinationQMGrHost 內容指定非空白值時，才使用此參數。
connectionQMGr	與代理程式或日誌程式相關聯配置的指令佇列管理程式名稱。

表 10: BFGSTDIN DD 陳述式的必要配置內容 (繼續)

內容	值
connectionQMgrHost	選用項目。執行指令佇列管理程式之系統的主機名稱。如果將此內容值保留為空白，則會採用連結模式連線。
connectionQMgrPort	選用項目。指令佇列管理程式接聽的埠號。只有在同時為 connectionQMgrHost 內容指定非空白值時，才使用此參數。
connectionQMgrChannel	選用項目。用來連接至指令佇列管理程式的通道。只有在同時為 connectionQMgrHost 內容指定非空白值時，才使用此參數。

z/OS 代理程式及日誌程式指令 JCL Script

MFT 指令 PDSE 程式庫資料集中的可用 JCL 指令集。

表 11: MFT 指令 PDSE 程式庫資料集中可用的 JCL 指令

成員	fte 指令行指令的說明
BFGCOPY	為此程式庫建立副本的工作
BFGCUSTM	為代理程式或日誌程式自訂此程式庫的工作
BFGCFCR	fteSetupCoordination
BFGCMCR	fteSetupCommands
BFGAGCR	fteCreateAgent 。只有在 SERVICE_TYPE 變數設為 AGENT 時才會建立。
BFGLGCRS	fteCreateLogger 。只有在 SERVICE_TYPE 變數設為 LOGGER 時才會建立。
BFGAGST	fteStartAgent 。只有在 SERVICE_TYPE 變數設為 AGENT 時才會建立。
BFGAGSTP	fteStartAgent 程序。只有在 SERVICE_TYPE 變數設為 AGENT 時才會建立。
BFGAGPI	ftePingAgent 。只有在 SERVICE_TYPE 變數設為 AGENT 時才會建立。
BFGAGSP	fteStopAgent 。只有在 SERVICE_TYPE 變數設為 AGENT 時才會建立。
BFGLGST	fteStartLogger 。只有在 SERVICE_TYPE 變數設為 LOGGER 時才會建立。
BFGLGSTP	fteStartLogger 程序。只有在 SERVICE_TYPE 變數設為 LOGGER 時才會建立。
BFGLGSP	fteStopLogger 。只有在 SERVICE_TYPE 變數設為 LOGGER 時才會建立。
BFGAGSH	fteShowAgentDetails 。只有在 SERVICE_TYPE 變數設為 AGENT 時才會建立。
BFGLGSH	fteShowLoggerDetails 。只有在 SERVICE_TYPE 變數設為 LOGGER 時才會建立。

表 11: MFT 指令 PDSE 程式庫資料集中可用的 JCL 指令 (繼續)

成員	fte 指令行指令的說明
BFGCFDF	fteChangeDefaultConfigurationOptions
BFGAGCL	fteCleanAgent 。只有在 SERVICE_TYPE 變數設為 AGENT 時才會建立。
BFGAGDE	fteDeleteAgent 。只有在 SERVICE_TYPE 變數設為 AGENT 時才會建立。
BFGLGDE	fteDeleteLogger 。只有在 SERVICE_TYPE 變數設為 LOGGER 時才會建立。
BFGPRSH	fteDisplayVersion
BFGAGLI	fteListAgents 。只有在 SERVICE_TYPE 變數設為 AGENT 時才會建立。
BFGMNL	fteListMonitors
BFGSTLI	fteListScheduledTransfers
BFGTMLI	fteListTemplates
BFGAGMG	fteMigrateAgent 。只有在 SERVICE_TYPE 變數設為 AGENT 時才會建立。
BFGLGMG	fteMigrateLogger 。只有在 SERVICE_TYPE 變數設為 LOGGER 時才會建立。
BFGCROBS	fteObfuscate 範例
BFGRAS	fteRAS
BFGAGTC	fteSetAgentTraceLevel 。只有在 SERVICE_TYPE 變數設為 AGENT 時才會建立。
BFGLGTC	第 428 頁的『 fteSetLoggerTraceLevel 』。只有在 SERVICE_TYPE 變數設為 LOGGER 時才會建立。
BFGPRANS	fteAnt 範例
BFGTRCAS	fteCancelTransfer 範例
BFGMNCRS	fteCreateMonitor 範例
BFGTMCRS	fteCreateTemplate 範例
BFGTRCRS	fteCreateTransfer 範例
BFGMNDES	fteDeleteMonitor 範例
BFGSTDES	fteDeleteScheduledTransfer 範例
BFGTMDES	fteDeleteTemplates 範例

> z/OS

註: 對於建立 MQSC 或參照刪除 Script 的指令, JCL 將要求您執行一個 Script, 但該 Script 已由工作執行。

> z/OS

執行驗證傳送

如何執行傳送以確認產品可正常運作。

重新命名並編輯成員 BFGTRCRS。

1. 在 %BFGCMD CMD=fteCreateTransfer -h 前面加上 /*
2. 移除成員中的其他註解。
3. 為 -sa 和 -da 指定現行代理程式名稱
4. 儲存 JCL
5. 提交 JCL

此 JCL 會連接到指令佇列管理程式。

配置記載作業

記載作業需要在與協調佇列管理程式相同的映像上執行。您可以記載到 Db2。

建立記載作業

複製 PDSE 以建立日誌程式專用的 PDSE。例如，user.MFT.LOGGER。

如果您需要使用不同的認證檔，請加以建立。如需相關資訊，請參閱第 60 頁的『在 z/OS 上配置 MQMFTCredentials.xml』。

檢閱成員 `BFGCUSTM`。請注意，有許多內容仍與先前的自訂相同。

但是，您必須：

- 變更 //SYSEXEC DD DSN=SCEN.FTE.JCL....
- 變更 LIBRARY 以符合代理程式 PDSE
- 將 QMGR 變更為協調佇列管理程式的名稱
- 使 SERVICE_TYPE=LOGGER
- 將 NAME 變更為日誌程式的名稱（符合 PDSE）
- 檢查 JOBCARD 並變更工作名稱，使該名稱與代理程式的工作名稱不同。
- 檢閱 `BFG_JVM_properties = "-Xmx1024M"`

如果您使用 Db2 日誌程式，建議您建立一個檔案，以便擷取 Db2 追蹤資料，來識別 Db2 問題。

檔案的名稱指定在 JVM 內容中，其中，JDBC 追蹤內容檔會有如下的內容

```
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
```

設定兩個 JVM 內容

```
BFG_JVM_PROPERTIES=-Ddb2.jcc.propertiesFile=/u/.../sql.properties
-Ddb2.jcc.ssid=DBC
```

其中 /u/.../sql.properties 是 Db2 追蹤內容檔的名稱，而 `DBC` 是 Db2 子系統的名稱。

提交此工作；請注意，此工作需要資料集的互斥存取權。代理程式的工作全都會有 `BFGLG*` 之類的名稱。

記載到檔案

如需如何記載至 Db2 的相關資訊，請參閱第 72 頁的『記載到 Db2 時建立記載作業』

重新命名成員 `BFGLGCRS`。此工作會更新 Managed File Transfer (MFT) 目錄中的檔案，並使用 CSQUTIL 在本端佇列管理程式中建立代理程式專用佇列。

原始檔案具有會列出指令語法的 %BFGCMD CMD=fteCreateLogger -h 指令。

若要建立日誌程式作業，請在陳述式前面放置 /* 以註銷 %BFGCMD CMD=fteCreateLogger -h，確保第一個直欄是空白的。

從第二個指令中移除註解，並配置陳述式。例如：

```
%BFGCMD CMD=fteCreateLogger +
-p MQPH +
-loggerMgr MQPH +
-loggerType FILE +
-fileLoggerMode circular +
-fileSize 5MB +
-fileCount 5 +
-p MQPH +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>'
LOGGER
```

檢查輸出，以確認是否已順利處理。

提示：將 `logger.properties` 檔案的路徑名稱從工作的輸出複製到代理程式在 PDSE 中的成員。

例如，複製到成員 APATH 中

```
/u/user_ID/fte/wmqmft/mqft/config/MQPH/loggers/LOGGER/logger.properties
```

如果您需要顯示內容檔，這將會有幫助。

將目錄新增至此檔案：

```
/u/user_ID/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/
```

如果您記載到此檔案，日誌檔將會儲存在此目錄中，例如 `LOGGER0-20140522123654897.log`。

追蹤檔案會位於日誌子目錄中，例如

```
/u/user_ID/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/logs
```

現在，您可以開始記載作業。

記載到 Db2 時建立記載作業

重新命名成員 BFGLGCRS。

此工作會更新 MFT 目錄中的檔案，並使用 CSQUTIL 在本端佇列管理程式中建立代理程式專用佇列。

您必須知道：

表 12: Db2 變數	
Db2 名稱	範例
-dbName databaseName	您可以從 Db2 子系統之訊息 DSNL004I 中的位置值取得此項目
-dbDriver filePath	例如 /db2/db2v10/jdbc/classes/db2jcc.jar
-dbLib filePath	例如 /db2/db2v10/jdbc/lib/libdb2jcct2zos_64.so

編輯檔案。原始檔案具有會列出指令語法的 %BFGCMD CMD=fteCreateLogger -h 指令。

從第二個指令中移除註解，並配置陳述式。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
```

若要建立日誌程式作業，請在陳述式前面放置 `/*` 以註銷 `%BFGCMD CMD=fteCreateLogger -h`，確保第一個直欄是空白的。

提交工作並檢查輸出，以確認是否已順利處理。

提示：將 `logger.properties` 檔案的路徑名稱從工作的輸出複製到代理程式在 PDSE 中的成員。

例如，複製到成員 `APATH` 中：

```
/u/user_ID/fte/wmqmft/mqft/config/MQPH/loggers/LOGGER/logger.properties into member USS
```

如果您需要顯示內容檔，這將會有幫助

追蹤檔案會位於日誌子目錄中，例如：

```
/u/user_ID/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/logs
```

建立 Db2 表格

您必須建立 Db2 表格。這些定義是在 `USS` 檔案 `mqft/sql/ftelog_tables_zos.sql` 中。

在您的 PDSE 中建立成員 `Db2`。編輯此成員，並在指令行上使用 `COPY` 指令。從 `USS` 定義檔中複製。

由於各站台的特定需求可能有很大的差異，此檔案所指定的只是表格的基本結構，及其所在的表格空間。

表格空間會由 `SQL Script` 指定，用以確保建立空間時所使用的緩衝池具有足以容納最大可能表格列的頁面大小。請注意，此時並不會指定 `LOB` 位置之類的屬性。

您的資料庫管理者可能會想修改此檔案的副本，以定義與效能相關的屬性。

此檔案也會採用 `FTELOG` 的預設綱目名稱、`FTELOGTS` 的預設表格空間名稱，以及 `FTELOGDB` 的資料庫名稱。如有需要，您可以變更這些名稱，以符合現有的資料庫和任何本端命名慣例；請遵循檔案開頭處的註解中說明的程序。

重要：請使用 `SPUFI` 之類的線上工具執行指令，因為檔案中有註解，而 `DSNTINAD` 之類的批次程式並不接受註解。

開始日誌程式作業

重新命名、檢查並提交成員 `BFGLGST`。您應會收到訊息 `BFGDB0023I`：日誌程式已完成啟動活動，目前正在執行中。

日誌程式作業

若要顯示日誌程式狀態，請重新命名、檢查並提交成員 `BFGLGSH`

若要停止日誌程式，請重新命名、檢查並提交成員 `BFGLGSP`。

z/OS

z/OS 上 MFT 的環境變數

如果您要直接從 `USS` 環境執行指令，或是執行您自己的 `JCL Script`，在完成自訂及配置之後，您必須在執行 `Managed File Transfer` 所提供的配置及管理 `Script` 之前，設定一些環境變數。您必須為每一個使用者以及在要呼叫 `Script` 的每一個環境中設定這些變數。

為避免與其他產品發生衝突，您可以選擇在起始目錄中建立 `.wmqfterc Script`。然後，每一個 `Managed File Transfer Script` 會呼叫 `.wmqfterc Script`，您可以使用這個 `Script` 來提供 `Managed File Transfer` 的自訂環境設定。

您還可以設定另外一個選用的環境變數 BFG_WTO，這樣當您從 JCL 執行代理程式時可將訊息傳送至操作員日誌。

表 13: 必要的 z/OS 環境變數	
環境變數	值
BFG_JAVA_HOME	Java 安裝的位置。如需支援的 Java 層次的相關資訊，請參閱 IBM MQ 系統需求 。
BFG_DATA	Managed File Transfer for z/OS 的資料目錄位置。這是 DATA_DIR 的路徑。
STEPLIB	<p>必須包括下列 IBM MQ 資料集：</p> <ul style="list-style-type: none"> • SCSQAUTH • SCSQANLE • SCSQLOAD <p>如果您想要在 z/OS 系統上執行資料庫日誌程式元件，STEPLIB 必須同時依顯示的順序包括下列 Db2 資料集：</p> <ul style="list-style-type: none"> • SDSNEXIT • SDSNLOD2 • SDSNLOAD
LIBPATH	必須包括「z/OS UNIX 系統服務」空間中 IBM MQJava 程式庫的位置 (對於 IBM MQ 8.0，預設值為 /mqm/V8R0M0/java/lib)。

下列是 .profile 範例，它正確配置 Managed File Transfer 的環境變數：

```
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
```

您也可以選擇性地設定下列環境變數：

表 14: 選用的 z/OS 環境變數

環境變數	值
BFG_WTO	<p>下列其中一個值將啟用 BFG_WTO:</p> <ul style="list-style-type: none"> • YES • 開啟 • TRUE <p>下列其中一值將停用 BFG_WTO。這些值不區分大小寫。</p> <ul style="list-style-type: none"> • NULL • NO • 關閉 • FALSE <p>啟用 z/OS 記載。依預設，會停用此環境變數。</p> <p>寫入代理程式事件日誌的訊息也會寫入 z/OS 操作員日誌機能，它可讓您在從 JCL 執行代理程式時更容易存取自動化產品。遞送程式碼是 Programmer Information (11)，描述子碼是 Informational (12)。</p>
BFG_GROUP_NAME	<p>mqm 檔案群組通常會與 Managed File Transfer 配置資料檔案和指令相關聯。因此，mqm 群組成員的所有使用者皆可存取及變更 Managed File Transfer 配置。如需相關資訊，請參閱第 294 頁的『IBM MQ 中 MFT 的檔案系統許可權』。</p> <p>對於 z/OS 系統，檔案群組是 USS 檔案系統實體，因此不一定需要定義 mqm 檔案群組。您可以使用 BFG_GROUP_NAME 環境變數，針對 Managed File Transfer 配置資料檔案定義替代的現有 z/OS USS 檔案系統群組。例如，在 USS Shell 提示中：</p> <pre style="background-color: #f0f0f0; padding: 5px;">export BFG_GROUP_NAME=FTEGB</pre> <p>可定義要與後續針對現行 USS 階段作業建立的配置檔相關聯的群組 FTEGB。</p> <p>您可以將 BFG_GROUP_NAME 設為空白值，或將它移除。</p> <p>V9.0.0.1 如果您在 z/OS 系統上執行 fteMigrateAgent 指令，且 BFG_GROUP_NAME 環境變數設定為非空白值，則該指令會查看使用者是否為 BFG_GROUP_NAME 變數所指名群組的成員。如果使用者不在具名群組中，指令可能會報告錯誤訊息 BFGCL0502E：您未獲授權執行所要求的作業。無法執行。如需使用者必須符合才能順利執行該指令的準則的詳細資料，請參閱第 407 頁的『fteMigrate 代理程式: 將 FTE 7.0 代理程式移轉至 MQ 7.5 或更新版本』。</p>

z/OS 配置內容檔

Managed File Transfer 中所使用內容的摘要。

- 第 456 頁的『MFT coordination.properties 檔』
- 第 460 頁的『MFT command.properties 檔』
- 第 463 頁的『MFT agent.properties 檔案』
- [日誌程式配置內容檔](#)

為 z/OS Automatic Restart Manager (ARM) 配置 MFT

Managed File Transfer 是已啟用 ARM 的應用程式。

開始之前

如需為您的系統啟用 ARM 以及定義 ARM 原則的相關資訊，請參閱[使用 z/OS Automatic Restart Manager \(ARM\)](#)。

如果想要使用 MFT DB 日誌程式功能自動重新啟動並重新連接至 Db2 資料庫，ARM 是唯一可用的受支援重新啟動管理程式。

關於這項作業

使用 ARM，可以透過設定代理程式/日誌程式內容 armELEMTYPE 及 armELEMENT，將代理程式及日誌程式配置為重新啟動。內容 armELEMTYPE 定義 ARM 元素的類型，內容 armELEMENT 則是 ARM 要登錄的元素名稱：

- 您可以將代理程式 ELEMTYPE 設為 SYSBFGAG，而 armELEMENT 可以設為對應於代理程式名稱。
- 您可以將日誌程式 ELEMTYPE 設為 SYSBFGGLG，而 armELEMENT 可以設為對應於日誌程式名稱。

註：配置為由 ARM 重新啟動的代理程式及日誌程式，只能透過批次工作或已啟動的作業順利執行。嘗試從 USS 指令行直接啟動代理程式或日誌程式將會失敗，並產生 ARM 錯誤原因碼。

範例

下列重新啟動原則的範例將代理程式 BFGFT7CAG1 定義為與佇列管理程式 FT7C 相依：

```
RESTART_ORDER
  LEVEL(3)
  ELEMENT_TYPE(SYSBFGAG, SYSBFGGLG)

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

搭配使用 Managed File Transfer for z/OS 與 JZOS Java 啟動程式

在 IBM MQ for z/OS 系統上，您可以套用本主題中的指示，作為在企業中使用 Managed File Transfer 的替代方法。

概觀

Managed File Transfer for z/OS (MFT) 使用標準 z/OS 安裝程序。執行 MFT 指令的另一種方式是使用 JCL 及 JZOS Java Launcher。

如需進一步詳細資料，請參閱[JZOS 批次啟動程式和工具箱](#)。

如果 JCL 無法正確處理，請參閱[第 253 頁的『JZOS 的一般 MFT 問題』](#)。

IBM MQ 8.0 以及更新版本的 JCL 範例



小心：若為 IBM WebSphere MQ File Transfer Edition 7.0，參數是以 FTE_ 開頭，而不是以 BFG_ 開頭。

```
//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 陳述式的位置
- (2) 包括來自 1) 的位置的指定 JCL 成員
- (2A) 這會延伸 //STDENV - 請參閱下方說明
- (3) 這是要執行的指令，不含前導 fte 字首
- (4) 這幾行是必要項目，它們可設定 JZOS 的資訊
- (5) 指令的參數
- BFGJCL8 成員（您可以選取自己的名稱）會呼叫 JZOS。此成員有執行 MFT 所需的 STEPLIB 和其他 JCL。

您需要包含的其他 JCL

您應該包括 IBM MQ for z/OS 程式庫的 JCL，如果您使用的是 Db2 日誌程式，則還要包括 Db2 程式庫的 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.SDSNLOAD2
//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 程式的名稱。請查看 SYS1.SIEALNKE 以瞭解您系統上的版本。請新增 ,PARM='+T' 以提供其他診斷資訊。
- (2) 這是含有 JZOS 程式的資料集。
- (3) 這是 Shell Script 的成員名稱。它會定義 MFT 所需的參數。請參閱 [第 77 頁的『用於定義 MFT 的 Shell Script』](#)。

它可以是任何資料集及成員。它必須位於檔案末尾，因為 JCL 工作會加以延伸。請參閱 [第 76 頁的『IBM MQ 8.0 以及更新版本的 JCL 範例』](#) 中的 2A。

用於定義 MFT 的 Shell Script

在 [第 77 頁的『您需要包含的其他 JCL』](#) 範例中，使用了成員 BFGZENV8。這是基於 JZOS 設定檔。

您可以對 MFT 第 8 版和 IBM WebSphere MQ File Transfer Edition 7.0 使用相同的配置檔，只進行一些小小的變更。請記住，在 MFT 第 8 版之前，參數是以 FTE 開頭。請參閱第 78 頁的『範例檔』。

您必須知道：

- 安裝 Java 的位置
- IBM MQ for z/OS Java 程式庫的位置
- MFT 檔案的位置
- 使用者 ID 需要位於特定群組內，才能視為 IBM MQ for z/OS 管理者。您需要此群組的名稱
- 如果您不要使用英文訊息，則需要知道要指定哪一種語言。

範例檔

```
# 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"

對於 Unicode 轉換是必要項目

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

是 Java 目錄的位置。請指定 Java 的路徑名稱。此目錄包含 bin 和其他目錄。

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

設定 Java 可執行陳述式的路徑陳述式

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

設定 Java 可執行陳述式的程式庫路徑

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

在 LIBPATH 陳述式中新增更多 Java 程式庫。

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

在程式庫路徑中新增 IBM MQ for z/OS 程式庫。請指定 USS 中的 IBM MQ for z/OS 程式庫的名稱。

export LIBPATH

使 LIBPATH 可用於 JZOS

export BFG_JAVA_HOME="\${JAVA_HOME}"

將 BFG_JAVA_HOME 設為上方指定的 JAVA_HOME 的值

export BFG_WTO="YES"

將 BFG_WTO 設為 YES 會使用 WTO 將訊息顯示在工作日誌中

export BFG_GROUP_NAME=MQADM

屬於所指定群組之成員的使用者 ID，將視為 IBM MQ for z/OS 管理者

export BFG_PROD="/HMF8800/"

是 MFT 程式碼所在位置的路徑

export BFG_DATA="/u/johndoe/fteconfig"

是儲存 MFT 配置資訊的位置

export BFG_LANG="-Duser.language=de "

是將語言定義為德文的已註銷陳述式

export BFG_LANG=" "

將語言指定為預設值（英文）。

`/lib/messages/BFGNVMessages_*.properties` 中的 MFT 產品內容列出可用的語言。預設值是將值保留為空白，表示將使用英文。

若為第 7 版，請指定：

```
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=" "
```

請注意，`/u/johndoe/fteconfig` 與 `BFG_DATA` 中的路徑不同

相關資訊

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

[規劃 Managed File Transfer](#)

IBM i 在 IBM i 上配置 MFT

在安裝 Managed File Transfer 之後，您必須完成協調佇列管理程式及代理程式的某些配置，才能開始使用它。

關於這項作業

安裝之後，您必須對新的協調佇列管理程式及新的代理程式執行 Managed File Transfer 所提供的配置 Script，才能使用協調佇列管理程式及代理程式來傳送檔案。然後，您必須啟動已建立的代理程式。

程序

- 對於所有新的協調佇列管理程式：針對協調佇列管理程式執行 `coordination_qmgr_name.mqsc` 檔案中的 MQSC 指令。如果協調佇列管理程式與安裝不在相同的電腦上，請將 MQSC Script 檔複製到佇列管理程式所在的電腦上，然後再執行 Script。
 - 從 IBM i 指令行使用下列指令啟動 qshell：CALL QSHELL
 - 切換至下列目錄：`/QIBM/UserData/mqm/mqft/config/coordination_qmgr_name`
 - 發出下列指令，將 `coordination_qmgr_name` 取代為佇列管理程式的名稱：

```
/QSYS.LIB/QMQM.LIB/RUNMQSC.PGM coordination_qmgr_name < coordination_qmgr_name.mqsc
```

您可以改用手動方式配置協調佇列管理程式。如需相關資訊，請參閱第 82 頁的『配置 MFT 的協調佇列管理程式』。

- 對於所有新的代理程式：針對代理程式佇列管理程式執行 `agent_name_create.mqsc` 檔案中的 MQSC 指令。

如果代理程式佇列管理程式與代理程式不在相同的電腦上，請將 MQSC Script 檔複製到佇列管理程式所在的電腦上，然後再執行 Script。

 - 從 IBM i 指令行使用下列指令啟動 qshell：CALL QSHELL

- b) 切換至下列目錄： `/QIBM/UserData/mqm/mqft/config/agent_qmgr_name/agents`
- c) 發出下列指令，將 `agent_qmgr_name` 取代為代理程式佇列管理程式的名稱，並將 `agent_name` 取代為代理程式的名稱：

```
/QSYS.LIB/QMQM.LIB/RUNMQSC.PGM agent_qmgr_name < agent_name_create.mqsc
```

您可以改用手動方式配置代理程式佇列管理程式。如需相關資訊，請參閱第 83 頁的『配置 MFT 代理程式佇列管理程式』。

3. 如果您在安裝期間尚未啟動 QMFT 子系統，請從 IBM i 指令行使用下列指令來啟動 QMFT 子系統：
`STRSBS SBSD(QMQMMFT/QMFT)` 或 `STRSBS QMQMMFT/QMFT`
4. 使用 `fteStartAgent` 指令來啟動新的代理程式。
 - a) 從 IBM i 指令行使用下列指令啟動 qshell：`CALL QSHELL`
 - b) 切換至下列目錄：`/QIBM/ProdData/mqm/bin`
 - c) 發出下列指令，將 `AGENT` 取代為代理程式的名稱：

```
./fteStartAgent AGENT
```

下一步

建議您設定沙盤推演，以限定代理程式可存取的檔案系統區域。第 39 頁的『使用 MFT 代理程式沙盤推演』中說明了此特性。

相關概念

第 80 頁的『初次使用時配置 MFT』

初次使用 Managed File Transfer 代理程式及佇列管理程式時，必須先執行某些配置作業。

初次使用時配置 MFT

初次使用 Managed File Transfer 代理程式及佇列管理程式時，必須先執行某些配置作業。

連線至 IBM MQ

使用 IBM MQ 佇列管理程式的所有網路通訊（包括與 Managed File Transfer 相關的通訊）都涉及 IBM MQ 通道。IBM MQ 通道代表網路鏈結的一端。通道分類為訊息通道或 MQI 通道。

Managed File Transfer 及通道

Managed File Transfer 使用 MQI 通道在用戶端模式下將代理程式連接至其代理程式佇列管理程式，以及將指令應用程式（例如，`fteCreateTransfer`）連接至其指令及協調佇列管理程式。在預設配置中，使用稱為 `SYSTEM.DEF.SVRCONN` 的 `SVRCONN` 通道（依預設，存在於所有佇列管理程式上）進行這些連線。由於這些預設值，您無需變更基本 Managed File Transfer 安裝的任何 MQI 通道。

訊息通道端點有六種類型，但本主題只會說明傳送端至接收端通道配對。請參閱分散式佇列元件，以取得其他通道組合的相關資訊。

所需的訊息路徑

IBM MQ 訊息只能透過訊息通道傳送，因此，您必須確保通道可用於 Managed File Transfer 所需的所有訊息路徑。這些路徑不必是直接的；必要時，可以透過中間佇列管理程式傳送訊息。此主題僅涵蓋直接點對點通訊。如需這些選項的相關資訊，請參閱如何進入遠端佇列管理程式。

Managed File Transfer 使用的通訊路徑如下：

代理程式至代理程式

在之間傳送檔案的任何兩個代理程式，需要在其關聯的佇列管理程式之間進行雙向通訊。由於此路徑會傳送大量資料，因此，請根據您的需要考量儘可能讓路徑簡短、快速或便宜。

代理程式至協調

參與傳送的代理程式中的日誌訊息，必須能夠抵達協調佇列管理程式。

指令至代理程式

指令應用程式或「IBM MQ Explorer」（使用指令佇列管理程式）所連接的任何佇列管理程式，必須能夠將訊息傳送至使用這些指令應用程式控制的代理程式的佇列管理程式。若要啟用指令顯示的回饋訊息，請使用雙向連線。

如需相關資訊，請參閱適用於您企業所使用之一個以上平台的驗證 *IBM MQ* 安裝。

相關概念

第 86 頁的『配置多重實例佇列管理程式以使用 MFT』

IBM WebSphere MQ 7.0.1 及更新版本支援建立多重實例佇列管理程式。多重實例佇列管理程式可自動在待命伺服器上重新啟動。Managed File Transfer 支援與多重實例代理程式佇列管理程式、多重實例協調佇列管理程式，以及多重實例指令佇列管理程式的連線。

相關工作

第 81 頁的『配置 MFT 網路佇列管理程式』

如果您的 Managed File Transfer 網路包含多個 IBM MQ 佇列管理程式，這些 IBM MQ 佇列管理程式必須能夠在遠端彼此進行通訊。

第 82 頁的『配置 MFT 的協調佇列管理程式』

執行 **fteSetupCoordination** 指令之後，請執行 *MQ_DATA_PATH/mqft/config/coordination_qmgr_name* 目錄中的 *coordination_qmgr_name.mqsc* Script，以執行協調佇列管理程式所需的配置。但如果要手動執行此配置，請在協調佇列管理程式上執行下列步驟。

配置 MFT 網路佇列管理程式

如果您的 Managed File Transfer 網路包含多個 IBM MQ 佇列管理程式，這些 IBM MQ 佇列管理程式必須能夠在遠端彼此進行通訊。

關於這項作業

有以下兩種方式來將佇列管理程式配置為能夠彼此進行通訊：

- 藉由設定 IBM MQ 佇列管理程式叢集。

如需 IBM MQ 佇列管理程式叢集及如何配置它們的相關資訊，請參閱[配置佇列管理程式叢集](#)。

- 藉由設定佇列管理程式之間的通道（如下所述）：

設定佇列管理程式之間的通道

在佇列管理程式之間設定下列訊息通道：

- 從代理程式佇列管理程式至協調佇列管理程式
- 從指令佇列管理程式至代理程式佇列管理程式。
- 從代理程式佇列管理程式至指令佇列管理程式（使回饋訊息能夠以指令顯示）。
- 從指令佇列管理程式至協調佇列管理程式
- 從代理程式佇列管理程式至 Managed File Transfer 網路中的任何其他代理程式佇列管理程式

如果您需要如何設定此通訊的進一步相關資訊，請從以下資訊開始：[使用 MQSC 管理遠端 IBM MQ 物件](#)。

部分建議的範例步驟如下：

程序

1. 使用與協調佇列管理程式相同的名稱，在 IBM MQ 佇列管理程式上建立傳輸佇列。

您可以使用下列 MQSC 指令：

```
DEFINE QLOCAL(coordination-qmgr-name) USAGE(XMITQ)
```

2. 在 IBM MQ 佇列管理程式上，建立 Managed File Transfer 協調佇列管理程式的傳送端通道。在上一步中建立的傳輸佇列名稱是此通道所需的參數。如果需要與 IBM WebSphere MQ 7.5 或 Managed File Transfer 代理程式中的 Managed File Transfer 通訊，請確定傳送端通道的 CONVERT 參數設為 no。(舊版 IBM WebSphere MQ File Transfer Edition 一律以 UTF-8 格式發佈訊息，這表示任何資料轉換都會毀損訊息。對於 IBM MQ 8.0 或更新版本中 Managed File Transfer 上的代理程式而言，這是不必要的，因為會以空白格式發佈訊息。)

您可以使用下列 MQSC 指令：

```
DEFINE CHANNEL(channel-name) CHLTYPE(SDR) CONNAME('coordination-qmgr-host(coordination-qmgr-port)')
XMITQ(coordination-qmgr-name) CONVERT(NO)
```

註：只有在必要時設定 CONVERT(NO)。

3. 在 Managed File Transfer 協調佇列管理程式上，建立 IBM MQ 佇列管理程式的接收端通道。為此接收端通道指定與 IBM MQ 佇列管理程式的傳送端通道相同的名稱。

您可以使用下列 MQSC 指令：

```
DEFINE CHANNEL(channel-name) CHLTYPE(RCVR)
```

下一步

接下來，請遵循協調佇列管理程式的配置步驟：[配置協調佇列管理程式](#)。

配置 MFT 的協調佇列管理程式

執行 `fteSetupCoordination` 指令之後，請執行 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` 目錄中的 `coordination_qmgr_name.mqsc` Script，以執行協調佇列管理程式所需的配置。但如果要手動執行此配置，請在協調佇列管理程式上執行下列步驟。

關於這項作業

程序

1. 建立稱為 SYSTEM.FTE 的本端佇列。
2. 將 SYSTEM.FTE 佇列新增至 SYSTEM.QPUBSUB.QUEUE.NAMELIST 名稱清單。
3. 使用 SYSTEM.FTE 的主題字串建立稱為 SYSTEM.FTE 的主題。
4. 確保 SYSTEM.FTE 主題的非持續性訊息遞送 (NPMGDLV) 及持續性訊息遞送 (PMGDLV) 屬性設定為 ALLAVAIL。
5. 確保協調佇列管理程式的「發佈/訂閱」模式 (PSMODE) 屬性設定為 ENABLED。

下一步

如果您在已配置為協調佇列管理程式的佇列管理程式上執行 `strmqm -c` 指令，則該指令會刪除 [步驟 2](#) 中所做的變更 (將 SYSTEM.FTE 佇列新增至 SYSTEM.QPUBSUB.QUEUE.NAMELIST 名單)。這是因為 `strmqm -c` 會重建預設的 IBM MQ 物件，並反轉 Managed File Transfer 變更。因此，如果您已使用 `strmqm -c` 啟動佇列管理程式，請完成下列其中一個步驟：

- 再次於佇列管理程式上執行 `coordination_qmgr_name.mqsc` Script。
- 重複 [步驟 2](#)。

相關概念

[第 80 頁的『連線至 IBM MQ』](#)

使用 IBM MQ 佇列管理程式的所有網路通訊 (包括與 Managed File Transfer 相關的通訊) 都涉及 IBM MQ 通道。IBM MQ 通道代表網路鏈結的一端。通道分類為訊息通道或 MQI 通道。

[第 86 頁的『配置多重實例佇列管理程式以使用 MFT』](#)

IBM WebSphere MQ 7.0.1 及更新版本支援建立多重實例佇列管理程式。多重實例佇列管理程式可自動在待命伺服器上重新啟動。Managed File Transfer 支援與多重實例代理程式佇列管理程式、多重實例協調佇列管理程式，以及多重實例指令佇列管理程式的連線。

相關工作

第 81 頁的『[配置 MFT 網路佇列管理程式](#)』

如果您的 Managed File Transfer 網路包含多個 IBM MQ 佇列管理程式，這些 IBM MQ 佇列管理程式必須能夠在遠端彼此進行通訊。

相關參考

第 433 頁的『[fteSetupCoordination](#)』

fteSetupCoordination 指令會建立 Managed File Transfer 的內容檔及協調佇列管理程式目錄。

配置 MFT 代理程式佇列管理程式

安裝之後，請執行 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` 目錄中的 `agent_name_create.mqsc` Script，為代理程式佇列管理程式執行所需的配置。但是，如果您要手動執行此配置，請在代理程式佇列管理程式上完成以下步驟。

程序

1. 建立代理程式作業佇列。

請命名以下佇列：

- `SYSTEM.FTE.COMMAND.agent_name`
- `SYSTEM.FTE.DATA.agent_name`
- `SYSTEM.FTE.EVENT.agent_name`
- `SYSTEM.FTE.REPLY.agent_name`
- `SYSTEM.FTE.STATE.agent_name`

如需佇列參數的相關資訊，請參閱 [第 522 頁的『MFT 代理程式佇列設定』](#)。

2. 建立代理程式權限佇列。

請命名以下佇列：

- `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`

如需佇列參數的相關資訊，請參閱 [第 522 頁的『MFT 代理程式佇列設定』](#)。

下一步

如需建立和配置通訊協定橋接器代理程式的相關資訊，請參閱 [第 330 頁的『fteCreateBridgeAgent \(建立及配置 MFT 通訊協定橋接器代理程式\)』](#) 和 [第 191 頁的『將通訊協定橋接器配置用於 FTPS 伺服器』](#)。

建立 IBM MQ File Transfer 結構

您可以透過同一部機器上，連線到佇列管理程式的單一代理程式配置 Managed File Transfer 結構。

關於這項作業

MFT 配置會儲存在代理程式所在機器之 IBM MQ DataPath 下的檔案結構中。

下列範例配置適用於名為 SAMPLECOORD (已停用安全) 的 MFT IBM MQ 8.0 佇列管理程式，以及名為 SAMPLEAGENT 的單一 MFT 代理程式：

```
+--- config
      +--- SAMPLECOORD
            +--- command.properties
            +--- coordination.properties
            +--- SAMPLECOORD.mqsc
            +--- agents
                  +--- SAMPLEAGENT
                        +--- agent.properties
                        +--- SAMPLEAGENT_create.mqsc
                        +--- SAMPLEAGENT_delete.mqsc

+---logs
      +--- SAMPLECOORD
            +--- agents
                  +--- SAMPLEAGENT
                        +--- logs
```

此範例假設已停用佇列管理程式的安全功能。下列指令（在 **runmqsc** 中執行）會在重新啟動佇列管理程式之後停用安全功能：

```
runmqsc queue manager
alter qmgr CONNAUTH(NONE);
alter qmgr CHLAUTH(DISABLED);
end;
```

對於在 IBM MQ 8.0 或更新版本的 MFT 中啟用安全的配置，**CONNAUTH** 需要使用佇列管理程式連接的所有 MFT 指令提供使用者 ID 和密碼認證。您可以為每個指令套用額外的參數 **-mquserid** 和 **-mqpassword**，或是定義 MQMFTCredentials.xml 檔案。下列範例認證檔會定義使用者 ID **fteuser**；在連接到佇列管理程式 SAMPLECOORD 時，會對此 ID 使用 MyPassword 密碼：

```
<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>
```

如需相關資訊，請參閱第 36 頁的『MFT 及 IBM MQ 連線鑑別』。

附註：

- 若要尋找 MFT 配置目錄，請使用 **fteDisplayVersion -v** 指令。
- z/OS 使用者的 MQMFTCredential.xml 檔案可在具有變數記錄格式 (RECFM=V) 或未定義記錄格式 (RECFM=U) 之分割資料集中的成員中找到。
- 對於已啟用安全的配置，請將下列參數新增至下列步驟，以將認證與相關佇列管理程式相關聯：
-credentialsFile full credential file path。
- MQMFTCredential.xml 中的明碼密碼可以使用下列指令加以模糊：

```
fteObfuscate -credentialsFile full file path to MQMFTCredentials.xml
```

程序

1. 建立協調佇列管理程式。

協調佇列管理程式是單一佇列管理程式，用於接收來自其代理程式的所有傳輸日誌及狀態資訊。請執行下列指令：

```
fteSetupCoordination -coordinationQMgr coordination_qmgr_name
```

這會建立基本最上層配置，並建立 IBM MQ Script 檔來呼叫 **coordination_qmgr_name.mqsc**。

接著需要執行下列 IBM MQ 指令，將配置載入佇列管理程式：

```
runmqsc queue manager name < coordination_qmgr_name.mqsc
```

註：您可以使用如下，將 TCP 用戶端連線到佇列管理程式：

```
fteSetupCoordination -coordinationQMGr coordination_qmgr_name  
-coordinationQMGrHost coordination_qmgr_host -coordinationQMGrPort coordination_qmgr_port  
-coordinationQMGrChannel coordination_qmgr_channel
```

您必須為所建立的 *coordination_qmgr_name.mqsc*，在協調佇列管理程式執行所在的機器上，執行 **runmqsc** 指令。

2. 建立指令佇列管理程式。

指令佇列管理程式是單一佇列管理程式，此佇列管理程式已預先配置，讓 IBM MQ 基礎架構可以將 MFT 要求路由到相關代理程式。請執行下列指令：

```
fteSetupCommands -connectionQMGr Command QM Name -p Coordination QM Name
```

這會在協調目錄中建立 *command.properties* 檔案。請注意，**-p** 為選用項目；對於針對預設協調而設定的指令並非必要。

註：您可以使用如下，將 TCP 用戶端連線到佇列管理程式：

```
fteSetupCommands -p coordination_qmgr_name -commandQMGr connection_qmgr_name  
-commandQMGrHost connection_qmgr_host -commandQMGrPort connection_qmgr_port  
-commandQMGrChannel connection_qmgr_channel
```

3. 建立代理程式。

代理程式是可以用於傳送及接收檔案的應用程式。請執行下列指令：

```
fteCreateAgent -p coordination_qmgr_name -agentName agent_name -agentQMGr agent_qmgr_name
```

這會在協調下建立代理程式配置，並建立一個 IBM MQ Script 檔，以呼叫代理程式配置目錄中的 *agent_name.mqsc*。

執行下列 IBM MQ 指令，將 IBM MQ Script 檔載入佇列管理程式中：

```
runmqsc agent_qmgr_name < agent_name_create.mqsc file
```

註：您可以使用如下，將 TCP 用戶端連線到佇列管理程式：

```
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. 啟動代理程式。

請執行下列指令：

```
fteStartAgent -p coordination_qmgr_name agentName
```

代理程式會在背景中啟動，並傳回命令提示字元。若要檢查代理程式確在執行，請執行下列指令：

```
fteListAgents -p coordination_qmgr_name
```

這會顯示代理程式的狀態。如果代理程式順利執行，則會報告其處於「備妥」狀態。

結果

基本 MFT 基礎架構已經就緒，您已可使用 **fteCreateTransfer** 指令要求傳輸。或者，如果可以使用「IBM MQ Explorer」，則可使用 MFT 外掛程式來建立及監視傳輸。

您可以重複「步驟 3：建立代理程式」，為配置新增更多的代理程式。若是使用 TCP 用戶端連線，這些代理程式可以位在不同的機器上。若為不同的機器，每台機器都必須重複 **fteSetupCoordination** 及 **fteSetupCommands** 指令，但不需要再執行 mqsc Script。

比較複雜之配置的協調及各個代理程式可有個別的佇列管理程式。在這些情況下，必須連接各個佇列管理程式。

相關參考

[第 433 頁的『fteSetupCoordination』](#)

fteSetupCoordination 指令會建立 Managed File Transfer 的內容檔及協調佇列管理程式目錄。

[第 432 頁的『fteSetupCommands：建立 MFT command.properties 檔』](#)

fteSetupCommands 指令會建立 Managed File Transfer command.properties 檔案。此內容檔針對您發出指令時連接至 IBM MQ 網路的佇列管理程式，指定其詳細資料。

[第 326 頁的『fteCreateAgent（建立 MFT 代理程式）』](#)

fteCreateAgent 指令會建立 Managed File Transfer Agent 及其相關聯的配置。

[第 417 頁的『fteObfuscate：加密機密資料』](#)

fteObfuscate 指令可為認證檔案中的機密資料加密。這會讓取得檔案存取權的人員無法讀取認證檔案的內容。

[第 696 頁的『MFT 認證檔格式』](#)

MQMFTCredentials.xml 檔案包含機密的使用者 ID 及密碼資訊。MQMFTCredentials.xml 檔案中的元素必須符合 MQMFTCredentials.xsd 綱目。認證檔的安全由使用者負責維護。

[第 463 頁的『MFT agent.properties 檔案』](#)

每一個 Managed File Transfer Agent 都有自己的內容檔 agent.properties，其中必須包含代理程式用來連接至其佇列管理程式的資訊。agent.properties 檔案也可以包含變更代理程式行為的內容。

[第 253 頁的『fteListAgents 指令未列出您的 MFT 代理程式時應該如何處理』](#)

如果您的代理程式未由 **fteListAgents** 指令列出或未顯示在 IBM MQ Explorer 中，或您的檔案傳送未顯示在 IBM MQ Explorer 的傳送日誌中，您可以執行許多問題判斷步驟來調查原因。

配置多重實例佇列管理程式以使用 MFT

IBM WebSphere MQ 7.0.1 及更新版本支援建立多重實例佇列管理程式。多重實例佇列管理程式可自動在待命伺服器上重新啟動。Managed File Transfer 支援與多重實例代理程式佇列管理程式、多重實例協調佇列管理程式，以及多重實例指令佇列管理程式的連線。

配置多重實例佇列管理程式

重要：如需配置 IBM MQ 多重實例佇列管理程式的相關資訊，請參閱[多重實例佇列管理程式](#)。在嘗試配置多重實例佇列管理程式以使用 Managed File Transfer 之前，請確定您已閱讀此資訊。

使用多重實例佇列管理程式作為代理程式佇列管理程式

若要讓代理程式能夠同時連接至多重實例佇列管理程式的作用中及待用實例，請將 agentQMGrStandby 內容新增至代理程式的 agent.properties 檔。agentQMGrStandby 內容可定義待命佇列管理程式實例的用戶端連線所使用的主機名稱及埠號。此內容的值必須以 MQ CONNAME 格式指定，亦即 `host_name(port_number)`。

agentQMGr 內容可指定多重實例佇列管理程式的名稱。agentQMGrHost 內容可指定作用中佇列管理程式實例的主機名稱，agentQMGrPort 內容則會指定作用中佇列管理程式實例的埠號。代理程式必須以用戶端模式連接至多重實例佇列管理程式的作用中及待命實例。

如需相關資訊，請參閱[第 463 頁的『MFT agent.properties 檔案』](#)。

此範例顯示連接至多重實例佇列管理程式 QM_JUPITER 之 AGENT1 的 agent.properties 檔案內容。QM_JUPITER 的作用中實例位於系統 host1 上，並使用埠號 1414 進行用戶端連線。QM_JUPITER 的待命實例位於系統 host2 上，並使用埠號 1414 進行用戶端連線。

```
agentName=AGENT1
agentDesc=
agentQMgr=QM_JUPITER
agentQMgrPort=1414
agentQMgrHost=host1
agentQMgrChannel=SYSTEM.DEF.SVRCONN
agentQMgrStandby=host2(1414)
```

使用多重實例佇列管理程式作為協調佇列管理程式

若要啟用與多重實例協調佇列管理程式的作用中及待命實例的連線，請將 coordinationQMgrStandby 內容新增至 Managed File Transfer 拓撲中的所有 coordination.properties 檔案。

如需相關資訊，請參閱第 456 頁的『MFT coordination.properties 檔』。

此範例顯示指定多重實例協調佇列管理程式 QM_SATURN 之連線詳細資料的 coordination.properties 檔案內容。QM_SATURN 的作用中實例位於系統 coordination_host1 上，並使用埠號 1420 進行用戶端連線。QM_SATURN 的待命實例位於系統 coordination_host2 上，並使用埠號 1420 進行用戶端連線。

```
coordinationQMgr=QM_SATURN
coordinationQMgrHost=coordination_host1
coordinationQMgrPort=1420
coordinationQMgrChannel=SYSTEM.DEF.SVRCONN
coordinationQMgrStandby=coordination_host2(1420)
```

Managed File Transfer 獨立式日誌程式必須一律以連結模式連接至其佇列管理程式。搭配使用獨立式日誌程式與多重實例協調佇列管理程式時，請以連結模式將獨立式日誌程式連接至不同的佇列管理程式。此作業步驟詳述於第 105 頁的『MFT 獨立式日誌程式的替代配置』中。您必須使用多重實例協調佇列管理程式的兩個實例的主機名稱及埠號，定義獨立式日誌程式的佇列管理程式與協調佇列管理程式之間的通道。如需如何執行此作業的相關資訊，請參閱多重實例佇列管理程式。

IBM MQ Explorer 的 Managed File Transfer 外掛程式以用戶端模式連接至協調佇列管理程式。如果多重實例協調佇列管理程式的作用中實例失敗，則協調佇列管理程式的待命實例即會成為作用中實例，讓外掛程式重新連接。

Managed File Transfer 指令 **ftelist*** 及 **fteShowAgentDetails** 可直接連接至協調佇列管理程式。如果多重實例協調沒有可用的作用中實例，這些指令將會嘗試連接至協調佇列管理程式的待用實例。

使用多重實例佇列管理程式作為指令佇列管理程式

若要啟用與多重實例指令佇列管理程式的作用中及待命實例的連線，請將 connectionQMgrStandby 內容新增至 Managed File Transfer 拓撲中的所有 command.properties 檔案。

如需相關資訊，請參閱第 460 頁的『MFT command.properties 檔』。

此範例顯示指定多重實例指令佇列管理程式 QM_MARS 之連線詳細資料的 command.properties 檔案內容。QM_MARS 的作用中實例位於系統 command_host1 上，並使用埠號 1424 進行用戶端連線。QM_MARS 的待命實例位於系統 command_host2 上，並使用埠號 1424 進行用戶端連線。

```
connectionQMgr=QM_SATURN
connectionQMgrHost=command_host1
connectionQMgrPort=1424
connectionQMgrChannel=SYSTEM.DEF.SVRCONN
connectionQMgrStandby=command_host2(1424)
```

相關概念

第 80 頁的『連線至 IBM MQ』

使用 IBM MQ 佇列管理程式的所有網路通訊（包括與 Managed File Transfer 相關的通訊）都涉及 IBM MQ 通道。IBM MQ 通道代表網路鏈結的一端。通道分類為訊息通道或 MQI 通道。

相關工作

第 81 頁的『配置 MFT 網路佇列管理程式』

如果您的 Managed File Transfer 網路包含多個 IBM MQ 佇列管理程式，這些 IBM MQ 佇列管理程式必須能夠在遠端彼此進行通訊。

第 82 頁的『配置 MFT 的協調佇列管理程式』

執行 `fteSetupCoordination` 指令之後，請執行 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` 目錄中的 `coordination_qmgr_name.mqsc` Script，以執行協調佇列管理程式所需的配置。但如果要手動執行此配置，請在協調佇列管理程式上執行下列步驟。

保留 MFT 日誌訊息

Managed File Transfer 會將檔案傳送進度及日誌資訊，傳送至協調佇列管理程式。協調佇列管理程式會將此資訊發佈至 SYSTEM.FTE 主題的任何相符訂閱。如果沒有訂閱，則不保留此資訊。

如果傳送進度或日誌資訊對您的業務很重要，您必須採取下列其中一個步驟，以確保將資訊保留下來：

- 使用 Managed File Transfer 資料庫日誌程式，將發佈至 SYSTEM.FTE/Log 主題的訊息複製到 Oracle 或 Db2 資料庫。
- 定義 SYSTEM.FTE 主題，將發佈資訊儲存在 IBM MQ 佇列上。在傳送任何檔案傳送之前定義此訂閱，以確保所有進度及日誌訊息都保留在佇列上。
- 撰寫應用程式，以使用訊息佇列介面 (MQI) 或 IBM MQ JMS 來建立可延續訂閱，並處理分送至訂閱的發佈。在傳送任何檔案之前，此應用程式必須在作業中，以確保應用程式能夠接收所有進度及日誌訊息。

下列幾節會詳述以上每一種方法。

請不要依賴「IBM MQ Explorer」外掛程式來保留日誌資訊。

使用 Managed File Transfer 資料庫日誌程式來保留日誌訊息

資料庫日誌程式是 Managed File Transfer 的選用元件，可用來將日誌資訊複製到資料庫中進行分析及審核。資料庫日誌程式是一個獨立式 Java 應用程式，可安裝在管理協調佇列管理程式及資料庫的系統上。如需資料庫日誌程式的相關資訊，請參閱第 89 頁的『配置 MFT 日誌程式』。

使用 IBM MQ 探險家外掛程式保留進度及日誌訊息

第一次啟動「IBM MQ Explorer」外掛程式的實例時，此實例會在協調佇列管理程式上建立可延續訂閱。這個可延續訂閱用來收集「傳送日誌」及「現行傳送進度」視圖中顯示的資訊。

可延續訂閱的名稱會加上字首，以顯示該訂閱是由「IBM MQ Explorer」MFT 外掛程式、主機名稱和使用者的名稱所建立，例如 `MQExplorer_MFT_Plugin_HOST_TJWatson`。

新增此字首是為了方便管理者刪除「IBM MQ Explorer」外掛程式實例目前不再使用的可延續訂閱。

在協調佇列管理程式上使用可延續訂閱會造成在 SYSTEM.MANAGED.DURABLE 佇列上建置訊息。如果您有流量高的 Managed File Transfer 網路，而且（或者）又不常使用「IBM MQ Explorer」外掛程式，則此訊息資料可能會填入本端檔案系統中。

若要停止發生此情況，可以指定 IBM MQ Explorer 外掛程式對協調佇列管理程式使用不可延續的訂閱。請在 IBM MQ Explorer 中執行下列步驟：

1. 選取視窗 > 喜好設定 > MQ Explorer > Managed File Transfer
2. 從傳送日誌訂閱類型清單中，選擇 NON_DURABLE。

將發佈儲存在 IBM MQ 佇列

若要將日誌或進度訊息儲存在 IBM MQ 佇列，請在協調佇列管理程式上配置訂閱，這會將訊息轉遞至此佇列。例如，若要將所有日誌訊息轉遞至名為 LOG.QUEUE 的佇列，請提交下列 MQSC 指令：

```
define sub(MY.SUB) TOPICSTR('Log/#') TOPICOBJ(SYSTEM.FTE) DEST(LOG.QUEUE) WSCHEMA(TOPIC)
```


在日誌訊息轉遞至 IBM MQ 佇列之後，它們會持續保存在佇列上，直到使用該佇列的 IBM MQ 應用程式處理這些訊息為止。

撰寫應用程式來管理 SYSTEM.FTE 主題的可延續訂閱

您可以使用 IBM MQ 支援的其中一個應用程式設計介面，撰寫應用程式來管理他們自己對 SYSTEM.FTE 主題的可延續訂閱。這些應用程式可以接收 IBM MQ 佇列或日誌訊息，並採取符合業務需求的方式進行處理。

如需可用應用程式設計介面的相關資訊，請參閱開發應用程式。

配置 MFT 日誌程式

當 Managed File Transfer 傳輸檔案時，它會將其動作的相關資訊發佈至協調佇列管理程式上的主題。資料庫日誌程式是 Managed File Transfer 的選用元件，可用於將此資訊複製到資料庫中進行分析及審核。

日誌程式有三個版本：

- 獨立式檔案日誌程式
- 獨立式資料庫日誌程式
- Java Platform, Enterprise Edition (Java EE) 日誌程式

重要：IBM i 平台不支援 Managed File Transfer 日誌程式。

獨立式檔案日誌程式

獨立式檔案日誌程式是一種 Java 程序，可以在管理協調佇列管理程式的系統上執行，或是在管理與協調佇列管理程式連接之佇列管理程式的系統上執行。獨立式檔案日誌程式使用 IBM MQ 連結來連接至其相關聯的佇列管理程式。獨立式日誌程式是使用 `fteCreateLogger` 指令建立的。

您可以 Windows 服務方式執行獨立式檔案日誌程式，以確保檔案日誌程式在您從 Windows 階段作業登出時繼續執行，且可以配置為在系統重新啟動時自動啟動。如需相關指示，請參閱第 90 頁的『安裝 MFT 獨立式檔案日誌程式』。

z/OS 或 IBM i 上不支援獨立式檔案日誌程式。

獨立式資料庫日誌程式

獨立式資料庫日誌程式是一種 Java 應用程式，可安裝在管理佇列管理程式及資料庫的系統上。獨立式資料庫日誌程式通常會與協調佇列管理程式安裝在相同的系統上，但也可安裝在與任何可連線至協調佇列管理程式的佇列管理程式相同的系統上。獨立式資料庫日誌程式使用 IBM MQ 連結來連接到其相關聯的佇列管理程式，並使用第 2 類或第 4 類 JDBC 驅動程式連接至 Db2 或 Oracle 資料庫。需要這些連線類型的原因在於，獨立式資料庫日誌程式使用佇列管理程式的 XA 支援來協調佇列管理程式及資料庫上的廣域交易，進而保護資料。

如果您使用的是 Windows 系統，則可以將獨立式日誌程式作為 Windows 服務來執行，以確保日誌程式在您從 Windows 階段作業登出後能夠繼續執行。如需相關指示，請參閱適用於獨立式資料庫日誌程式的第 96 頁的『安裝 MFT 獨立式資料庫日誌程式』。

Java EE 資料庫日誌程式

Java EE 資料庫日誌程式以 EAR 檔形式提供，可以將其安裝到應用程式伺服器中。如果具備現有 Java EE 應用程式伺服器環境可供使用，則與使用獨立式資料庫日誌程式相比，Java EE 資料庫日誌程式可能更為方便，因為可以在管理 JEE 資料庫日誌程式的同時管理其他企業應用程式。您也可以將 Java EE 資料庫日誌程式，安裝在與管理 IBM MQ 伺服器及資料庫的系統不相同的系統上。Java EE 資料庫日誌程式支援與 Db2 及 Oracle 資料庫搭配使用。當安裝在 WebSphere Application Server 7.0 上時，Java EE 資料庫日誌程式也支援 Oracle Real Application Clusters。

如需如何配置日誌程式的相關指示，請參閱下列主題：

- [第 90 頁的『安裝 MFT 獨立式檔案日誌程式』](#)
- [第 96 頁的『安裝 MFT 獨立式資料庫日誌程式』](#)

- [第 105 頁的『安裝 MFT 的 Java EE 資料庫日誌程式』](#)

安裝 MFT 獨立式檔案日誌程式

獨立式檔案日誌程式是必須使用 IBM MQ 連結連接至協調佇列管理程式的 Java 處理程序。若要定義獨立式檔案日誌程式，請使用 **fteCreateLogger** 指令，並遵循本主題中的步驟。

關於這項作業

如需獨立式檔案日誌程式的相關資訊，請參閱第 89 頁的『[配置 MFT 日誌程式](#)』。本主題中的步驟會配置日誌程式，以連接至協調佇列管理程式。如需替代日誌程式配置，請參閱第 105 頁的『[MFT 獨立式日誌程式的替代配置](#)』。

z/OS 或 IBM i 上不支援獨立式檔案日誌程式。

程序

1. 確定您已安裝 Managed File Transfer Logger 元件。如需相關資訊，請參閱 [Managed File Transfer 產品選項](#)
2. 執行 **fteCreateLogger** 指令，以指定協調佇列管理程式，並將參數 `-loggerType` 設為 FILE，以建立您的獨立式檔案日誌程式。如需相關資訊，請參閱第 343 頁的『[fteCreate 日誌程式 \(建立 MFT 檔案或資料庫日誌程式\)](#)』。
3. 選擇性的: 如果您想要使用自訂格式，請修改 **fteCreateLogger** 指令所建立的 XML 檔。日誌格式定義位於 `FileLoggerFormat.xml` 檔中。如需相關資訊，請參閱第 90 頁的『[MFT 獨立式檔案日誌程式格式](#)』。
4. 針對協調佇列管理程式執行 **fteCreateLogger** 指令所提供的 MQSC 指令，以建立日誌程式佇列。
5. 找出要執行日誌程式程序的使用者，並配置該使用者的權限。如需相關資訊，請參閱第 95 頁的『[配置 MFT 獨立式檔案日誌程式的使用者存取權](#)』。
6. 選擇性的: 您可以編輯您在執行 **fteCreateLogger** 指令時所建立的 `logger.properties` 檔，進一步配置獨立式檔案日誌程式。此檔案是由鍵值組所組成的 Java 內容檔。`logger.properties` 檔案位於 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 目錄中。如需可用內容及其影響的相關資訊，請參閱第 99 頁的『[MFT 日誌程式配置內容](#)』。
7. **Windows**
選擇性的: 如果您使用的是 Windows 系統，則可以將獨立式檔案日誌程式當作 Windows 服務來執行。請搭配執行 **fteModifyLogger** 指令與 `-s` 參數。如需相關資訊，請參閱第 415 頁的『[fteModify 日誌程式 \(以 Windows 服務方式執行 MFT 日誌程式\)](#)』。
8. 使用 **fteStartLogger** 指令來啟動獨立式檔案日誌程式。如需相關資訊，請參閱第 447 頁的『[fteStartLogger](#)』。
如果您在 Windows 上執行前一個步驟，並且搭配使用 **fteModifyLogger** 指令與 `-s` 參數，則獨立式檔案日誌程式會以 Windows 服務方式啟動。
9. 檢查日誌程式輸出。獨立式檔案日誌程式會產生兩種類型的輸出：檔案傳送審核資料及日誌程式診斷資料。檔案傳送審核資料位於 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs` 中。日誌程式診斷資料位於 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name` 中
10. 您可以使用 **fteStopLogger** 指令來停止日誌程式。如需相關資訊，請參閱第 450 頁的『[fteStopLogger](#)』。

結果

MFT 獨立式檔案日誌程式格式

檔案日誌程式寫入的訊息資訊格式可定義於 `FileLoggerFormat.xml` 檔中。

日誌程式的配置目錄位於 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 中。在建立新的檔案日誌程式時會建立此檔案，其中包含檔案日誌程式所使用的預設定義集。如需預設日誌格式定義的相關資訊，請參閱第 514 頁的『[MFT 獨立式檔案日誌程式的預設日誌格式](#)』。

如果您要指定自己的自訂日誌格式，請編輯 `FileLoggerFormat.xml` 檔案。

自訂日誌格式定義

日誌格式定義由一組訊息類型組成，各個訊息類型分別有其格式定義。訊息類型的格式定義包含 XPATH 格式中提供的一組插入項目，以及用來區隔各個插入項目的分隔字元。插入項目的排序可決定在對日誌檔產生的輸出字行中放置內容的順序。例如，以下是 `callStarted` 訊息類型的定義：

```
<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>
```

此格式會在日誌檔中產生如下的字行：

```
2011-11-25T10:53:04;414d5120514d5f67627468696e6b20206466cf4e20004f02;[CSTR];
AGENT1;AGENT_QM;Managed Call;executable;echo;call test;
```

在格式定義中提供的插入項目所使用的順序，即為資訊出現在日誌檔中字行的順序。如需定義 `FileLoggerFormat.xml` 檔格式的 XML 綱目的相關資訊，請參閱第 518 頁的『[獨立式檔案日誌程式格式 XSD](#)』。

訊息類型

FTE 代理程式會將一系列不同訊息類型寫入 `SYSTEM.FTE/Log` 子主題。如需相關資訊，請參閱第 521 頁的『[SYSTEM.FTE 主題](#)』。日誌檔定義可包含下列訊息類型的格式定義：

```
callCompleted
callStarted
monitorAction
monitorCreate
monitorFired
notAuthorized
scheduleDelete
scheduleExpire
scheduleSkipped
scheduleSubmitInfo
scheduleSubmitTransfer
scheduleSubmitTransferSet
transferStarted
transferCancelled
transferComplete
transferDelete
transferProgress
```

訊息格式可能不同。大部分的訊息類型會在日誌檔中為每一個取自 `SYSTEM.FTE/Log` 子主題的日誌訊息寫入單一行。這會導致日誌格式定義中提供的 XPATH 位址與訊息根目錄產生關聯。以下是使用此方法寫入輸出的訊息類型：

```
callCompleted
callStarted
monitorAction
```

```
monitorCreate
monitorFired
notAuthorized
scheduleDelete
scheduleExpire
scheduleSkipped
scheduleSubmitInfo
scheduleSubmitTransfer
transferStarted
transferCancelled
transferComplete
transferDelete
```

另一種用來寫入日誌訊息的方法，是使用多行來表示日誌訊息內傳送集中的項目。在此情況下，會將提供的格式套用至日誌訊息內傳送集中的每一個項目。如果您想要併入傳送集內每一個項目的特定資訊，則提供的 XPATH 就必須以項目作為其 XPATH 根目錄。以下是使用此方法寫入輸出的訊息類型：

```
scheduleSubmitTransferSet
transferProgress
```

傳送集中的每一個項目會分別寫入成一行輸出。您想要對傳送集中所有項目進行修正的資訊，仍可使用與日誌訊息的根目錄相對的 XPATH 位址。在下列簡化的 `transferProgress` 格式定義範例中，會對時間戳記及傳送 ID 進行修正。任何以某項目作為其根目錄而相對的資訊，都會隨著寫入的每一個字行而有所不同。在此範例中，會寫入每一個項目的來源及目的地檔案資訊。

```
<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>
```

這會產生一行以上的日誌檔項目，格式如下：

```
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;;
```

插入項目格式

定義訊息類型的格式時，有兩種類型的插入項目可供使用：**使用者及系統**。插入項目的類型定義於插入元素的 `type` 屬性中。這兩種類型的插入項目也可透過插入元素的 `width` 及 `ignoreNull` 屬性來自訂其佈置。例如：

```
<insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
```

在此範例中，插入項目會採用在 `/transaction/@ID` 的日誌訊息中找到的資訊，並將其修除或填補為 48 個字元，然後再寫入日誌中。如果 `/transaction/@ID` 的內容是空值，則會在資訊填補為 48 個字元後寫

入字串 null，因為 ignoreNull 屬性設為 false。如果 ignoreNull 設為 true，則空字串會填補為 48 個字元，並加以寫入。設定 width="0" 表示不修整欄寬，但不表示修整寬度為 0。當發現非預期的空值時，可以透過此方式使用 ignoreNull 屬性來偵測日誌中的空值。這在對新的日誌檔定義進行除錯時，可能會很有幫助。

使用者定義插入項目

使用者插入項目包含要寫入該插入項目的資訊的 XPATH 位址。此位址參照在 FTE 日誌訊息中找到的部分資訊。如需日誌訊息格式的相關資訊，請參閱：

- [第 637 頁的『檔案傳送日誌訊息格式』](#)
- [第 658 頁的『排定的檔案傳送日誌訊息格式』](#)
- [第 663 頁的『MFT 監視器日誌訊息格式』](#)

系統定義插入項目

系統定義插入項目包含一個關鍵字，其參照無法在日誌訊息中找到或難以使用 XPATH 語言定義的部分資訊。

支援的系統插入項目是：

- type - 以簡短格式寫入日誌訊息類型。
- callArguments - 以空格區隔格式寫入提供給受管理呼叫的引數集。
- transferMetaData - 以逗點區隔的 key=value 格式寫入為傳送定義的 meta 資料項目集。

下表列出每一個訊息類型的系統定義插入項目的 "type" 值。

訊息類型	"type" 系統插入項目的值
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]

表 15: 支援的訊息類型及其 "type" 系統插入項目的摘要。(繼續)

訊息類型	"type" 系統插入項目的值
transferCancelled	[TCAN]
transferComplete	[TCOM]
transferDelete	[TDEL]
transferProgress	[TPRO]

在 MFT 獨立式檔案日誌程式中排除訊息類型

如果您想要將特定訊息類型排除在檔案日誌程式輸出之外，可以使用空的訊息類型元素。

範例

例如，下列格式定義將會阻止檔案日誌程式輸出 transferProgress 訊息。

```
<?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>
```

定義 MFT 獨立式檔案日誌程式的自訂格式

您可以在日誌格式定義中定義自訂訊息類型的子集，以減少自訂日誌檔格式時需要的配置量。

關於這項作業

如果 FileLoggerFormat.xml 檔中不包括 messageTypes 元素，則該訊息類型格式將會使用預設格式。您只需要指定想要與預設值不同的格式。

範例

在此範例中，格式定義會將 transferStarted 訊息類型的預設格式取代為此精簡版本，只輸出啟動傳送的使用者。其他訊息類型未併入此日誌格式定義中，因此全都使用預設格式：

```
<?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>
```

相關參考

第 514 頁的『MFT 獨立式檔案日誌程式的預設日誌格式』

Managed File Transfer 獨立式檔案日誌程式的預設日誌檔格式定義。

第 518 頁的『獨立式檔案日誌程式格式 XSD』

獨立式檔案格式的綱目。

減少 *MFT* 獨立式檔案日誌程式中重複的訊息

獨立式檔案日誌程式的日誌中可能會出現重複的日誌訊息。您可以使用 `logger.properties` 檔案來調整獨立式檔案日誌程式及減少重複數。

檔案日誌程式日誌中的重複訊息

在失敗的情況下，可能會將日誌訊息寫入獨立式檔案日誌程式的日誌中，而不會耗用來自 `SYSTEM.FTE/Log# topic committed to IBM MQ`。若發生此情況，當獨立式檔案日誌程式重新啟動時，它會再次擷取相同的訊息，並重新將它寫入至日誌檔中。請在手動查看日誌檔時或自動處理日誌檔時，計劃處理這些重複的可能性。為了協助偵測重複，獨立式檔案日誌程式在啟動時，會將下列訊息輸出至日誌檔：

```
BFGDB0054I: The file logger has successfully started
```

重複一律會在獨立式檔案日誌程式的啟動時間發生，因為此時會處理前一個實例失敗之前所讀取的最後一個訊息。知道啟動新實例的時機後，您就可以偵測是否預期會有重複，以及是否需要處理它們。

減少重複數

獨立式檔案日誌程式會將它處理的日誌訊息群組成交易，以改進效能。此批次大小是失敗時您會看到的重複訊息數上限。若要減少重複數，您可以在 `logger.properties` 檔案中調整下列內容：

```
wmqfte.max.transaction.messages
```

例如，將此項設定為 1，則複製的訊息數上限會減少為 1。請注意，修改此值會影響獨立式檔案日誌程式的效能，因此需要進行徹底測試，以確保這不會對系統產生不利影響。

`logger.properties` 檔案位於 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 目錄中。如需可用內容及其影響的相關資訊，請參閱第 99 頁的『[MFT 日誌程式配置內容](#)』。

配置 *MFT* 獨立式檔案日誌程式的使用者存取權

在測試環境中，您可以為一般使用者帳戶新增任何所需的新專用權。在正式作業環境中，建議您以執行工作所需的最小權限來建立新使用者。

關於這項作業

您必須將獨立式檔案日誌程式及 IBM MQ 安裝在單一系統上。配置使用者的權限，如下所示：

程序

1. 確保使用者具有讀取權限及執行權限（必要時），並且已在安裝 Managed File Transfer 的過程中安裝檔案。
2. 確定使用者有權限在位於配置目錄內的 `logs` 目錄中建立及寫入任何檔案。此目錄用於事件日誌，必要時也會用於診斷追蹤及「首次失敗資料擷取 (FFDC)」檔案。
3. 確定使用者具有專屬的群組，且不屬於任何對協調佇列管理程式具備多項權限的群組。使用者不得屬於 `mqm` 群組。在特定平台上，也會對工作人員群組自動指定佇列管理程式存取權；因此獨立式檔案日誌程式使用者不應屬於工作人員群組。您可以使用「IBM MQ Explorer」，檢視佇列管理程式本身及其內含物件的權限記錄。用滑鼠右鍵按一下物件，然後選取物件權限 > 管理權限記錄。在指令行上，您可以使用 `dspmqaout`（顯示權限）或 `dmpmqaut`（傾出權限）指令。
4. 使用 IBM MQ Explorer 或 `setmqaut`（授與或撤銷權限）指令中的管理權限記錄視窗，以新增使用者專屬群組的權限（在 UNIX 上，IBM MQ 權限僅與群組相關聯，而不與個別使用者相關聯）。必要權限如下所示：
 - 佇列管理程式上的 `CONNECT` 及 `INQUIRE`（IBM MQ Java 程式庫需要 `INQUIRE` 權限才能運作）。
 - 對 `SYSTEM.FTE` 主題的 `SUBSCRIBE` 權限。

- 對 SYSTEM.FTE.LOG.RJCT.*logger_name* 佇列的 PUT 權限。
- 對 SYSTEM.FTE.LOG.CMD.*logger_name* 佇列的 GET 權限。

指定的拒絕及指令佇列名稱為預設名稱。如果您在配置獨立式檔案日誌程式佇列時選擇了不同的佇列名稱，請改為將權限新增至那些佇列名稱。

安裝 MFT 獨立式資料庫日誌程式

完成下列步驟，以安裝並配置獨立式資料庫日誌程式。

關於這項作業

重要: IBM i 平台不支援 Managed File Transfer 日誌程式。

如需獨立式資料庫日誌程式的相關資訊，請參閱第 89 頁的『[配置 MFT 日誌程式](#)』。

註: 您無法同時對資料庫中的相同綱目執行多個資料庫日誌程式（獨立式或 Java EE）。如果嘗試這麼做，在嘗試將傳送日誌資料寫入資料庫時將會當機。

程序

1. 使用資料庫的適用文件安裝資料庫軟體。
如果 JDBC 支援是資料庫的選用元件，則必須安裝此元件。
2. 執行 **fteCreateLogger** 指令將參數 **-loggerType** 設為 DATABASE，以建立您的獨立式資料庫日誌程式。如需相關資訊，請參閱第 343 頁的『[fteCreate 日誌程式 \(建立 MFT 檔案或資料庫日誌程式\)](#)』。
預設的綱目名稱是 FTELOG。如果使用 FTELOG 以外的綱目名稱，則必須在進行下一步驟前，先編輯所提供且適用於您資料庫的 SQL 檔案 `ftelog_tables_db2.sql` 或 `ftelog_tables_oracle.sql`，以反映此綱目名稱。如需相關資訊，請參閱第 99 頁的『[MFT 日誌程式配置內容](#)』中的 `wmqfte.database.schema`。
3. 使用資料庫的工具，建立所需的資料庫表格。

Multi 在 [Multiplatforms](#) 上，`ftelog_tables_db2.sql` 及 `ftelog_tables_oracle.sql` 檔案包含您可執行以建立表格的 SQL 指令。

z/OS 在 z/OS 上，您需要執行的檔案視您所使用的 Db2 for z/OS 版本而定：

- 若為 Db2 for z/OS 9.0 以及更舊版本，則執行 `ftelog_tables_zos.sql` 檔案來建立表格。此檔案建立表格時，對欄位使用 INTEGER 資料類型來表示傳送的檔案大小，以及與每一項傳送相關聯的表格 ID。
 - 若為 Db2 for z/OS 9.1 以及更新版本，則執行 `ftelog_tables_zos_bigint.sql` 檔案來建立表格。此檔案建立表格時，對欄位使用 BIGINT 資料類型來表示傳送的檔案大小，以及與每一項傳送相關聯的表格 ID。
4. 對日誌程式指令佇列管理程式執行 **fteCreateLogger** 指令所提供的 MQSC 指令，以建立日誌程式佇列。獨立式資料庫日誌程式在協調佇列管理程式上使用兩個佇列。第一個佇列是指令佇列，其中存放用於控制獨立式資料庫日誌程式作業的訊息。此指令佇列的預設名稱是 SYSTEM.FTE.LOG.CMD.*logger_name*。第二個佇列是拒絕佇列。由於獨立式資料庫日誌程式不會捨棄日誌訊息，因此如果日誌程式遇到它無法處理的訊息，它便會將該訊息放在拒絕佇列中以供檢查，有可能進行重新處理。不建議針對此用途使用佇列管理程式的無法傳送的郵件佇列，因為被拒絕的訊息沒有 DLH 標頭，並且不應因其他原因將被拒絕的訊息與無法傳送的郵件佇列中存在的訊息結合在一起。拒絕佇列的預設名稱是 SYSTEM.FTE.LOG.RJCT.*logger_name*。這兩個佇列定義於 **fteCreateLogger** 指令所產生的 MQSC Script 檔中。
 5. 選擇使用者及配置權限
 6. 選擇性的: 您可以編輯步驟第 96 頁的『[2](#)』中 **fteCreateLogger** 指令所建立的 `logger.properties` 檔，以進一步配置獨立式資料庫日誌程式。此檔案是由鍵值組所組成的 Java 內容檔。
`logger.properties` 檔案位於 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 目錄中。如需可用內容及其影響的相關資訊，請參閱第 99 頁的『[MFT 日誌程式配置內容](#)』。

7. **Windows**

- 選擇性的: 如果您使用的是 Windows 系統, 則可以將獨立式資料庫日誌程式作為 Windows 服務來執行。請搭配執行 **fteModifyLogger** 指令與 **-s** 參數。如需相關資訊, 請參閱 [第 415 頁的『fteModify 日誌程式 \(以 Windows 服務方式執行 MFT 日誌程式\)』](#)。
- 選擇性的: 如果使用的資料庫是 Oracle, 或是您要從遠端連接至 Db2 資料庫, 則必須指定日誌程式用來對資料庫伺服器進行鑑別的使用者名稱及密碼。這個使用者名稱及密碼會指定在與 **MQMFTCredentials.xsd** 綱目所定義的格式相符的認證檔案中。如需相關資訊, 請參閱 [第 696 頁的『MFT 認證檔格式』](#)。建立認證檔案後, 您必須使用 **wmqfte.database.credentials.file** 內容, 在 **logger.properties** 檔案中指定認證檔案的位置。
- 使用 **fteStartLogger** 指令來啟動獨立式資料庫日誌程式。依預設, 獨立式資料庫日誌程式會在背景執行, 並且獨立式資料庫日誌程式會將輸出放在 **logs** 目錄的某個檔案中。如果您想要在前景中執行獨立式資料庫日誌程式, 並且同時對主控台以及日誌檔產生輸出, 請在 **fteStartLogger** 指令中加入 **-F** 參數。

如果您在 Windows 上執行前一個步驟, 並且搭配使用 **fteModifyLogger** 指令與 **-s** 參數, 獨立式資料庫日誌程式即會以 Windows 服務方式啟動。

搭配使用 MFT 與遠端資料庫

您可以使用 Managed File Transfer 日誌程式, 與遠端系統上的資料庫進行通訊。

關於這項作業

如果安裝資料庫的機器不同於安裝有 Managed File Transfer 的機器, 則請完成下列步驟。除非另有說明, 否則這些步驟同時適用於 Db2 及 Oracle。

程序

- 在安裝有 Managed File Transfer 的系統上安裝資料庫用戶端。
- 將遠端資料庫伺服器新增至本端資料庫用戶端配置。Managed File Transfer 及 IBM MQ 必須更新此配置, 才能正確存取資料庫。
- 在 **logger.properties** 檔案中指定新內容, 以使用認證檔案連接至資料庫:
wmqfte.database.credentials.file。
註: 舊版的 Managed File Transfer 使用的是內容 **wmqfte.oracle.user** 或 **wmqfte.database.user**, 以及 **wmqfte.oracle.password** 或 **wmqfte.database.password**。這些內容現已淘汰。請改用 **wmqfte.database.credentials.file**。
- 僅限 Oracle: 若要容許遠端連接至資料庫, 請將協調佇列管理程式的 **qm.ini** 檔案中的 **XAResourceManager** 段落變更為以下內容 (確保變更資料庫名稱、使用者名稱及使用者密碼, 以符合自己的資訊):
**Oracle_XA+Acc=P/ftelog/
qgw783jhT+SesTm=35+DB=FTEAUDIT1+SqlNet=FTEAUDIT1+threads=false**, 以粗體強調顯示變更。
- 僅限 Oracle: 使用 **wmqfte.oracle.host** 及 **wmqfte.oracle.port** 內容, 在 **logger.properties** 檔中指定主機及埠。主機及埠的預設值可讓您使用本端資料庫用戶端, 因此如果先前使用了本端資料庫, 您可能並未設定這些值。

相關參考

[第 99 頁的『MFT 日誌程式配置內容』](#)

Managed File Transfer 日誌程式有一組配置內容。您可以在 **logger.properties** 檔案中指定這些內容, 該檔案位於 **MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name** 目錄中。

配置 MFT 獨立式資料庫日誌程式的使用者存取權

在測試環境中, 您可以為一般使用者帳戶新增任何所需的新專用權。在正式作業環境中, 建議您以執行工作所需的最小權限來建立新使用者。

關於這項作業

執行獨立式資料庫日誌程式所需的使用者帳戶的數目及類型，取決於您所使用的系統數目。您可以將獨立式資料庫日誌程式、IBM MQ 及您的資料庫安裝在單一系統上，或安裝在兩個系統上。獨立式資料庫日誌程式必須與 IBM MQ 位於相同的系統上。元件可安裝在下列拓撲中：

獨立式資料庫日誌程式、IBM MQ 及資料庫，全都在相同的系統上

您可以定義單一作業系統使用者，以與這三項元件搭配使用。此配置適用於獨立式資料庫日誌程式。獨立式資料庫日誌程式使用「連結」模式連接至 IBM MQ，連接至資料庫時則使用原生連線。

獨立式資料庫日誌程式及 IBM MQ 位於一個系統上，資料庫則位於個別系統

您可以為此配置建立兩個使用者：在執行獨立式資料庫日誌程式的系統上配置一個作業系統使用者，以及在資料庫伺服器上配置一個具有資料庫遠端存取權的作業系統使用者。此配置適用於使用遠端資料庫的獨立式資料庫日誌程式。獨立式資料庫日誌程式使用「連結」模式連接至 IBM MQ，存取資料庫時則使用用戶端連線。

為方便解說，本文的其餘指示均假設使用者的名稱為 `ftelog`，但您可以使用任何使用者名稱。配置使用者的權限，如下所示：

程序

1. 確保使用者具有讀取權限及執行權限（必要時），並且已將檔案作為 Managed File Transfer Remote Tools and Documentation 安裝的一部分進行安裝。
2. 確保使用者具有在 `logs` 目錄（位於配置目錄）中建立及寫入任何檔案的權限。此目錄用於事件日誌，必要的話還用於診斷追蹤及 FFDC 檔案。
3. 確定使用者具有專屬的群組，且不屬於任何對協調佇列管理程式具備多項權限的群組。使用者不得屬於 `mqm` 群組。在特定平台上，也會對工作人員群組自動指定佇列管理程式存取權；因此獨立式資料庫日誌程式使用者不應屬於工作人員群組。您可以使用「IBM MQ Explorer」，檢視佇列管理程式本身及其內含物件的權限記錄。用滑鼠右鍵按一下物件，然後選取 **物件權限 > 管理權限記錄**。在指令行上，您可以使用 `dspmqaout`（顯示權限）或 `dmpmqaut`（傾出權限）指令。
4. 使用 IBM MQ Explorer 或 `setmqaut`（授與或撤銷權限）指令中的 **管理權限記錄** 視窗，以新增使用者專屬群組的權限（在 UNIX 上，IBM MQ 權限僅與群組相關聯，而不與個別使用者相關聯）。必要權限如下所示：
 - 佇列管理程式上的 `CONNECT` 及 `INQUIRE`（IBM MQ Java 程式庫需要 `INQUIRE` 權限才能運作）。
 - 對 `SYSTEM.FTE` 主題的 `SUBSCRIBE` 權限。
 - 對 `SYSTEM.FTE.LOG.RJCT.logger_name` 佇列的 `PUT` 權限。
 - 對 `SYSTEM.FTE.LOG.CMD.logger_name` 佇列的 `GET` 權限。

指定的拒絕及指令佇列名稱預設名稱。如果您在配置獨立式資料庫日誌程式佇列時選擇了不同的佇列名稱，請改為將權限新增至那些佇列名稱。

5. 執行特定於所用資料庫的使用者配置。
 - 如果您的資料庫是 Db2，請執行下列步驟：

使用 Db2 時可透過數種機制來管理資料庫使用者。下列指示適用於以作業系統使用者為基礎的預設機制。

 - 確保 `ftelog` 使用者不屬於任何 Db2 管理群組（例如 `db2iadm1`、`db2fadm1` 或 `dasadm1`）。
 - 授與使用者連接至資料庫權限，以及對 **步驟 2：建立所需資料庫表格** 中所建立表格的選取、插入及更新權限
 - 如果您的資料庫是 Oracle，請執行下列步驟：
 - 確保 `ftelog` 使用者不在任何 Oracle 管理群組中（例如，Windows 上的 `ora_dba` 或 UNIX 上的 `dba`）
 - 授與使用者連接至資料庫權限，以及對 **步驟 2：建立所需資料庫表格** 中所建立表格的選取、插入及更新權限

MFT 日誌程式配置內容

Managed File Transfer 日誌程式有一組配置內容。您可以在 `logger.properties` 檔案中指定這些內容，該檔案位於 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 目錄中。

若為 IBM WebSphere MQ 7.5 版或更新版本，可在某些表示檔案或目錄位置的 Managed File Transfer 內容中使用環境變數。這可讓執行產品的組件時所使用的檔案或目錄，隨著環境變更（例如執行程序的使用者為何）而改變其所在位置。如需相關資訊，請參閱第 452 頁的『在 MFT 內容中使用環境變數』。

註：在 Windows 上指定檔案路徑時，反斜線 (\) 分隔字元必須呈現為兩條反斜線 (\\)（亦即，跳出 \）。或者，您也可以使用單一正斜線字元 (/) 作為分隔字元。如需 Oracle 中 Java 內容檔字元跳出方式的相關資訊，請參閱 Javadoc 中的 Properties 類別。

連結模式連線內容

內容名稱	說明	預設值
<code>wmqfte.logger.type</code>	使用中的日誌程式類型：檔案或資料庫。請將此值設為 FILE 或 DATABASE。	無預設值
<code>wmqfte.max.transaction.messages</code>	在確定交易之前，於交易中處理的訊息數目上限。在循環式記載模式中，佇列管理程式有固定的空間量可供進行中資料使用。請確保將此內容設定為足夠低的值，以避免耗盡可用空間。	50
<code>wmqfte.max.transaction.time</code>	在兩次交易確定之間經歷的時間長度上限（毫秒）。	5000
<code>wmqfte.max.consecutive.reject</code>	可連續拒絕的訊息數目（亦即，未發現有效訊息）上限。 若超出此數目，則日誌程式推斷問題與訊息本身無關，而與配置有關。例如，若您在資料庫中設定的代理程式名稱直欄不足以容納所有的代理程式名稱，則所有參照代理程式的訊息都會被拒絕。	50
<code>wmqfte.reject.queue.name</code>	日誌程式放置日誌程式無法處理之訊息的佇列名稱。若您有資料庫日誌程式，如需哪些訊息可能會放置到此佇列的詳細資料，請參閱資料庫日誌程式錯誤處理及拒絕。	<code>SYSTEM.FTE.LOG.RJCT.logger_name</code>
<code>wmqfte.command.queue.name</code>	佇列的名稱，日誌程式從中讀取控制其行為的指令訊息。	<code>SYSTEM.FTE.LOG.CMD.logger_name</code>
<code>wmqfte.queue.manager</code>	日誌程式所連接的佇列管理程式。這是必要參數，而且是佇列管理程式的連結模式連線所需要的全部。（如需連接遠端佇列管理程式的內容，請參閱第 104 頁的表 17。）	無預設值
<code>wmqfte.message.source.type</code>	下列其中一個值： automatic subscription 預設值。日誌程式會在 <code>SYSTEM.FTE/Log/#</code> 中定義的佇列管理程式上，建立及使用其專屬的可延續受管理訂閱。此值適用於大多數實務。 administrative subscription 如果自動訂閱不適用，您可以定義不同的訂閱（例如，透過使用「IBM MQ Explorer」、MQSC 或 PCF），並指示日誌程式使用該訂閱。例如，使用此值分割日誌空間，以便其中一個日誌程式處理 A-H 的代理程式，另一個日誌程式處理 I-P 的代理程式，而第三個日誌程式處理 Q-Z 的代理程式。 queue 如果 IBM MQ 拓撲表示不方便為日誌程式建立訂閱，您可以改用佇列。配置 IBM MQ，讓佇列接收通常由 <code>SYSTEM.FTE/Log/#</code> 。	<code>automatic subscription</code>

表 16: 連結模式的連線內容 (繼續)		
內容名稱	說明	預設值
wmqfte.message.source.name	如果訊息來源類型為 administrative subscription 或 queue, 則使用訂閱或佇列名稱。如果來源類型為 automatic subscription, 則會忽略此內容。	無預設值
wmqfte.database.credentials.file	包含用來連接資料庫的使用者名稱和密碼的檔案。 若為 IBM WebSphere MQ 7.5 或更新版本, 此內容值可包含環境變數。 如需相關資訊, 請參閱第 696 頁的『MFT 認證檔格式』。	<p>z/OS 如需建立「鑑別認證檔」的詳細資料, 請參閱 建立 MFT 認證檔</p> <p>ULW 如需此檔案的位置及許可權的相關資訊, 請參閱 配置 MQMFTCredentials.xml。</p> <p>ULW 關於建立「鑑別認證檔」的進一步詳細資料位於 MFT 和 IBM MQ 連線鑑別 中</p>
wmqfte.database.driver	資料庫的 JDBC 驅動程式類別位置。這通常是 JAR 檔的路徑及檔名。例如, AIX 系統上 Db2 的第 2 類驅動程式需要 /opt/IBM/db2/V9.5/java/db2jcc.jar 檔案。在 Windows 系統上, 將路徑分隔字元指定為正斜線字元 (/), 例如 C:/Program Files/IBM/SQLLIB/java/db2jcc.jar。 z/OS 在 z/OS 上, 指定 db2jcc.jar 檔案的完整路徑。例如, wmqfte.database.driver=/db2/db2v10/jdbc/classes/db2jcc.jar。 z/OS 在 z/OS 系統上, 您必須參照下列所有 JAR 檔: <ul style="list-style-type: none"> • db2jcc.jar • db2jcc_license_cisuz.jar • db2jcc_javax.jar 如果資料庫驅動程式包含多個 JAR 檔 (例如, Db2 9.1 需要驅動程式 JAR 檔及授權 JAR 檔), 請在此內容中包含所有這些 JAR 檔。請使用平台適用的類別路徑分隔字元來區隔多個檔名, 亦即: 在 Windows 系統上使用分號字元 (;), 在其他平台上則使用冒號字元 (:).	無預設值
wmqfte.database.exclude_duplicate_metadata	控制是否將項目儲存在 meta 資料表格中, 該表格所含資訊可在資料庫日誌程式綱目內的其他表格中找到。請將此值設定為 true 或 false。依預設不會再儲存這些 meta 資料項目, 因為這會與現有資料重複, 從而浪費資料庫儲存區容量。出現相同資料的內容項目及表格如下: <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 將此內容值設定為 false, 會導致將這些 meta 資料項目儲存在 meta 資料表格中。	true

表 16: 連結模式的連線內容 (繼續)		
內容名稱	說明	預設值
wmqfte.database.host	<p>僅限 Db2:</p> <p>若為 IBM WebSphere MQ 7.5 版或更新版本, 則是要使用第 4 類 JDBC 驅動程式連接的資料庫伺服器的主機名稱。如果指定了此內容的值, 則必須同時指定 <code>wmqfte.database.port</code> 的值。如果兩個內容皆未定義, 則資料庫日誌程式會使用預設 Type 2 JDBC 驅動程式進行連接。</p> <p>如果指定了此內容的值, 則此日誌程式的認證檔案 (由 <code>wmqfte.database.credentials.file</code> 內容定義檔案路徑) 必須存在且可存取, 以定義連接至資料庫的使用者名稱及密碼, 即使資料庫位於本端系統也一樣。</p>	無預設值
wmqfte.database.name	包含 Managed File Transfer 日誌表格的資料庫實例 (使用 Db2 for z/OS 時則為子系統) 名稱。	無預設值
wmqfte.database.type	使用中的資料庫管理系統: Db2 或 Oracle。將此值設定為 <code>db2</code> 或 <code>oracle</code> 。	db2
wmqfte.database.port	<p>僅限 Db2:</p> <p>若為 IBM WebSphere MQ 7.5 版或更新版本, 則是要使用第 4 類 JDBC 驅動程式連接的資料庫伺服器的埠號。如果指定了此內容的值, 則必須同時指定 <code>wmqfte.database.host</code> 的值。如果兩個內容皆未定義, 則資料庫日誌程式會使用預設 Type 2 JDBC 驅動程式進行連接。</p> <p>如果指定了此內容的值, 則此日誌程式的認證檔案 (由 <code>wmqfte.database.credentials.file</code> 內容定義檔案路徑) 必須存在且可存取, 以定義連接至資料庫的使用者名稱及密碼, 即使資料庫位於本端系統也一樣。</p>	無預設值
wmqfte.database.schema	包含 Managed File Transfer 日誌表格的資料庫綱目。在大部分情況下, 預設值即已適用, 但可能需要根據自己網站特定的資料庫考量指定其他替代值。	FTELOG
wmqfte.database.native.library.path	<p>包含您所選資料庫驅動程式所需原生程式庫的路徑 (如果有的話)。例如, AIX 系統上 Db2 的第 2 類驅動程式需要來自 <code>/opt/IBM/db2/V9.5/lib32/</code> 的程式庫。作為此內容的替代方案, 您可以使用其他方法設定 <code>java.library.path</code> 系統內容。</p> <p>在 Solaris 和 HP-UX 系統上, 在執行 fteStartLogger 指令之前, 您還必須設定及匯出 <code>LD_LIBRARY_PATH</code> 環境變數, 以包含此路徑。</p>	無預設值
wmqfte.file.logger.fileDirectory	檔案日誌程式日誌檔所處的目錄。	<code>mqft/logs/coordination_dir/loggers/logger_name/logs</code>
wmqfte.file.logger.fileSize	容許日誌檔增長到的大小上限。此大小值為大於零的正整數, 後面跟著下列其中一個單位: KB、MB、GB、m (分鐘)、h (小時)、d (日)、w (週)。例如, <code>wmqfte.file.logger.fileSize=5MB</code> 指定檔案大小上限為 5MB。 <code>wmqfte.file.logger.fileSize=2d</code> 指定檔案大小上限為 2 天資料。	10MB
wmqfte.file.logger.fileCount	要建立的日誌檔數上限。當資料量超出可儲存於此數目檔案中的數量上限時, 即會刪除最舊的檔案, 從而使檔案數永不超出指定的值。	3

表 16: 連結模式的連線內容 (繼續)

內容名稱	說明	預設值
wmqfte.file.logger.mode	<p>使用中的日誌程式模式：循環或線性。請將此值設為 CIRCULAR 或 LINEAR。</p> <p>CIRCULAR - 檔案日誌程式會將資訊寫入某個檔案中，直到該檔案達到使用</p> <p>wmqfte.file.logger.fileSize 內容定義的大小上限為止。達到大小上限時，檔案日誌程式即會啟動新檔案。以此模式寫入的檔案數上限，受制於使用 wmqfte.file.logger.fileCount 內容定義的值。達到此檔案數上限時，檔案日誌程式即會刪除第一個檔案，然後重建該檔案以用作目前作用中的檔案。如果 wmqfte.file.logger.fileSize 內容中定義的值是固定大小位元組單位（例如 KB、MB 或 GB），則此模式下使用的磁碟空間上限將等於 fileSize 乘以 fileCount。如果 wmqfte.file.logger.fileSize 內容中定義的值是時間單位（例如 m、h、d 或 w），則大小上限將取決於系統中這些時段內的日誌訊息傳輸量。以此模式執行時所使用的日誌檔命名慣例是：<i>Logger_namenumber-timestamp.log</i>，其中：</p> <ul style="list-style-type: none"> • <i>logger_name</i> 是在 fteCreateLogger 指令中指定給日誌程式的名稱。 • <i>number</i> 是檔案集內的檔案數目。 • <i>timestamp</i> 是檔案建立時的時間戳記。 <p>例如： LOGGER1-20111216123430147.log</p> <p>LINEAR - 檔案日誌程式會將資訊寫入某個檔案中，直到該檔案達到使用</p> <p>wmqfte.file.logger.fileSize 內容定義的大小上限為止。達到大小上限時，檔案日誌程式即會啟動新檔案。先前寫入的檔案不會刪除，容許保存作為日誌訊息的歷程記錄。以線性模式執行時不會刪除檔案，因此會忽略</p> <p>wmqfte.file.logger.fileCount 內容，因為可建立的檔案數並沒有上限。由於以此模式執行時並沒有上限，因此必須追蹤日誌檔所使用的磁碟空間量，以避免磁碟空間不足。以此模式執行時所使用的日誌檔命名慣例是：<i>logger_name-timestamp.log</i>，其中：</p> <ul style="list-style-type: none"> • <i>logger_name</i> 是在 fteCreateLogger 指令中指定給日誌程式的名稱。 • <i>timestamp</i> 是檔案建立時的時間戳記。 <p>例如： LOGGER-20111216123430147.log</p>	無預設值
wmqfte.max.retry.interval	<p>日誌程式遇到持續性錯誤時重試間隔的時間上限（以秒為單位）。</p> <p>部分錯誤狀況（如資料庫連線中斷）會使日誌程式無法繼續執行。發生這種情況時，日誌程式會先回復現行交易，然後在一段時間後重試。日誌程式起初等待的時間非常短，以便臨時錯誤能夠得到快速解決。但是，每次日誌程式重試時，其等待時間都會增加。這可避免在錯誤狀況持續較長時間時（如關閉資料庫以進行維護時）執行太多不必要的工作。</p> <p>使用此內容可設定等待時間長度的限制，以便解決錯誤狀況時在合理的時間內進行重試。</p>	600
loggerQMgrRetryInterval	日誌程式的處理程序控制器檢查佇列管理程式可用性的間隔（以秒為單位）。	30
maxRestartCount	在 maxRestartInterval 內容值所指定的時間間隔內，可進行的重新啟動次數上限。超出此值時，日誌程式的處理程序控制器會停止重新啟動日誌程式，並改為根據 maxRestartDelay 內容的值來執行動作。	4

表 16: 連結模式的連線內容 (繼續)		
內容名稱	說明	預設值
maxRestartInterval	日誌程式的處理程序控制器測量日誌程式重新啟動的間隔 (以秒為單位)。若在此間隔內重新啟動次數超出 maxRestartCount 內容的值, 日誌程式的程序控制器會停止重新啟動日誌程式。日誌程式的處理程序控制器會改為根據 maxRestartDelay 內容的值來執行動作。	120
maxRestartDelay	決定當日誌程式重新啟動率超出 maxRestartCount 和 maxRestartInterval 內容的值時, 日誌程式的處理程序控制器的行為。若您指定的值小於或等於零, 日誌程式的處理程序控制器會停止。若您指定的值大於零, 則必須先等待此秒數之後, 再重設日誌程式的處理程序控制器所保留的重新啟動歷程資訊, 以及重新啟動日誌程式。	-1
wmqfte.oracle.port	日誌程式用來連接至 Oracle 實例的埠。此埠也稱為 TNS 接聽器。	1521
wmqfte.oracle.host	日誌程式用來連接至 Oracle 實例的主機。	localhost
armELEMTYPE	選用內容。如果日誌程式配置為由 Automatic Restart Manager (ARM) 重新啟動, 則將此內容設為相關聯 ARM 原則中指定的 ARM ELEMTYPE 參數值。對於日誌程式, 請將 ELEMTYPE 設為 SYSBFLG。	未設定
armELEMENT	選用內容。如果日誌程式配置為由 Automatic Restart Manager (ARM) 重新啟動, 則將此內容設為相關聯 ARM 原則中指定的 ARM ELEMENT 參數值。您可以將 ELEMENT 值設為對應於日誌程式名稱。	未設定
loggerQMGrAuthenticationCredentialsFile	包含 MQ 連線認證 (用於日誌程式的協調佇列管理程式連線) 的檔案路徑。	 如需建立「鑑別認證檔」的詳細資料, 請參閱 建立 MFT 認證檔  如需此檔案的位置及許可權的相關資訊, 請參閱 配置 MQMFTCredentials.xml 。  關於建立「鑑別認證檔」的進一步詳細資料位於 MFT 和 IBM MQ 連線鑑別 中
trace	選用內容。日誌程式啟動時, 在啟用追蹤的情況下執行日誌程式的追蹤指定。追蹤指定是以逗點區隔的類別清單、等於字元及追蹤層次。 例如, com.ibm.wmqfte.databaselogger、com.ibm.wmqfte.databaselogger.operation=all。 您可以使用以冒號區隔的清單來指定多個追蹤指定。 例如: com.ibm.wmqfte.databaselogger=moderate: com.ibm.wmqfte.databaselogger.operation=all	無
traceFiles	選用內容。要保留的追蹤檔總數。此值會套用於日誌程式的程序控制器及其本身。	5
traceSize	選用內容。每一個追蹤檔在追蹤覆蓋至下一個檔案之前的大小上限 (以 MB 為單位)。此值會套用於日誌程式的程序控制器及其本身。	20

用戶端模式連線內容

V 9.0.4

下表顯示支援日誌程式佇列管理程式的用戶端模式連線所需的其他日誌程式內容。

表 17: 用戶端模式的連線內容		
內容名稱	說明	預設值
wmqfte.queue.manager.host	日誌程式佇列管理程式的主機名稱或 IP 位址。	無預設值
wmqfte.queue.manager.port	日誌程式佇列管理程式正在其中進行接聽的埠。	1414
wmqfte.queue.manager.channel	日誌程式佇列管理程式中伺服器連線通道的名稱。	SYSTEM.DEF.SVRCONN
wmqfte.Ssl.CipherSuite	指定日誌程式與日誌程式佇列管理程式如何交換資料的各個 TLS 層面。 wmqfte.Ssl.CipherSuite 的值是 CipherSuite 名稱。此 CipherSuite 名稱對映至日誌程式佇列管理程式通道上使用的 CipherSpec 名稱。 如需相關資訊，請參閱 CipherSuite 及 CipherSpec 名稱對映。	無預設值
wmqfte.Ssl.PeerName	指定必須符合日誌程式佇列管理程式所提供名稱的識別名稱架構。識別名稱用來檢查佇列管理程式在連線時提供的識別憑證。	無預設值
wmqfte.Ssl.TrustStore	指定日誌程式信任的憑證位置。 wmqfte.Ssl.TrustStore 的值是檔案路徑。如果此檔案路徑是 Windows 檔案路徑，則必須將反斜線字元 (\) 跳出為具有進一步的反斜線字元 (\\)。 請注意，此內容值可包含環境變數。	無預設值
wmqfte.Ssl.TrustStoreCredentialsFile	包含 wmqfte.Ssl.TrustStore 認證的檔案路徑。 請注意，此內容值可包含環境變數。	無預設值
wmqfte.Ssl.TrustStoreType	您想要使用的 SSL 金鑰儲存庫類型。支援 JKS 及 PKCS#12 金鑰儲存庫。此內容值可以是 jks 或 pkcs12。	jks
wmqfte.Ssl.KeyStore	指定日誌程式的私密金鑰位置。 wmqfte.Ssl.KeyStore 的值是檔案路徑。如果此檔案路徑是 Windows 檔案路徑，則必須將反斜線字元 (\) 跳出為具有進一步的反斜線字元 (\\)。 請注意，此內容值可包含環境變數。	無預設值
wmqfte.Ssl.KeyStore.CredentialsFile	包含 wmqfte.Ssl.KeyStore 認證的檔案路徑。 請注意，此內容值可包含環境變數。	無預設值
wmqfte.Ssl.KeyStoreType	您想要使用的 SSL 金鑰儲存庫類型。支援 JKS 及 PKCS#12 金鑰儲存庫。此內容值可以是 jks 或 pkcs12。	jks
wmqfte.Ssl.FipsRequired	指定您想要在日誌程式層次上啟用 FIPS 支援。此內容值可以是 true 或 false。如需相關資訊，請參閱第 560 頁的『MFT 中的 FIPS 支援』。	false

相關參考

第 452 頁的『[在 MFT 內容中使用環境變數](#)』

從 IBM WebSphere MQ 7.5 開始，可能會在代表檔案或目錄位置的 Managed File Transfer 內容中使用環境變數。這可讓執行產品的某部分時所使用的檔案或目錄，隨著環境變更而改變其所在位置。例如，執行程序的使用者為何。

第 508 頁的『[MFT 的 SSL 內容](#)』

對 IBM MQ 及 Managed File Transfer 使用 SSL 或 TLS，可避免代理程式與佇列管理程式之間未獲授權的連線，以及加密代理程式與佇列管理程式之間的訊息資料流量。

MFT 獨立式日誌程式的替代配置

一般而言，Managed File Transfer 獨立式日誌程式（無論是檔案或是資料庫類型）會與協調佇列管理程式位於相同的系統上，且會以 IBM MQ 連結模式連接至協調佇列管理程式。不過，也可以與任何可連線至協調佇列管理程式的佇列管理程式，安裝在相同的系統上。獨立式日誌程式可透過本身自動建立的訂閱來接收訊息。這是安裝指示中所述的配置。

但是，如果有網站特定的考量，您可以將獨立式日誌程式配置成以兩種其他方式接收訊息；此配置由 `wmqfte.message.source.type` 內容控制。此內容詳述於[資料庫日誌程式內容](#)中。

管理訂閱

依預設，獨立式日誌程式會使用預設的可延續訂閱選項及受管理訂閱，就 `SYSTEM.FTE/Log/#` 主題建立其專屬訂閱（也就是說，佇列管理程式會控制在將訊息傳遞至應用程式之前用來保留這些訊息的備份佇列）。如果訂閱或佇列上需要使用其他選項，您可以自行建立訂閱、設定您需要的選項，並將獨立式日誌程式配置成改用該訂閱。請記得要新增獨立式日誌程式的權限，才能使用您所建立的訂閱。

舉例來說，使用兩個萬用字元訂閱對日誌空間進行分割、將日誌從名稱以 `FINANCE` 開頭的代理程式傳送至某個資料庫中，以及將日誌從名稱以 `ACCOUNTING` 開頭的代理程式傳送至另一個資料庫中，皆屬於此配置的運用。使用此類型配置需要兩個獨立式日誌程式實例，且兩者分別要有其專屬 `logger.properties` 檔來參照必要的訂閱及其專屬的指令佇列及拒絕佇列。

若要僅收集來自名稱以 `ACCOUNTING` 開頭的代理程式的日誌訊息，請在協調佇列管理程式上以主題字串 `SYSTEM.FTE/Log/ACCOUNTING*` 建立訂閱物件。請將萬用字元用法值設為字元層次萬用字元。您還必須將項目新增至日誌程式的 `logger.properties` 檔。比方說，例如您以這些設定建立稱為 `ACCOUNTING.LOGS` 的訂閱物件，請將下列項目新增至 `logger.properties` 檔：

```
wmqfte.message.source.type=administrative subscription
wmqfte.message.source.name=ACCOUNTING.LOGS
```

獨立式日誌程式只會處理以 `SYSTEM.FTE/Log/` 的主題字串開頭的日誌訊息。您可以指定更為嚴格的主題字串，但不能指定限制較少的字串。如果您在錯誤中指定了較鬆散的限定字串，則所有與 `SYSTEM.FTE/Log/` 以外的主題字串相關的發佈都會移至拒絕佇列，且獨立式日誌程式將會產生錯誤訊息 `BFGDB0002E`。此錯誤訊息意味著獨立式日誌程式配置有問題。

佇列

一般拓撲是指獨立式日誌程式與協調佇列管理程式執行於相同系統的架構。如果這種情況不可行，您可以在協調佇列管理程式上使用其他佇列管理程式上的佇列建立訂閱，作為訂閱目的地（使用遠端佇列定義，或者使用訂閱的 `DESTQMGR` 內容）。然後，日誌程式將可在管理第二個佇列管理程式的系統上執行，並讀取來自佇列的訊息。為確保交易完整性，獨立式日誌程式必須一律以連結模式連接至其佇列管理程式。您必須在獨立式日誌程式所連接的相同佇列管理程式上，定義拒絕佇列及指令佇列。佇列管理程式必須使用 IBM WebSphere MQ 7.5 或更新版本。

例如，若要收集訂閱放置在佇列 `USER.QUEUE` 上的日誌訊息，請將這些項目新增至 `logger.properties` 檔：

```
wmqfte.message.source.type=queue
wmqfte.message.source.name=USER.QUEUE
```

安裝 MFT 的 Java EE 資料庫日誌程式

請遵循下列指示來安裝及配置 Java EE 資料庫日誌程式，以與 Managed File Transfer 搭配使用。

關於這項作業

如需 Java EE 資料庫日誌程式的相關資訊，請參閱第 89 頁的『[配置 MFT 日誌程式](#)』主題。

註: Java EE 資料庫日誌程式無法與獨立式日誌程式同時執行, 除非這些日誌程式使用的是不同的資料庫實例。

程序

1. 在安裝 Java EE 資料庫日誌程式之前, 必須先準備環境。請遵循第 106 頁的『[準備安裝 MFT 的 Java EE 資料庫日誌程式](#)』主題中的指示作業。
2. 在符合 Java Platform, Enterprise Edition (Java EE) 標準的應用程式伺服器中安裝 Java EE 資料庫日誌程式。如需相關指示, 請參閱下列主題:
 - 第 109 頁的『[使用 WebSphere Application Server 7.0 安裝 MFT 的 Java EE 資料庫日誌程式](#)』
 - 第 112 頁的『[使用 WebSphere Application Server Community Edition 來安裝 MFT 的 Java EE 資料庫日誌程式](#)』

相關工作

第 106 頁的『[準備安裝 MFT 的 Java EE 資料庫日誌程式](#)』

在安裝 Java EE 資料庫日誌程式之前, 請遵循下列指示來準備 Managed File Transfer 環境。

第 109 頁的『[使用 WebSphere Application Server 7.0 安裝 MFT 的 Java EE 資料庫日誌程式](#)』

遵循下列指示, 使用 WebSphere Application Server 7.0 來安裝及配置 Managed File Transfer 的 Java Platform, Enterprise Edition (Java EE) 資料庫日誌程式。

第 112 頁的『[使用 WebSphere Application Server Community Edition 來安裝 MFT 的 Java EE 資料庫日誌程式](#)』

遵循下列指示, 使用 WebSphere Application Server Community Edition 來安裝及配置 Managed File Transfer 的 Java Platform, Enterprise Edition (Java EE) 資料庫日誌程式。

第 115 頁的『[配置 MFT 的 Java EE 資料庫日誌程式的使用者存取權](#)』

當您配置 Managed File Transfer 的 Java Platform, Enterprise Edition (Java EE) 資料庫日誌程式時, 您需要使用者帳戶才能存取 IBM MQ、資料庫及作業系統。所需的作業系統使用者數目, 取決於您用來管理這些元件的系統數目。

第 117 頁的『[從獨立式資料庫日誌程式移轉至 MFT 的 Java EE 資料庫日誌程式](#)』

您可以從獨立式資料庫日誌程式移轉至 Java EE 資料庫日誌程式。必須停止獨立式資料庫日誌程式, 然後再安裝 Java EE 資料庫日誌程式。若要避免日誌項目遺失或重複, 您必須在停止獨立式資料庫日誌程式之前, 阻止訊息被發佈至 SYSTEM.FTE 主題, 然後在 JEE 資料庫日誌程式安裝完成之後再加以重新啟動。移轉之前, 請先備份資料庫。

相關參考

第 304 頁的『[MFT 日誌程式的權限](#)』

執行日誌程式的作業系統使用者需要對日誌程式佇列及 SYSTEM.FTE 主題具有特定 IBM MQ 權限。

準備安裝 MFT 的 Java EE 資料庫日誌程式


在安裝 Java EE 資料庫日誌程式之前, 請遵循下列指示來準備 Managed File Transfer 環境。

關於這項作業

如需 Java EE 資料庫日誌程式的相關資訊, 請參閱第 89 頁的『[配置 MFT 日誌程式](#)』主題。

程序

1. 使用資料庫的適用文件安裝資料庫軟體。
如果 JDBC 支援是資料庫的選用元件, 則必須安裝此元件。
2. 使用您的資料庫提供的工具來建立資料庫。資料庫必須具有至少 8K 分頁大小的表格空間及緩衝池。
預設的綱目名稱是 FTELOG。如果您使用 FTELOG 以外的綱目名稱, 必須先編輯您的資料庫所適用的 SQL 檔 `ftelog_tables_db2.sql` 或 `ftelog_tables_oracle.sql` 予以反映, 才能繼續下一步。
3. 使用資料庫的工具, 建立所需的資料庫表格。

 在 Multiplatforms 上, `ftelog_tables_db2.sql` 及 `ftelog_tables_oracle.sql` 檔案包含您可執行以建立表格的 SQL 指令。

在 z/OS 上，您需要執行的檔案視您所使用的 Db2 for z/OS 版本而定：

- 若為 Db2 for z/OS 9.0 以及更舊版本，則執行 `ftelog_tables_zos.sql` 檔案來建立表格。此檔案建立表格時，對欄位使用 `INTEGER` 資料類型來表示傳送的檔案大小，以及與每一項傳送相關聯的表格 ID。
 - 若為 Db2 for z/OS 9.1 以及更新版本，則執行 `ftelog_tables_zos_bigint.sql` 檔案來建立表格。此檔案建立表格時，對欄位使用 `BIGINT` 資料類型來表示傳送的檔案大小，以及與每一項傳送相關聯的表格 ID。
4. 如果您將綱目名稱變更為 `FTELOG` 以外的名稱，則必須在 `EAR` 檔中變更綱目名稱。如需相關資訊，請參閱第 107 頁的『變更 MFT 的 Java EE 資料庫日誌程式中的綱目名稱』。
 5. 在 IBM MQ 中建立拒絕佇列。
因為日誌程式一律不捨棄日誌訊息，所以日誌程式在遇到無法處理的訊息時，會將該訊息放置在拒絕佇列上，以進行檢查且可能會重新處理。請不要將佇列管理程式的「無法傳送的郵件」佇列用於此用途，因為拒絕的訊息並沒有 `DLH` 標頭，且不可與因為其他原由而放置到「無法傳送的郵件」佇列的訊息結合。`fteCreateLogger` 指令會建立拒絕佇列。此拒絕佇列的預設名稱是 `SYSTEM.FTE.LOG.RJCT.logger_name`
 6. 請遵循配置 JEE 日誌程式的使用者存取權主題中的指示作業。

下一步

現在您可以將 Java EE 資料庫日誌程式安裝在符合 Java EE 標準的應用程式伺服器中。請根據您所使用的應用程式伺服器，遵循下列主題中的指示作業：

- 第 109 頁的『使用 WebSphere Application Server 7.0 安裝 MFT 的 Java EE 資料庫日誌程式』
- 第 112 頁的『使用 WebSphere Application Server Community Edition 來安裝 MFT 的 Java EE 資料庫日誌程式』

變更 MFT 的 Java EE 資料庫日誌程式中的綱目名稱

Java Platform, Enterprise Edition (Java EE) 資料庫日誌程式可使用具有非預設綱目名稱的資料庫。您必須變更 Java EE 資料庫日誌程式 `EAR` 檔中的綱目名稱。

關於這項作業

若要變更 Java EE 資料庫日誌程式使用的綱目名稱，請完成下列步驟：

程序

1. 使用下列指令，從 `EAR` 檔中解壓縮 `JPA JAR` 檔：

```
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`，視您使用的是 Db2 還是 Oracle 而定。
 - `jpa_file` 是 `com.ibm.wmqfte.web.jpa.oracle.jar` 或 `com.ibm.wmqfte.web.jpa.jar`，視您使用的是 Db2 還是 Oracle 而定。
2. 使用下列指令，從 `JPA JAR` 檔解壓縮 `persistence.xml` 檔：

```
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`，視您使用的是 Db2 還是 Oracle 而定。
3. 編輯 `persistence.xml` 檔，以變更下列字行：

```
<property name="openjpa.jdbc.Schema" value="schema_name"/>
```

其中

- *schema_name* 是您想要使用的綱目名稱。

4. 使用下列指令，以修改過的 *persistence.xml* 檔更新 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*，視您使用的是 Db2 還是 Oracle 而定。

5. 使用下列指令，以修改過的 JPA JAR 檔更新 EAR 檔：

```
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*，視您使用的是 Db2 還是 Oracle 而定。
- *jpa_file* 是 *com.ibm.wmqfte.web.jpa.oracle.jar* 或 *com.ibm.wmqfte.web.jpa.jar*，視您使用的是 Db2 還是 Oracle 而定。

下一步

使用修改過的 EAR 檔來安裝 Java EE 資料庫日誌程式。

相關工作

第 109 頁的『[使用 WebSphere Application Server 7.0 安裝 MFT 的 Java EE 資料庫日誌程式](#)』

遵循下列指示，使用 WebSphere Application Server 7.0 來安裝及配置 Managed File Transfer 的 Java Platform, Enterprise Edition (Java EE) 資料庫日誌程式。

第 112 頁的『[使用 WebSphere Application Server Community Edition 來安裝 MFT 的 Java EE 資料庫日誌程式](#)』

遵循下列指示，使用 WebSphere Application Server Community Edition 來安裝及配置 Managed File Transfer 的 Java Platform, Enterprise Edition (Java EE) 資料庫日誌程式。

在 WebSphere Application Server 7.0 中設定原生程式庫路徑

如果您在 WebSphere Application Server 7.0 上部署 Java Platform, Enterprise Edition (Java EE) 資料庫日誌程式應用程式，並且想要在應用程式與 IBM MQ 之間使用連結模式連線，則必須使用系統上 IBM MQ 原生程式庫的位置來配置 IBM MQ 傳訊提供者。

關於這項作業

如果您未在應用程式伺服器中設定原生程式庫路徑，則可能會在 WebSphere Application Server 7.0 系統輸出日誌中收到下列錯誤訊息：

```
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]
```

請使用 WebSphere Application Server 7.0 管理主控台來完成下列步驟：

程序

1. 在導覽窗格中，展開 **資源 > JMS > JMS 提供者**。
2. 選取位於 Connection Factory 之正確範圍的 IBM MQ 傳訊提供者，或建立連結模式連線的啟動規格。

註：使用 Server 範圍的原生路徑資訊勝過更高範圍的原生路徑資訊，而使用 Node 範圍的原生路徑資訊則勝過 Cell 範圍的原生路徑資訊。

3. 在「一般內容」下，於**原生程式庫路徑**欄位中輸入包含 IBM MQ 原生程式庫的目錄完整名稱。
例如，在 Linux 上，輸入 /opt/mqm/java/lib。只輸入一個目錄名稱。
4. 按一下**確定**。
5. 重新啟動應用程式伺服器以，重新整理配置。
6. 必要性的: 再次重新啟動應用程式伺服器，以載入程式庫。

使用 WebSphere Application Server 7.0 安裝 MFT 的 Java EE 資料庫日誌程式

遵循下列指示，使用 WebSphere Application Server 7.0 來安裝及配置 Managed File Transfer 的 Java Platform, Enterprise Edition (Java EE) 資料庫日誌程式。

開始之前

安裝 JEE 資料庫日誌程式應用程式之前，請遵循第 106 頁的『準備安裝 MFT 的 Java EE 資料庫日誌程式』及第 108 頁的『在 WebSphere Application Server 7.0 中設定原生程式庫路徑』主題中的指示。

關於這項作業

如需 Java EE 資料庫日誌程式的相關資訊，請參閱第 89 頁的『配置 MFT 日誌程式』。

程序

1. 設定 XA JDBC 提供者:
 - a) 從 WebSphere Application Server 7.0 管理主控台導覽中選取 **資源 > JDBC > JDBC 提供者**。
 - b) 使用主控台精靈，按一下**新建**，以建立 JDBC 提供者。
 - c) 在精靈的步驟 1 中，從**資料庫類型**清單中選取您所使用的資料庫，然後從**提供者類型**清單中選取關聯的提供者類型。從**實作類型**清單中，選取 **XA 資料來源**。按下一步。
 - d) 在精靈的步驟 2，確定所需的資料庫 Jar 檔的目錄位置已正確設定。按下一步。
 - e) 在摘要頁面上按一下**完成**，以建立 JDBC 提供者。
2. 建立鑑別別名。您可以分別為資料來源及 IBM MQ 建立不同的別名:
 - a) 從 WebSphere Application Server 7.0 管理主控台導覽中選取 **安全 > 廣域安全**。
 - b) 在**鑑別**標題下，展開 **Java 鑑別和授權服務**。
 - c) 按一下 **J2C 鑑別資料**。鑑別別名頁面即會開啟。
 - d) 建立資料來源的鑑別別名:
 - i) 按一下**新建**。
 - ii) 輸入**別名、使用者 ID、密碼及說明**的詳細資料。在**使用者 ID 及密碼**欄位中輸入的詳細資料，必須符合您建立資料庫使用者時所輸入的詳細資料。如需相關資訊，請參閱配置 JEE 資料庫日誌程式的使用者存取權。
 - iii) 按一下**確定**。
 - e) 建立 IBM MQ 的鑑別別名:
 - i) 按一下**新建**。
 - ii) 輸入**別名、使用者 ID、密碼及說明**的詳細資料。在**使用者 ID 及密碼**欄位中輸入的詳細資料，必須符合 IBM MQ 安裝的使用者及密碼設定。
 - iii) 按一下**確定**。
3. 建立資料來源:
 - a) 從 WebSphere Application Server 7.0 管理主控台導覽中選取 **資源 > JDBC > 資料來源**。
 - b) 選取**範圍**下拉清單，然後將範圍變更為適當的值。例如，Node=yourNode, Server=yourServer。
 - c) 使用主控台精靈，按一下**新建**，以建立資料來源。
 - d) 在精靈的步驟 1，在**資料來源名稱**欄位中輸入 wmqfte-database，然後在 **JNDI 名稱**欄位中輸入 jdbc/wmqfte-database。按下一步。

- e) 在精靈的步驟 2 中，使用**選取現有的 JDBC 提供者**下拉清單，以選取上述步驟所建立的 JDBC 提供者。按**下一步**。
 - f) **Db2**：在精靈的步驟 3，在**驅動程式類型**欄位中輸入 4。
 - g) **Db2**：在**資料庫名稱**、**伺服器名稱**及**埠號**欄位中輸入詳細資料，然後按**下一步**。
Oracle：在 **URL** 欄位中輸入連線 URL，然後在**資料庫儲存庫 Helper 類別名稱**欄位中選擇正確的資料儲存庫 Helper。
Oracle RAC：在連接至 Oracle Real Application Cluster 時，連線 URL 必須包括連接至所有可用資料庫實例所需的主機資訊。
 - h) 在精靈的步驟 4 中，從 **XA 回復的鑑別別名**清單中選取您在步驟 2d 中定義的資料來源鑑別別名的名稱。請在**元件管理的鑑別別名**及**儲存器管理的鑑別別名**清單中，選取相同的名稱。
 - i) 在摘要頁面上按一下**完成**，以建立資料來源。
4. 選擇性的：驗證資料來源的配置：
- a) 從 WebSphere Application Server 7.0 管理主控台導覽中選取 **資源 > JDBC > 資料來源**。
 - b) 按一下**測試連線**按鈕。
5. 建立主題。
- a) 從 WebSphere Application Server 7.0 管理主控台導覽中，按一下**資源 > JMS > 主題**。
 - b) 選取**範圍**下拉清單，然後將範圍變更為適當的值。例如，Node=yourNode，Server=yourServer。
 - c) 按一下**新建**。
 - d) 按一下 **IBM MQ 傳訊提供者**。
 - e) 在主題中「內容」頁的「管理」畫面上，為**名稱**及 **JNDI 名稱**欄位選擇唯一值；您在後續配置時將參考這些資訊。
 - f) 在「**IBM MQ 主題**」畫面的**主題名稱**欄位中，輸入 SYSTEM.FTE/Log/#。
6. 建立啟動規格：
- a) 從 WebSphere Application Server 7.0 管理主控台導覽中，按一下**資源 > JMS > 啟動規格**。
 - b) 選取**範圍**下拉清單，然後將範圍變更為適當的值。例如，Node=yourNode，Server=yourServer。
 - c) 按一下**新建**。
 - d) 按一下 **IBM MQ 傳訊提供者**。
 - e) 在精靈的步驟 1 中，為**名稱**及 **JNDI 名稱**欄位選擇唯一值；您在後續配置時將再次參考這些資訊。
 - f) 在步驟 1.1 中，將您在步驟 5 中所設定主題的 JNDI 名稱輸入目的地 **JNDI 名稱**欄位中。
 - g) 從**目的地類型**清單中，選取**主題**。
 - h) 在精靈的步驟 1.2 中，選取**可延續訂閱**。在**訂閱名稱**欄位中，輸入 SYSTEM.FTE.DATABASELOGGER.AUTO。
 - i) 在精靈的步驟 2 中，選取**將所有必要資訊輸入此精靈中**。
 - j) 在步驟 2.1 中，將您的佇列管理程式名稱輸入**佇列管理程式或佇列共用群組名稱**欄位中。
 - k) 在步驟 2.2 中，從**傳輸**清單中選取您所選擇的傳輸方法。如果您選取**連結**，則不需要提供其他資訊。如果您選取**用戶端**或**先連結後用戶端**，請輸入**主機名稱**、**埠**及**伺服器連線通道**的詳細資料。
 - l) 選擇性的：按一下**測試連線**以確認佇列管理程式是否存在。不過，在步驟 6n 中參照鑑別別名之前，您應該會收到 NOT_AUTHORIZED。
 - m) 按一下**儲存**。
 - n) 按一下您建立的「啟動規格」名稱。在 **配置** 標籤的**一般內容** 區段中，向下捲動至「**進階**」畫面，並在**用戶端 ID** 欄位中輸入唯一名稱以識別您的 IBM MQ 連線。您必須完成此步驟，否則 IBM MQ 會拒絕您的連線，並顯示 JMSCC0101 錯誤碼。
 - o) 如果您選擇**用戶端**作為傳輸方法，請向下捲動至「**安全設定**」畫面，然後從**鑑別別名**清單中選取您在步驟 8 中定義的鑑別別名。

- p) 按一下套用。
- q) 在配置標籤的「其他內容」區段中，按一下進階內容。在「進階內容」畫面的「連線消費者」區段中，在伺服器階段作業數上限欄位中輸入 1。
註：請確保完成此步驟，然後再繼續。若未完成此步驟，日誌程式將無法正確運作。
- r) 在配置標籤的「其他內容」區段中，按一下進階內容。將如果訊息遞送失敗即停止端點的值設為最小值 1。

如果 `_numberOfFailedAttemptsBeforeReject` 內容的值設為大於 1 的值（如需相關資訊，請參閱 9j），則如果訊息遞送失敗即停止端點至少應設為 `_numberOfFailedAttemptsBeforeReject` 內容的值。這可以在接收到無法處理的訊息（例如，形態異常的傳送日誌訊息）時防止端點停止。如需相關資訊，請參閱第 278 頁的『MFT 日誌程式錯誤處理及拒絕』。

7. 建立 Queue Connection Factory。

- a) 從 WebSphere Application Server 7.0 管理主控台導覽中，按一下資源 > JMS > Queue Connection Factory。
- b) 選取範圍下拉清單，然後將範圍變更為適當的值。例如，Node=yourNode，Server=yourServer。
- c) 按一下新建。
- d) 按一下 IBM MQ 傳訊提供者。
- e) 在精靈的步驟 1 中，為名稱及 JNDI 名稱欄位選擇唯一值；您在後續配置時將再次參考這些資訊。
- f) 在步驟 2 中，選取將所有必要資訊輸入此精靈中。
- g) 在步驟 2.1 中，將您的佇列管理程式名稱輸入佇列管理程式或佇列共用群組名稱欄位中。
- h) 在步驟 2.2 中，從傳輸清單中選取您所選擇的傳輸方法。如果您選取連結，則不需要提供其他資訊。如果您選取用戶端或先連結後用戶端，請輸入主機名稱、埠及伺服器連線通道的詳細資料。
- i) 選擇性的：按一下測試連線以確認佇列管理程式是否存在。不過，在步驟 7h 中參照鑑別別名之前，您應該會收到 NOT_AUTHORIZED。
- j) 如果您選取用戶端或先連結後用戶端作為傳輸方法，請按一下您剛才建立的 Queue Connection Factory 名稱。向下捲動至配置標籤的「安全設定」畫面，然後從 **XA** 回復的鑑別別名及儲存器管理的鑑別別名清單中，選取您在步驟 2e 中定義的鑑別別名。

8. 在 WebSphere Application Server 中建立拒絕佇列：

- a) 從 WebSphere Application Server 7.0 管理主控台導覽中，按一下資源 > JMS > 佇列。
- b) 選取範圍下拉清單，然後將範圍變更為適當的值。例如，Node=yourNode，Server=yourServer。
- c) 按一下新建。
- d) 按一下 IBM MQ 傳訊提供者。
- e) 為名稱及 JNDI 名稱欄位選擇唯一值；您在後續配置時將再次參考這些資訊。
- f) 在佇列名稱欄位中輸入 `SYSTEM.FTE.LOG.RJCT.logger_name`。請確定您已在協調佇列管理程式上建立此佇列。
- g) 在佇列管理程式名稱欄位中，輸入您的佇列管理程式名稱。
- h) 按一下確定。

9. 安裝 JEE 資料庫日誌程式應用程式：

- a) 從 WebSphere Application Server 7.0 管理主控台，選取應用程式 > 新建應用程式。
- b) 選取範圍下拉清單，然後將範圍變更為適當的值。例如，Node=yourNode，Server=yourServer。
- c) 從選項清單中，選取新建企業應用程式。
- d) 在「準備安裝應用程式」頁面上，從 Managed File Transfer Service 安裝架構的 `MQ_INSTALLATION_PATH/mqft/web` 目錄中選取 `com.ibm.wmqfte.databaselogger.jee.ear` 檔或 `com.ibm.wmqfte.databaselogger.jee.oracle.ear` 檔，然後按下一步。

- e) 在下列畫面上，選取**詳細**以顯示所有安裝選項及參數，然後按下一步。
- f) 在精靈步驟 1 至 4 中按下一步，以接受預設值。
- g) 在精靈的步驟 5 **連結訊息驅動 Bean 的接聽器**中，捲動至**接聽器連結**區段。按一下**啟動規格**。
輸入下列欄位的必要值：

目標資源 JNDI 名稱

您在步驟 6d 中建立啟動規格時所指定的 JNDI 名稱。

目的地 JNDI 名稱

您在步驟 5d 中建立主題時所指定的 JNDI 名稱。

按下一步。

- h) 在精靈的步驟 6 **將資源參照對映至資源**中，在**目標資源 JNDI 名稱**欄位中輸入詳細資料。此名稱是您在步驟 7c 中為拒絕 Queue Connection Factory 指定的 JNDI 名稱。按下一步。
- i) 在精靈的步驟 7 **將資源環境項目參照對映至資源**中，在**目標資源 JNDI 名稱**欄位中輸入詳細資料。此名稱是您在步驟 8d 中所建立的拒絕佇列的 JNDI 名稱。按下一步。
- j) 在精靈的步驟 8 **對映 EJB 模組的環境項目**中，接受預設值 1。按下一步。

Oracle RAC: 連接至 Oracle Real Application Cluster 時，您必須將

`_numberOfFailedAttemptsBeforeReject` 內容的值至少設為 2。此內容決定日誌程式在失敗之後嘗試處理審核訊息的次數。在資料庫失效接手的情況下，通常至少發生一次失敗。為避免無謂地將訊息移至拒絕佇列，您可以增加此值以容許第二次嘗試，如此通常即可在新資料庫實例的連線建立時成功處理。如果您在測試期間發現訊息在資料庫實例的失效接手期間仍移至拒絕佇列，請進一步提高此值：實例切換的時機若不湊巧，相同訊息就可能失敗多次。不過，請注意，提高此值將會對所有失敗案例造成影響（例如，形態異常的訊息），而非只影響資料庫失效接手，因此請謹慎地提高此值，以避免發生不必要的重試。

- k) 在精靈的步驟 9 **模組的 meta 資料**中，按下一步。

- l) 在精靈的步驟 10 **摘要**中，按一下**完成**。

10. 您現在可以從 WebSphere Application Server 7.0 管理主控台中啟動該應用程式：

- a) 從主控台導覽中，選取**應用程式 > 應用程式類型 > WebSphere 企業應用程式**。
- b) 從收集表格中選取**日誌程式企業應用程式**的勾選框，然後按一下**啟動**。

使用 *WebSphere Application Server Community Edition* 來安裝 *MFT* 的 *Java EE* 資料庫日誌程式

遵循下列指示，使用 WebSphere Application Server Community Edition 來安裝及配置 Managed File Transfer 的 Java Platform, Enterprise Edition (Java EE) 資料庫日誌程式。

開始之前

安裝 Java EE 資料庫日誌程式應用程式之前，請遵循第 106 頁的『準備安裝 MFT 的 Java EE 資料庫日誌程式』主題中的指示。

關於這項作業

如需 Java EE 資料庫日誌程式的相關資訊，請參閱主題 第 89 頁的『配置 MFT 日誌程式』。

程序

1. 部署 IBM MQ 資源配接器 `wmq.jmsra.rar`。

- 若要使用協調佇列管理程式 `QM_JUPITER` 為 JEE 資料庫日誌程式部署 IBM MQ 資源配接器，請執行下列步驟。如果 WebSphere Application Server Community Edition 實例與您要連接的 IBM MQ 佇列管理程式皆在相同系統上執行，則適用此範例。

- a. 建立計劃檔案以定義 MFT 協調佇列管理程式的連線。下列範例計劃檔案會定義稱為 `QM_JUPITER` 的佇列管理程式連線，以及該佇列管理程式上稱為 `SYSTEM.FTE.LOG.RJCT.LOGGER1` 的佇列參照。

```
<?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>

```

若要在您的環境中使用此計劃檔案，請將 QM_JUPITER 變更為您的協調佇列管理程式的名稱。

- b. 開啟 WebSphere Application Server CE 管理主控台。
 - c. 從「歡迎使用頁面」上的**共用主控台動作**清單中，按一下**部署新的應用程式 > 新部署**。
 - d. 在**保存**欄位中，輸入 `mq_install_root/java/lib/jca/wmq.jmsra.rar`
 - e. 在**計劃**欄位中，鍵入您在步驟 1a 中建立的計劃檔案的路徑。
- 如果 WebSphere Application Server Community Edition 實例與您要連接的 IBM MQ 佇列管理程式不在相同系統上執行，請執行下列步驟以部署 IBM MQ 資源配接器。
 - a. 建立一個計劃檔案，以定義 WMQFTE 協調佇列管理程式的連線。下列範例計劃檔案會定義佇列管理程式 QM_SATURN（與 WebSphere Application Server Community Edition 安裝位於不同的系統上）的連線，以及該佇列管理程式上稱為 SYSTEM.FTE.LOG.RJCT.LOGGER1 的佇列參照。QM_SATURN 的主機名稱是 saturn.example.com。QM_SATURN 的埠是 1415。QM_SATURN 的通道是 SYSTEM.DEF.SVRCONN。

因為應用程式伺服器與佇列管理程式位於不同的系統上，所以您必須使用用戶端模式的佇列管理程式連線。下列計劃檔案會將具有名稱 `transportType` 的 `<config-property-setting>` 元素值設為 `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>
<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>

```

若要在您的環境中使用此計劃檔案，請將 QM_SATURN 變更為您的協調佇列管理程式的名稱。請將主機名稱、埠及通道的值，變更為您的協調佇列管理程式的值。

- b. 將 `mq_install_root/java/lib/jca/wmq.jmsra.rar` 檔案從 IBM MQ 安裝所在的系統，複製到 WebSphere Application Server CE 安裝所在的系統。
 - c. 開啟 WebSphere Application Server CE 管理主控台。
 - d. 從「歡迎使用頁面」上的共用主控台動作清單中，按一下部署新的應用程式 > 新部署。
 - e. 在保存欄位中，鍵入您所取得的 `wmq.jmsra.rar` 檔案副本的路徑。
 - f. 在計劃欄位中，鍵入您所建立的計劃檔案的路徑。
2. 您必須定義資料庫連接器，JEE 資料庫日誌程式應用程式才能從 WebSphere Application Server Community Edition 環境內存取所需的資料庫。

從 WebSphere Application Server Community Edition 管理主控台中執行下列步驟：

 - a) 視您使用的 WebSphere Application Server Community Edition 層次而定，從主控台導覽中，選取服務 > 資料庫儲存區，或選取資源 > 資料來源。
 - b) 使用 Geronimo 資料庫儲存區精靈，建立資料庫儲存區。在資料庫儲存區的名稱欄位中，鍵入 `jdbc/wmqfte-database`。
 - c) 根據您的資料庫，將 DB2 XA 或 Oracle Thin 選取為資料庫類型。
 - d) 按下一步。
 - e) 在驅動程式 JAR 欄位中，選取資料庫的適當 JAR。
 - f) 在資料庫名稱欄位中，鍵入您所連接以取得傳送狀態資訊的資料庫名稱。
 - g) 在使用者名稱欄位中，鍵入您在連接資料庫以及對其進行鑑別時所使用的使用者名稱。
 - h) 在密碼及確認密碼欄位中，鍵入對您的資料庫進行鑑別時所使用的密碼。
 - i) 在埠號欄位中，鍵入您所使用的埠號（如果不是預設埠）。
 - j) 確認驅動程式類型的值是 4。
 - k) 從交易類型清單中選取 XA。
 - l) 按一下部署。
 3. 為您的環境更新 Managed File Transfer JEE 資料庫日誌程式應用程式 `openejb-jar.xml` 檔。請使用 Java SDK Jar 公用程式來完成下列步驟：
 - a) 執行下列指令，從提供的 EAR 檔解壓縮 EJB JAR 檔：

```
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`，視您使用的是 Db2 還是 Oracle 而定。EAR 檔位於 IBM WebSphere MQ File Transfer Edition Server 安裝的 `MQ_INSTALLATION_PATH/mqft/web` 目錄中。

- b) 執行下列指令，從先前解壓縮的 EJB JAR 檔 `com.ibm.wmqfte.databaselogger.jee.ejb.jar` 中解壓縮 META-INF/openejb-jar.xml 檔：

```
jar -xf com.ibm.wmqfte.databaselogger.jee.ejb.jar META-INF/openejb-jar.xml
```

- c) 使用文字編輯器，對解壓縮的 META-INF/openejb-jar.xml 檔進行編輯。將下列 `activation-config-property` 值變更為您的環境所適用的值：

queueManager

JEE 資料庫日誌程式使用的 IBM MQ 佇列管理程式的名稱。

hostName

用來連接指定 IBM MQ 佇列管理程式的主機名稱。如果您已在連結模式下連接至佇列管理程式，則不需要此值。

transportType

以用戶端模式還是連結模式連接至指定的 IBM MQ 佇列管理程式。

port

如果您指定連結的 **transportType**，則不需要。用來連接至指定的 IBM MQ 佇列管理程式的埠。

channel

如果您指定連結的 **transportType**，則不需要。用來連接至指定的 IBM MQ 佇列管理程式的伺服器通道。

- d) 執行下列指令，以修改過的 META-INF/openejb-jar.xml 檔更新 EJB JAR 檔：

```
jar -uf com.ibm.wmqfte.databaselogger.jee.ejb.jar META-INF/openejb-jar.xml
```

- e) 執行下列指令，以更新過的 EJB JAR 檔更新提供的 EAR 檔：

```
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`，視您的資料庫而定。

4. 若要將 EAR 檔部署至應用程式伺服器，請從 WebSphere Application Server Community Edition 管理主控台完成下列步驟。

- 從主控台導覽功能表中，選取 **應用程式 > 新部署**。
- 在 **保存** 欄位中指定 EAR 檔：`com.ibm.wmqfte.databaselogger.jee.ear` 或 `com.ibm.wmqfte.databaselogger.jee.oracle.ear`，視您的資料庫而定。
- 將 **計劃** 欄位保留為空白。
- 確定已選取在 **安裝後啟動應用程式** 方框。
- 按一下 **安裝**。JEE 資料庫日誌程式應用程式即會安裝並且啟動。

配置 MFT 的 Java EE 資料庫日誌程式的使用者存取權

當您配置 Managed File Transfer 的 Java Platform, Enterprise Edition (Java EE) 資料庫日誌程式時，您需要使用者帳戶才能存取 IBM MQ、資料庫及作業系統。所需的作業系統使用者數目，取決於您用來管理這些元件的系統數目。

關於這項作業

執行 Java EE 資料庫日誌程式所需的使用者帳戶的數目及類型，取決於您使用的系統數目。使用者帳戶需要存取下列三種環境：

- 本端作業系統
- IBM MQ
- 資料庫

您可以將 Java EE 資料庫日誌程式、IBM MQ 及您的資料庫安裝在單一系統上，或安裝在數個系統上。元件可安裝在下列範例拓撲中：

Java EE 資料庫日誌程式、IBM MQ 及資料庫全都在同一系統上

您可以定義單一作業系統使用者，以與這三項元件搭配使用。日誌程式使用「連結」模式連接至 IBM MQ，連接至資料庫時則使用原生連線。

Java EE 資料庫日誌程式和 IBM MQ 位於一個系統上，資料庫則位於另一個系統上

您可以為此配置建立兩個使用者：在執行日誌程式的系統上配置一個作業系統使用者，以及在資料庫伺服器上配置一個具有資料庫遠端存取權的作業系統使用者。日誌程式會使用「連結」模式連接至 IBM MQ，存取資料庫時則使用用戶端連線。

Java EE 資料庫日誌程式位於一個系統上，IBM MQ 在另一個系統上，資料庫又在另一個系統上

您可以為此配置建立三個使用者：一個作業系統使用者負責啟動應用程式伺服器、一個 IBM MQ 使用者負責存取要使用的佇列及主題，一個資料庫伺服器使用者負責存取及插入至資料庫表格。日誌程式會使用「用戶端」模式存取 IBM MQ，並使用用戶端連線來存取資料庫。

為方便解說，本文的其餘指示均假設使用者的名稱為 `ftelog`，但您可以使用任何使用者名稱，新建或現有的都可以。請如下所示配置使用者的權限：

程序

1. 確定作業系統使用者具有專屬的群組，且不屬於任何對協調佇列管理程式具備多項權限的群組。使用者不得屬於 `mqm` 群組。在特定平台上，也會對工作人員群組自動指定佇列管理程式存取權；因此日誌程式使用者不應屬於工作人員群組。您可以使用「IBM MQ Explorer」，檢視佇列管理程式本身及其內含物件的權限記錄。用滑鼠右鍵按一下物件，然後選取物件權限 > 管理權限記錄。在指令行上，您可以使用 `dspmqaout`（顯示權限）或 `dmpmqaut`（傾出權限）指令。
2. 使用 IBM MQ Explorer 中的「管理權限記錄」視窗或 `setmqaut`（授與或撤銷權限）指令來新增 IBM MQ 使用者專屬群組的權限（在 UNIX 上，IBM MQ 權限僅與群組相關聯，而與個別使用者無關）。必要權限如下所示：
 - 對佇列管理程式的 `CONNECT` 及 `INQUIRE`（IBM MQ Java 程式庫需要 `INQUIRE` 權限才能運作）。
 - 對 `SYSTEM.FTE` 主題的 `SUBSCRIBE` 權限。
 - 對 `SYSTEM.FTE.LOG.RJCT.logger_name` 佇列的 `PUT` 權限。

指定的拒絕及指令佇列名稱為預設名稱。如果您在配置日誌程式佇列時選擇了不同的佇列名稱，請改為將權限新增至那些佇列名稱。

3. 執行所使用的資料庫專用的資料庫使用者配置。
 - 如果您的資料庫是 Db2，請執行下列步驟：

註：使用 Db2 時可透過數種機制來管理資料庫使用者。下列指示適用於以作業系統使用者為基礎的預設機制。

 - 確保 `ftelog` 使用者不屬於任何 Db2 管理群組（例如 `db2iadm1`、`db2fadm1` 或 `dasadm1`）。
 - 對使用者指定適當的權限，使其能夠連接至資料庫，以及對於在步驟 2：建立必要的資料庫表格中建立的表格，進行選取、插入及更新作業
 - 如果您的資料庫是 Oracle，請執行下列步驟：
 - 確保 `ftelog` 使用者不在任何 Oracle 管理群組中（例如，Windows 上的 `ora_dba` 或 UNIX 上的 `dba`）
 - 授與使用者連接至資料庫權限，以及對步驟 2：建立所需資料庫表格中所建立表格的選取、插入及更新權限

移轉 Java EE 資料庫日誌程式

若要將 WebSphere Application Server 7.0 上的 Java EE 資料庫日誌程式從 IBM WebSphere MQ File Transfer Edition 7.0 移轉至 IBM WebSphere MQ 7.5 或更新版本，請完成下列步驟：

程序

1. 開啟 WebSphere Application Server 主控台。
2. 按一下 **應用程式 > 應用程式類型 > 企業應用程式**。從應用程式清單中找出 IBM WebSphere MQ File Transfer Edition 資料庫日誌程式應用程式。如果資料庫日誌程式應用程式尚未停止，請選取該應用程式，然後按一下 **停止**。
3. 記下您先前對 JEE 資料庫日誌程式設定的配置設定。稍後在步驟 [第 117 頁的『7』](#) 中您會需要這些設定。
 - a) 若您最初在安裝資料庫日誌程式時，變更了 EJB 模組的預設值（如需相關資訊，請參閱步驟 9），請按一下 **企業應用程式 > WebSphere MQ File Transfer Edition 資料庫日誌程式 > EJB 模組的環境項目**，並記下該窗格中的設定。
 - b) 按一下 **企業應用程式 > WebSphere MQ File Transfer Edition 資料庫日誌程式 > 訊息驅動 Bean 接聽器連結**，並記下所使用的啟動規格，即目標資源 JNDI 名稱和目的地 JNDI 名稱。
 - c) 按一下 **企業應用程式 > WebSphere MQ File Transfer Edition 資料庫日誌程式 > 資源參照**，並記下拒絕 Queue Connection Factory 詳細資料。
 - d) 按一下 **企業應用程式 > WebSphere MQ File Transfer Edition 資料庫日誌程式 > 資源環境項目參照**，並記下拒絕佇列詳細資料。
4. 按一下 **應用程式 > 應用程式類型 > 企業應用程式**，將 IBM WebSphere MQ File Transfer Edition 資料庫日誌程式應用程式解除安裝。選取資料庫日誌程式應用程式，並按一下 **解除安裝**。
5. 選用項目：如果是使用多重安裝以移轉至 IBM WebSphere MQ 7.5 或更新版本，且原生程式庫路徑不相同，請按一下 **資源 > JMS 提供者 > WebSphere MQ 傳訊提供者** 以變更路徑。

例如，如果原生程式庫路徑為 C:\Program Files\IBM\WebSphere MQ\java\lib，請將路徑變更為 C:\Program Files\IBM\New MQ Installation Location\java\lib
6. 選用項目：如果是使用多重安裝以移轉至 IBM WebSphere MQ 7.5 或更新版本，則必須使用 `setmqm` 指令將佇列管理程式與新安裝相關聯。
7. 使用 [使用 WebSphere Application Server 7.0 安裝 IBM MQ Managed File Transfer Java EE 資料庫日誌程式](#) 中的資訊，以及您先前在步驟 [第 117 頁的『3』](#) 中記錄的資訊，重新安裝資料庫日誌程式應用程式。
8. 按一下 **應用程式 > 應用程式類型 > 企業應用程式**，來啟動新的資料庫日誌程式。選取資料庫日誌程式應用程式，並按一下 **啟動**。
9. 如果要驗證移轉，請檢查資料庫，以確定寫入項目。

從獨立式資料庫日誌程式移轉至 MFT 的 Java EE 資料庫日誌程式

您可以從獨立式資料庫日誌程式移轉至 Java EE 資料庫日誌程式。必須停止獨立式資料庫日誌程式，然後再安裝 Java EE 資料庫日誌程式。若要避免日誌項目遺失或重複，您必須在停止獨立式資料庫日誌程式之前，阻止訊息被發佈至 SYSTEM.FTE 主題，然後在 JEE 資料庫日誌程式安裝完成之後再加以重新啟動。移轉之前，請先備份資料庫。

關於這項作業


程序

1. 在停止資料庫之前，請先針對協調佇列管理程式執行以下 MQSC 指令：ALTER QM PSMODE(COMPAT) 這會停止將訊息發佈至 SYSTEM.FTE/Log 主題。靜待日誌程式處理其訂閱的所有訊息。依預設，此訂閱稱為 SYSTEM.FTE.LOGGER.AUTO。
2. 使用 `fteStopLogger` 指令停止資料庫日誌程式。
3. 使用資料庫軟體隨附的工具備份資料庫。

4. 刪除屬於獨立式資料庫日誌程式的訂閱。

依預設，此訂閱稱為 SYSTEM.FTE.LOGGER.AUTO。

5. 如果您的資料庫綱目為舊版，則必須按次序將綱目移轉至每個後續層次。例如，如果資料庫綱目位於 7.0.1 且您要移轉至 7.0.4，則必須將綱目從 7.0.1 移轉至 7.0.2，然後從 7.0.2 移轉至 7.0.3，然後從 7.0.3 移轉至 7.0.4。透過針對必須移轉的每個資料庫綱目版本執行下列其中一項動作，可以將該綱目從 *old* 版本移轉至 *new* 版本（其中 *old* 及 *new* 為說明綱目版本的變數）：

-  如果資料庫是 z/OS 上的 Db2，且您要在 7.0.2 與 7.0.3 綱目之間或在 7.0.3 與 7.0.4 綱目之間移轉，則必須建立新的資料庫綱目並將現有資料複製到其中。如需相關資訊，請參閱 [第 19 頁的『將 Db2 on z/OS 上的資料庫表格移轉至 IBM MQ 8.0』](#)。
 - 如果資料庫不是 Db2，或建立資料庫時的頁面大小超過 8K，您可以透過完成下列步驟，採用與其他版本相同的方式移轉綱目。
 - 在任何其他情況下，如果要在資料庫表格之間進行移轉，請完成下列步驟：
 - a. 選擇適用於資料庫平台並且名稱中包含字串 *old-new* 的檔案。此檔案位於 Remote Tools and Documentation 安裝的 `MQ_INSTALLATION_PATH/mqft/sql` 目錄中。
 - b. 如果已對起始綱目進行修改，請檢閱移轉檔案，以確定該檔案與修改後的資料庫相容。
 - c. 針對您的資料庫執行 SQL 檔案。
6. 安裝 Java EE 資料庫日誌程式 EAR 檔。
7. 部署 Java EE 資料庫日誌程式。如需相關資訊，請參閱 [第 105 頁的『安裝 MFT 的 Java EE 資料庫日誌程式』](#)。
8. 針對協調佇列管理程式執行以下 MQSC 指令：ALTER QMGR PSMODE(ENABLED)
這會將訊息發佈至 SYSTEM.FTE/Log 主題。

結果

配置 Connect:Direct 橋接器

配置 Connect:Direct 橋接器，以在 Managed File Transfer 網路與 Connect:Direct 網路之間傳送檔案。Connect:Direct 橋接器的元件為 Connect:Direct 節點，以及專門用來與該節點通訊的 Managed File Transfer 代理程式。此代理程式稱為 Connect:Direct 橋接器代理程式。

關於這項作業

完成下列步驟以配置 Connect:Direct 橋接器：

程序

1. [第 118 頁的『為 Connect:Direct 橋接器代理程式及節點選擇作業系統』](#)。
2. [第 119 頁的『選擇及配置 Connect:Direct 節點』](#)。
3. [第 119 頁的『建立及配置 Connect:Direct 橋接器代理程式』](#)。
4. [第 120 頁的『配置 ConnectDirectNodeProperties.xml 檔案以包含遠端 Connect:Direct 節點的相關資訊』](#)。
5. [第 120 頁的『在 Connect:Direct 橋接器代理程式與 Connect:Direct 節點之間配置安全連線』](#)。

為 Connect:Direct 橋接器代理程式及節點選擇作業系統

開始之前

組成 Connect:Direct 橋接器的代理程式及節點必須在同一系統上，或可以存取同一檔案系統（例如透過共用 NFS 裝載）。此檔案系統用於在涉及 Connect:Direct 橋接器的檔案傳送期間，將檔案暫時儲存在 `cdTmpDir` 參數所定義的目錄中。Connect:Direct 橋接器代理程式及 Connect:Direct 橋接器節點，必須能使用相同路徑名稱，對此目錄進行定址。例如，如果代理程式和節點是在不同的 Windows 系統上，則這些系統必須使用相同的磁碟機代號來裝載共用檔案系統。下列配置容許代理程式及節點使用相同路徑名稱：

- 代理程式和節點是在同一個系統上，它可能執行 Windows 或 Linux for System x
- 代理程式是在 Linux for System x 上，而節點是在 UNIX 上
- 代理程式是在一個 Windows 系統上，而節點是在另一個 Windows 系統上

下列配置不容許代理程式及節點使用相同路徑名稱：

- 代理程式是在 Linux for System x 上，而節點是在 Windows 上
- 代理程式是在 Windows 上，而節點是在 UNIX 上

規劃 Connect:Direct 橋接器安裝時，請考量此限制。

如需 Connect:Direct 橋接器支援的作業系統版本的相關詳細資料，請參閱 [IBM MQ 系統需求網頁](#)。

程序

1. 選擇執行 Windows 或 Linux on System x 的系統，以在其中安裝 Connect:Direct 橋接器代理程式。
2. 選擇 Connect:Direct for Windows 或 Connect:Direct for UNIX 支援的作業系統，以在其中安裝 Connect:Direct 橋接器節點。

選擇及配置 Connect:Direct 節點

開始之前

在遵循下列指示之前，您必須已安裝 Connect:Direct 節點。

程序

1. 選擇要與 Managed File Transfer 代理程式通訊的 Connect:Direct 節點。
2. 檢查所選 Connect:Direct 節點的網路對映。如果該網路對映包含在 Windows 作業系統上執行的遠端節點的任何項目，您必須確保這些項目指定節點在 Windows 上執行。
 - a) 如果您為 Connect:Direct 橋接器選取的 Connect:Direct 節點是在 Windows 上執行，請使用「Connect:Direct 要求端」來編輯網路對映。對於在 Windows 上執行的任何遠端節點，請確定作業系統欄位均設定為 **Windows**。

建立及配置 Connect:Direct 橋接器代理程式

關於這項作業

Connect:Direct 橋接器代理程式是一種 Managed File Transfer 代理程式，專門用來與 Connect:Direct 節點通訊。

程序

1. 使用 **fteCreateCDAgent** 指令建立 Connect:Direct 橋接器代理程式。
 - a) 您必須為 **cdNode** 參數提供值。此參數會指定代理程式用於 Connect:Direct 節點的名稱，而該節點是 Connect:Direct 橋接器的一部分。請使用您在上一節中選擇的 Connect:Direct 節點名稱。
 - b) 為 **cdNodeHost** 及 **cdNodePort** 參數提供值，這兩個參數可定義與代理程式通訊的 Connect:Direct 節點。
如果您不為 **cdNodeHost** 參數提供值，則會使用本端系統的主機名稱或 IP 位址。如果您不為 **cdNodePort** 參數提供值，則會使用值 1363。
 - c) 使用第 339 頁的『[fteCreateCDAgent \(建立 Connect:Direct 橋接器代理程式\)](#)』中的資訊來決定是否需要指定 **cdTmpDir** 參數的值。
2. 將 Managed File Transfer 使用的使用者認證，對映至 Connect:Direct 節點上的使用者認證。您可以使用下列其中一種方法來對映認證：
 - 建立用來定義認證對映資訊的 `ConnectDirectCredentials.xml` 檔案。如需相關資訊，請參閱第 120 頁的『[使用 ConnectDirectCredentials.xml 檔案對映 Connect:Direct 的認證](#)』。

- 撰寫使用者結束程式，以執行 Connect:Direct 橋接器的認證對映。如需相關資訊，請參閱 [第 123 頁的『使用結束程式類別對映 Connect:Direct 的認證』](#)。


配置 ConnectDirectNodeProperties.xml 檔案以包含遠端 Connect:Direct 節點的相關資訊

開始之前

在遵循下列指示之前，您必須已建立 Connect:Direct 橋接器代理程式。

程序

編輯 Connect:Direct 橋接器代理程式配置目錄中的 ConnectDirectNodeProperties.xml 範本。針對您要定義其相關資訊的每個 Connect:Direct 節點或節點群組，執行下列步驟：

- a) 在 nodeProperties 元素內，建立 node 元素。
- b) 將 name 屬性新增至 node 元素。請將此屬性的值，指定為符合一個以上遠端 Connect:Direct 節點名稱的型樣。
- c) 選擇性的：將 pattern 屬性新增至 node 元素，以指定 name 屬性值中的型樣類型。有效值為 regex 及 wildcard。預設選項為 wildcard。
- d) 將 type 屬性新增至 node 元素，以指定執行 name 屬性所指定遠端 Connect:Direct 節點的作業系統。
下列為有效值：
 - Windows - 節點在 Windows 上執行
 - UNIX - 節點在 UNIX 或 Linux 上執行
 -  z/OS、zos、os/390 或 os390 - 節點在 z/OS 上執行

此屬性的值不區分大小寫。Connect:Direct 橋接器不支援傳送至其他作業系統上的遠端節點。

如需相關資訊，請參閱 [第 494 頁的『Connect:Direct 節點內容檔格式』](#)。

在 Connect:Direct 橋接器代理程式與 Connect:Direct 節點之間配置安全連線

關於這項作業

依預設，Connect:Direct 橋接器代理程式會使用 TCP/IP 通訊協定連接至 Connect:Direct 節點。如果要在 Connect:Direct 橋接器代理程式與 Connect:Direct 節點之間建立安全連線，則可使用 SSL 通訊協定或 TLS 通訊協定。

程序

配置安全連線。如需如何執行操作的範例，請參閱 [第 46 頁的『在 Connect:Direct 橋接器代理程式與 Connect:Direct 節點之間配置 SSL 或 TLS』](#)。

對映 Connect:Direct 的認證

使用 Connect:Direct 橋接器代理程式的預設認證對映功能，或撰寫專屬的使用者結束程式，即可將 Managed File Transfer 中的使用者認證，對映至 Connect:Direct 節點上的使用者認證。Managed File Transfer 提供一個使用者結束程式範例，可執行使用者認證對映。

使用 ConnectDirectCredentials.xml 檔案對映 Connect:Direct 的認證

使用 Connect:Direct 橋接器代理程式的預設認證對映功能，將 Managed File Transfer 中的使用者認證對映至 Connect:Direct 節點上的使用者認證。Managed File Transfer 提供 XML 檔案，您可以編輯此檔案以包括自己的認證資訊。

關於這項作業

使用 **fteCreateCDAgent** 指令建立 Connect:Direct 橋接器代理程式之後，需要手動建立 ConnectDirectCredentials.xml 檔案。您必須先編輯此檔案以併入主機、使用者及認證資訊，才能使用 Connect:Direct 橋接器代理程式。如需相關資訊，請參閱第 491 頁的『Connect:Direct 認證檔格式』。依預設，此檔案會從現行使用者的起始目錄載入，例如 /home/fteuser/ConnectDirectCredentials.xml。若要使用其他位置，請使用 ConnectDirectNodeProperties.xml 檔案中的 <credentialsFile> 元素指定該位置。

程序

1. 確定 <tns:pnode name="Connect:Direct node host" pattern="wildcard"> 元素中的 name 屬性，包含 Connect:Direct 橋接器代理程式所連接的 Connect:Direct 節點名稱的值。此值必須是您為 **fteCreateCDAgent -cdNode** 參數指定的同一值。
pattern 屬性的值可以是 wildcard 或 regex。如果未指定此屬性，預設值將為 wildcard。
2. 將使用者 ID 及認證資訊，作為 <tns:pnode> 的子元素插入檔案中。
您可以將下列 <tns:user> 元素的一個以上實例插入檔案中：

```
<tns:user name="name"
          pattern="pattern"
          ignorecase="ignorecase"
          cdUserId="cdUserId"
          cdPassword="cdPassword"
          pnodeUserId="pnodeUserId"
          pnodePassword="pnodePassword">
</tns:user>
```

其中：

- *name* 是符合與 MFT 傳送要求相關聯的 MQMD 使用者 ID 的型樣。
- *pattern* 可指定為 name 屬性所指定的型樣，是萬用字元表示式還是 Java 正規表示式。pattern 屬性的值可以是 wildcard 或 regex。如果未指定此屬性，預設值將為 wildcard。
- *ignorecase* 可指定，是否區分 name 屬性所指定型樣的大小寫。如果未指定此屬性，預設值將為 true。
- *cdUserId* 為使用者 ID，可供 Connect:Direct 橋接器代理程式，用來連接至 <tns:pnode> 元素之 name 屬性所指定的 Connect:Direct 節點。如有可能，請確保 *cdUserId* 為 Connect:Direct 管理者使用者 ID。如果 *cdUserId* 不得為 Connect:Direct 管理者，請確保該使用者 ID 在 Connect:Direct 橋接器節點上具有下列職能權限：
 - 若為 Windows 節點，請設定下列權限。為了提高可讀性，以下範例使用換行進行格式化：

```
View Processes in the TCQ      value: yes
Issue the copy receive, copy send, run job, and run task Process statements
Issue the submit Process statement      value: yes
Monitor, submit, change, and delete all Processes      value: all
Access Process statistics      value: all
Use the trace tool or issue traceon and traceoff commands      value: yes
Override Process options such as file attributes and remote node ID      value: yes
```

- 若為 UNIX 節點，請在 `userfile.cfg` 檔案中設定下列參數：

```
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` 是與 `cdUserId` 屬性所指定使用者 ID 相關聯的密碼。
- 您可以選擇性地指定 `pnodeUserId` 屬性。此屬性的值為使用者 ID，可供 `<tns:pnode>` 元素之 `name` 屬性所指定的 `Connect:Direct` 節點，用來提交 `Connect:Direct` 程序。如果您沒有指定 `pnodeUserId` 屬性，`Connect:Direct` 節點會使用 `cdUserId` 屬性所指定的使用者 ID，來提交 `Connect:Direct` 程序。
- 您可以選擇性地指定 `pnodePassword` 屬性。此屬性的值是與 `pnodeUserId` 屬性所指定使用者 ID 相關聯的密碼。

如果沒有任何使用者元素符合 MQMD 使用者 ID，傳送將會失敗。

3. 選擇性的: 您可以併入一個以上的 `<tns:snode>` 元素，作為 `<tns:user>` 元素的子元素。`<tns:snode>` 元素指定屬於 `Connect:Direct` 橋接器的 `Connect:Direct` 節點所使用的認證。這些認證皆為使用者 ID 及密碼，可供 `Connect:Direct` 橋接器節點，用來連接至作為檔案傳送來源或目的地的 `Connect:Direct` 節點。

請將下列一個或多個元素插入檔案中：

```
<tns:snode name="name"
  pattern="pattern"
  userId="userId"
  password="password" />
```

其中：

- `name` 是符合作為檔案傳送來源或目的地的 `Connect:Direct` 節點名稱的型樣。
- `pattern` 可指定為 `name` 屬性所指定的型樣，是萬用字元表示式還是 Java 正規表示式。`pattern` 屬性的值可以是 `wildcard` 或 `regex`。如果未指定此屬性，預設值將為 `wildcard`。
- `userId` 為使用者 ID，可供 `<tns:pnode>` 元素之 `name` 屬性所指定的 `Connect:Direct` 節點，用來連接至符合 `<tns:snode>` 之 `name` 屬性所指定型樣的 `Connect:Direct` 節點。
- `password` 是與 `userId` 屬性所指定使用者 ID 相關聯的密碼。

如果沒有任何 `<tns:snode>` 元素符合檔案傳送的次要節點，這並不會導致傳送失敗。傳送仍會啟動，但不會指定與 `snode` 搭配使用的使用者 ID 及密碼。

結果

在搜尋使用者名稱或 `Connect:Direct` 節點名稱的型樣相符項時，`Connect:Direct` 橋接器代理程式會從檔案開頭開始搜尋到檔案結尾。找到的第一個相符項，即為所用的相符項。

相關工作

第 118 頁的『[配置 Connect:Direct 橋接器](#)』

配置 `Connect:Direct` 橋接器，以在 `Managed File Transfer` 網路與 `Connect:Direct` 網路之間傳送檔案。

`Connect:Direct` 橋接器的元件為 `Connect:Direct` 節點，以及專門用來與該節點通訊的 `Managed File Transfer` 代理程式。此代理程式稱為 `Connect:Direct` 橋接器代理程式。

相關參考

第 491 頁的『[Connect:Direct 認證檔格式](#)』

Managed File Transfer Agent 配置目錄中的 `ConnectDirectCredentials.xml` 檔案定義 `Connect:Direct` 代理程式用來向 `Connect:Direct` 節點進行自我授權的使用者名稱及認證資訊。

第 339 頁的『[fteCreateCDAgent \(建立 Connect:Direct 橋接器代理程式\)](#)』

`fteCreateCDAgent` 指令可建立 Managed File Transfer Agent 及其與 `Connect:Direct` 橋接器搭配使用的關聯配置。

使用結束程式類別對映 `Connect:Direct` 的認證

如果您不想要使用 `Connect:Direct` 橋接器代理程式的預設認證對映功能，您可以撰寫專屬的使用者結束程式，將 Managed File Transfer 中的使用者認證對映至 `Connect:Direct` 節點上的使用者認證。配置專屬的認證對映使用者結束程式，將會停用預設認證對映功能。

關於這項作業

為對映 `Connect:Direct` 認證而建立的使用者結束程式，必須實作 `com.ibm.wmqfte.exitroutine.api.ConnectDirectCredentialExit` 介面。如需相關資訊，請參閱第 761 頁的『[CDCredentialExit.java 介面](#)』。

在 `Connect:Direct` 橋接器代理程式與 `Connect:Direct` 節點之間配置 SSL 或 TLS

透過建立金鑰儲存庫與信任儲存庫，以及設定 `Connect:Direct` 橋接器代理程式內容檔中的內容，即可將 `Connect:Direct` 橋接器代理程式與 `Connect:Direct` 節點，配置為透過 SSL 通訊協定彼此連接。

關於這項作業

下列步驟包含一些指示來取得透過憑證管理中心簽署的金鑰。如果您不使用憑證管理中心，則可產生自簽憑證。如需產生自簽憑證的相關資訊，請參閱在 [UNIX 和 Windows 系統上使用 SSL 或 TLS](#)。

下列步驟包含為 `Connect:Direct` 橋接器代理程式，建立新的金鑰儲存庫及信任儲存庫的指示。如果 `Connect:Direct` 橋接器代理程式，已經具有用於安全連接至 IBM MQ 佇列管理程式的金鑰儲存庫及信任儲存庫，則可在安全連接至 `Connect:Direct` 節點時使用現有的金鑰儲存庫及信任儲存庫。如需相關資訊，請參閱第 44 頁的『[配置 MFT 的 SSL 或 TLS 加密](#)』。

程序

對於 `Connect:Direct` 節點，請完成下列步驟：

1. 產生用於 `Connect:Direct` 節點的金鑰及已簽章的憑證。
您可以使用 IBM MQ 隨附的 IBM Key Management 工具來執行此動作。如需相關資訊，請參閱[使用 SSL 或 TLS](#)。
2. 向憑證管理中心傳送簽署金鑰的要求。您會收到回報的憑證。
3. 建立包含憑證管理中心公開金鑰的文字檔，例如 `/test/ssl/certs/CAcert`。
4. 在 `Connect:Direct` 節點上安裝 Secure+ Option。
如果該節點已經存在，您可以透過下列方式來安裝 Secure+ Option：再次執行安裝程式，指定現有安裝的位置，然後選擇僅安裝 Secure+ Option。
5. 建立新的文字檔，例如 `/test/ssl/cd/keyCertFile/node_name.txt`。
6. 將您從憑證管理中心收到的憑證，以及位於 `/test/ssl/cd/privateKeys/node_name.key` 中的私密金鑰，複製到該文字檔中。

`/test/ssl/cd/keyCertFile/node_name.txt` 的內容必須採用下列格式：

```
-----BEGIN CERTIFICATE-----
MIIcCnzCCAgigAwIBAgIBGjANBgkqhkiG9w0BAQUFADBeMQswCQYDVQQGEWJHqjES
MBAGA1UECBMJSgFtcHNoaXJlMRAwDgYDVQQHEWdIdXJzbGV5M0wwCgYDVQQKEWVj
Qk0xDjAMBGNVBAStBU1RSVBUMQswCQYDVQQDEWJQDTAeFw0xMTAzMDEwNjIwNDZa
Fw0yMTAyMjYxNjIwNDZaMFAXCzAjbG9NVBAyTAKdCMRIwEAYDVQQIEWlIYW1wc2hp
cmUxDDAKBGNVBAoTA01CTTEOMAwGA1UECXMFTVFGVEUxDzANBgNVBAMTBmJpbmJh
ZzCBnzANBgkqhkiG9w0BAQEFAA0BjQAwgYkCgYEAvgP1QIk1U9ypSKD1XoOdo1yk
EyMFXB0UpzRrDVxjoSEC0vtWncJ199e+Vc4UpNybdyBu+Nkd1MNoF4QxeQcLAFj
WnhakqCiQ+JIAD5Aurhn1wChe0MV3kjA84GKH/10SVqt1984mu/1DyS819XcfSSn
c00MsK1KbneVSCIV2XECaWAAaA7MHkwCQYDVR0TBAlwADAAsBg1ghkgBhvhCAQ0E
```

```

HxYdT3Blb1NTTCBHZW51cmF0ZWQgQ2VydG1maWNhdGUwHQYDVR00BBYEFNXMIpSc
csBXUniW4A3UirZnCRsv3MB8GA1UdIwQYMBaAFDXY8rmj41Vz5+FVAoQb++cns+B4
MA0GCSqGSIB3DQEBBQUAA4GBAFc7k1Xa4pGKYgwchxKpE3ZF6FNwy4vBXS216/ja
8h/v18+iv010CL8t0ZOKSU95fyZLzOPKnCH7v+ItFSE3CIIEk9D1z2U6W09LICwn
17PL72Tdfal3kabwHYVf17IVcuL+VZsZ3HjLggP2qH09ZuJPspeT9+AxFVMLiaAb
8eHw
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4, ENCRYPTED
DEK-Info: DES-EDE3-CBC,64A02DA15B6B6EF9

57kqxL0J/gRU0IQ6hVK2YN13B4E1jAi1gSme0I5ZpEIG8CHXISKB7/0cke2FTqsV
lvI99QyCxsDwMnt5fj51v7aPmVeS60b0m+U1Gre8B/Ze18JVj204K2Uh72rDCXE
5e6eFxDmU207sQDy20euBVELJtM2k0kL1R0doQQS1U3XQNgJw/t3ZIx5hPXWEQT
rjRQ064BEhb+PzzxPF8uwzZ9IrkUK9BJ/UUnqC60dBR87IeA4pnJD1Jvb2ML7EN9Z
5Y+50hTKI80GvBvWX04fHyvIX5as1whBoArXIS1AtNTprtPvoaP1zyIAeZ6OCVo/
SFO+A2UhmteEJe0JaZG2XZ3H495fAw/EHmjehzIACwukQ9nSIETgu4A1+CV64RJED
aYBCM8UjaAkbZDH5gn7+eBov0ssXAXWdyJBVhU0jXjvAj/e1h+kcSF1hax5D//AI
66nRMZzboSxNqkjcVd8wfdwP+bejDzUaaarJTS7lIFeLlw7eJ8MNAkMGicDkycL0
EPBU9X5QnHKLK0fYHN/1WgUk8qt3UytFXXfzTXGF3EbsWbBupkT5e5+1YcX80VZ6
sHFPN1HluCNy/riUcBy9iviVeodX8Iom0chSy05DK18bwZNjYtUP+CtYHNFU5BaD
I+1uU0AeJ+wjQYKT1WaeIGZ3VxuNITJu18y5qDTXXfX7vxM50oWXa6U5+AYuGUMg
/itPZmUmNzHjTk7ghT6i1IQ0aBowXXKJB1Mmq/6BQXN2Ihkd9ys2qrvM1hdi5nAf
egmdiG50loLnBRqWbfr+DykpAhK4SaDi2F52Uxovw3Lhwi8dQP71zQ==
-----END RSA PRIVATE KEY-----

```

7. 啟動 Secure+ Admin Tool。

- 在 Linux 或 UNIX 系統上，執行 **spadmin.sh** 指令。
- 在 Windows 系統上，按一下開始 > 程式集 > **Sterling Commerce Connect:Direct > CD Secure+ Admin Tool**

CD Secure+ Admin Tool 即會啟動。

8. 在 CD Secure+ Admin Tool 中，按兩下 **.Local** 行以編輯主要 SSL 或 TLS 設定。

- 視您使用的通訊協定而定，選取**啟用 SSL 通訊協定**或**啟用 TLS 通訊協定**。
- 選取**停用置換**。
- 至少選取一個密碼組合。
- 如果您想要雙向鑑別，請將 **啟用用戶端鑑別** 的值變更為 Yes。
- 在**授信主要憑證**欄位中，輸入憑證管理中心的公開憑證檔路徑 (/test/ssl/certs/CAcert)。
- 在**金鑰憑證檔**欄位中，輸入您所建立的檔案路徑 (/test/ssl/cd/keyCertFile/node_name.txt)。

9. 按兩下 **.Client** 行以編輯主要 SSL 或 TLS 設定。

- 視您使用的通訊協定而定，選取**啟用 SSL 通訊協定**或**啟用 TLS 通訊協定**。
- 選取**停用置換**。

對於 Connect:Direct 橋接器代理程式，請執行下列步驟：

10. 建立信任儲存庫。您可以透過先建立後刪除虛擬金鑰來執行此作業。

您可以使用下列指令：

```
keytool -genkey -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
```

```
keytool -delete -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
```

11. 將憑證管理中心的公開憑證匯入信任儲存庫。

您可以使用下列指令：

```
keytool -import -trustcacerts -alias myCA
-file /test/ssl/certs/CAcert
-keystore /test/ssl/fte/stores/truststore.jks
```

12. 編輯 Connect:Direct 橋接器代理程式內容檔。

將下列行併入檔案中的任何位置：

```
cdNodeProtocol=protocol
cdNodeTruststore=/test/ssl/fte/stores/truststore.jks
cdNodeTruststorePassword=password
```

在此步驟的範例中，*protocol* 是您使用的通訊協定（SSL 或 TLS），*password* 是您在建立信任儲存庫時指定的密碼。

13. 如果要進行雙向鑑別，請為 Connect:Direct 橋接器代理程式建立金鑰及憑證。

a) 建立金鑰儲存庫及金鑰。

您可以使用下列指令：

```
keytool -genkey -keyalg RSA -alias agent_name
        -keystore /test/ssl/fte/stores/keystore.jks
        -storepass password -validity 365
```

b) 產生簽署要求。

您可以使用下列指令：

```
keytool -certreq -v -alias agent_name
        -keystore /test/ssl/fte/stores/keystore.jks -storepass password
        -file /test/ssl/fte/requests/agent_name.request
```

c) 將您從之前步驟中收到的憑證匯入金鑰儲存庫。此憑證必須採用 x.509 格式。

您可以使用下列指令：

```
keytool -import -keystore /test/ssl/fte/stores/keystore.jks
        -storepass password -file certificate_file_path
```

d) 編輯 Connect:Direct 橋接器代理程式內容檔。

將下列行併入檔案中的任何位置：

```
cdNodeKeystore=/test/ssl/fte/stores/keystore.jks
cdNodeKeystorePassword=password
```

在此步驟的範例中，*password* 是您在建立金鑰儲存庫時指定的密碼。

相關工作

[第 118 頁的『配置 Connect:Direct 橋接器』](#)

配置 Connect:Direct 橋接器，以在 Managed File Transfer 網路與 Connect:Direct 網路之間傳送檔案。

Connect:Direct 橋接器的元件為 Connect:Direct 節點，以及專門用來與該節點通訊的 Managed File Transfer 代理程式。此代理程式稱為 Connect:Direct 橋接器代理程式。

使用 ConnectDirectProcessDefinition.xml 檔案指定要啟動的 Connect:Direct 程序

指定要在 Managed File Transfer 傳送過程中啟動的 Connect:Direct 程序。Managed File Transfer 提供 XML 檔案，您可以編輯此檔案以指定程序定義。

關於這項作業

fteCreateCDAgent 指令在代理程式配置目錄 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name` 中建立 `ConnectDirectProcessDefinitions.xml` 檔案。您必須先編輯此檔案以設定程序定義，然後才能從 Connect:Direct 橋接器代理程式，呼叫使用者定義的 Connect:Direct 程序。

對於您要指定在透過 Connect:Direct 橋接器進行傳送過程中呼叫的每個程序，請執行下列步驟：

程序

1. 定義您要讓 Connect:Direct 橋接器代理程式，在傳送過程中呼叫的 Connect:Direct 程序，然後將程序範本儲存在檔案中。
2. 在文字編輯器中開啟 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name/ConnectDirectProcessDefinitions.xml` 檔案。
3. 建立 `<processSet>` 元素。
4. 在 `<processSet>` 元素內，建立 `<condition>` 元素。
5. 在 `<condition>` 元素內，建立一或多個元素來定義傳送要求必須符合的條件，以呼叫您在步驟 1 中定義的 Connect:Direct 程序。這些元素可以是 `<match>` 元素或 `<defined>` 元素。
 - 使用 `<match>` 元素可指定變數的值必須符合某個型樣。請建立具有下列屬性的 `<match>` 元素：
 - `variable` - 要比較其值的變數名稱。此變數為本質符號。如需相關資訊，請參閱 [第 554 頁的『與使用者定義的 Connect:Direct 程序搭配使用的替代變數』](#)。
 - `value` - 要與指定變數值進行比較的型樣。
 - 選用項目: `pattern` - `value` 屬性值所用的型樣類型。此型樣類型可以是 `wildcard` 或 `regex`。此為選用屬性，預設值為 `wildcard`。
 - 使用 `<defined>` 元素可指定必須為變數定義值。請建立具有下列屬性的 `<defined>` 元素：
 - `variable` - 必須為其定義值的變數名稱。此變數為本質符號。如需相關資訊，請參閱 [第 554 頁的『與使用者定義的 Connect:Direct 程序搭配使用的替代變數』](#)。

在 `<condition>` 元素內指定的條件，可使用邏輯 AND 結合在一起。必須符合所有條件，Connect:Direct 橋接器代理程式才能呼叫此 `<processSet>` 元素指定的程序。如果您沒有指定 `<condition>` 元素，此程序集將符合所有傳送。

6. 在 `<processSet>` 元素內，建立 `<process>` 元素。
7. 在 `<process>` 元素內，建立 `<transfer>` 元素。

`transfer` 元素可指定 Connect:Direct 橋接器代理程式，在傳送過程中呼叫的 Connect:Direct 程序。請建立具有下列屬性的 `<transfer>` 元素：

- `process` - 您在步驟 1 中定義之 Connect:Direct 處理程序的位置。這個檔案的位置是以絕對路徑或相對於 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name` 目錄來指定。

結果

在搜尋條件相符項時，Connect:Direct 橋接器代理程式會從檔案開頭開始搜尋到檔案結尾。找到的第一個相符項，即為所用的相符項。

相關工作

[第 118 頁的『配置 Connect:Direct 橋接器』](#)

配置 Connect:Direct 橋接器，以在 Managed File Transfer 網路與 Connect:Direct 網路之間傳送檔案。Connect:Direct 橋接器的元件為 Connect:Direct 節點，以及專門用來與該節點通訊的 Managed File Transfer 代理程式。此代理程式稱為 Connect:Direct 橋接器代理程式。

相關參考

[第 495 頁的『Connect:Direct 程序定義檔案格式』](#)

Connect:Direct 橋接器代理程式配置目錄中的 `ConnectDirectProcessDefinitions.xml` 檔案，會指定要在檔案傳送過程中啟動的使用者定義的 Connect:Direct 程序。

[第 339 頁的『fteCreateCDAgent \(建立 Connect:Direct 橋接器代理程式\)』](#)

`fteCreateCDAgent` 指令可建立 Managed File Transfer Agent 及其與 Connect:Direct 橋接器搭配使用的關聯配置。

使用 MSCS 配置 MFT 代理程式

如果平台受 MFT 支援且執行其中一個 Windows 版本，則支援 Managed File Transfer (MFT) 代理程式 Microsoft Cluster Service (MSCS) 設定。

關於這項作業

此作業說明兩個實務範例，您可以遵循這些實務範例來實現 MFT 代理程式的失效接手：

- 實務範例 1: 將代理程式配置為 MSCS 資源。
- 實務範例 2: 將代理程式佇列管理程式及代理程式配置為 MSCS 資源。

程序

實務範例 1: 將代理程式配置為 MSCS 資源

- 若要將代理程式配置為 MSCS 資源，請完成下列步驟：
 - a) 在叢集中的每個機器本端安裝 Managed File Transfer。
請參閱[安裝 Managed File Transfer](#)。
 - b) 在叢集中的主要機器上建立代理程式。
代理程式應該配置為使用 CLIENT 傳輸連接至代理程式佇列管理程式。請確定您在佇列管理程式上建立此代理程式的所有物件。如需如何執行此動作的相關資訊，請參閱[設定代理程式](#)。
 - c) 透過將 Windows 服務工具中代理程式服務的 **啟動類型** 欄位設為 **手動**，修改代理程式以作為 Windows 服務執行，並將其配置為在 Windows 重新啟動時不會自動啟動。
如需相關資訊，請參閱 [以 Windows 服務方式啟動 MFT 代理程式](#)。
 - d) 在次要機器上重複實務範例 1 的步驟 [第 127 頁的『2』](#) 及步驟 [第 127 頁的『3』](#)。
這可確保日誌、內容等的檔案結構存在於叢集中的其他機器上。請注意，不需要如步驟 [第 127 頁的『2』](#) 一樣建立佇列管理程式物件。
 - e) 在主要機器上，將代理程式新增為 MSCS 控制下的「一般服務」。
若要執行此作業：
 - a. 用滑鼠右鍵按一下叢集，然後選取 **角色-> 新增資源-> 「一般服務」**。
 - b. 從 Windows 服務清單中，選取代理程式服務，然後按 **下一步** 來完成配置精靈。代理程式服務現在已新增為 MSCS 資源。如果發生失效接手，則會在另一部機器上啟動代理程式服務。

實務範例 2: 將代理程式佇列管理程式及代理程式配置為 MSCS 資源

- 若要將代理程式佇列管理程式及代理程式配置為 MSCS 資源，請完成下列步驟：
 - a) 將代理程式佇列管理程式配置成作為 MSCS 資源執行。
如需如何執行此動作的相關資訊，請參閱 [將佇列管理程式置於 MSCS 控制下](#)。
 - b) 在叢集中的主要機器上建立代理程式。
代理程式應該配置為使用 BINDINGS 傳輸連接至代理程式佇列管理程式。請確定您在佇列管理程式上建立此代理程式的所有物件。如需如何執行此動作的相關資訊，請參閱 [設定代理程式](#)。
 - c) 透過將 Windows 服務工具中代理程式服務的 **啟動類型** 欄位設為 **手動**，修改代理程式以作為 Windows 服務執行，並將其配置為在 Windows 重新啟動時不會自動啟動。
如需相關資訊，請參閱 [以 Windows 服務方式啟動 MFT 代理程式](#)。
 - d) 請確定代理程式佇列管理程式 (在 MSCS 控制下) 在次要機器上執行。
在此機器上建立的代理程式將使用 BINDINGS 傳輸來連接至佇列管理程式，因此在建立代理程式時需要可用。
 - e) 在次要機器上重複實務 2 的步驟 [第 127 頁的『2』](#) 和步驟 [第 127 頁的『3』](#)。
這可確保日誌、內容等的檔案結構存在於叢集中的其他機器上。請注意，不需要如步驟 [第 127 頁的『2』](#) 一樣建立佇列管理程式物件。
 - f) 將代理程式新增為 MSCS 控制下的「通用服務」。
若要執行此作業：
 - a. 用滑鼠右鍵按一下叢集，然後選取 **角色-> 新增資源-> 「一般服務」**。
 - b. 從 Windows 服務清單中，選取代理程式服務，然後按 **下一步** 來完成配置精靈。
 - g) 修改代理程式服務的資源內容，以在相依關係清單中新增佇列管理程式資源。

這可確保在啟動代理程式之前啟動佇列管理程式資源。

- h) 讓佇列管理程式資源離線，然後讓代理程式資源連線。請驗證佇列管理程式資源和代理程式是否都已啟動。

如果發生失效接手，則會在次要機器上啟動代理程式服務及代理程式佇列管理程式。

管理 MFT

使用 Managed File Transfer 指令來管理 Managed File Transfer。您也可以使用「IBM MQ Explorer」來執行部分管理作業。

將訊息放置在代理程式指令佇列中以啟動傳送

您也可以將檔案傳送訊息放置在來源代理程式的指令佇列上以啟動檔案傳送。

SYSTEM.FTE.COMMAND.AGENT01 是指令佇列名稱範例。您必須確定訊息送達正確來源代理程式的指令佇列；如果代理程式收到的訊息與 XML 中的來源資訊不符，則會拒絕訊息。

The transfer request XML must conform to the FileTransfer.xsd schema and use the <request> element as the root element. 如需傳送要求訊息的結構及內容的相關資訊，請參閱檔案傳送要求訊息格式。將傳送要求訊息放置在代理程式指令佇列上的作法視作業而定。例如，您可以使用 IBM MQ Java API，以程式設計方式將訊息放置在佇列上。

啟動 MFT 代理程式

必須先啟動 Managed File Transfer 代理程式，才能將它用來進行檔案傳送。

關於這項作業

您可以從指令行啟動 Managed File Transfer Agent。在此情況下，當您登出系統時，代理程式程序即會停止。

ULW 在 UNIX, Linux, and Windows 上，您可以配置代理程式，讓它在您從系統登出時繼續執行並可繼續接收檔案傳送。

z/OS 在 z/OS 上，您可以將代理程式配置為從 JCL 以啟動作業的方式來啟動代理程式，而不需要互動式階段作業。

請注意，如果代理程式在執行時發生無法復原的錯誤，則會產生首次失敗資料擷取 (FDC) 並停止代理程式。

程序

- 若要從指令行啟動代理程式，請使用 **fteStartAgent** 指令。
如需相關資訊，請參閱 [第 445 頁的『fteStartAgent』](#)。
- ULW**
 - 若要配置代理程式以讓它在您從系統登出時繼續執行，請完成下列動作：
 - Windows** 在 Windows 上，將代理程式配置為以 Windows 服務方式執行。如需相關資訊，請參閱 [第 129 頁的『以 Windows 服務方式啟動 MFT 代理程式』](#)。
 - Linux** **UNIX** 在 UNIX 及 Linux 上，使用 Script 檔將代理程式配置為在重新啟動期間自動啟動。如需相關資訊，請參閱 [第 130 頁的『在 UNIX 及 Linux 系統啟動時啟動 MFT 代理程式』](#)。
- z/OS**
 - 在 z/OS 上，將代理程式配置為從 JCL 以啟動作業的方式來啟動代理程式，而不需要互動式階段作業。如需相關資訊，請參閱 [第 132 頁的『在 z/OS 上啟動 MFT 代理程式』](#)。

Windows 以 Windows 服務方式啟動 MFT 代理程式

您可以 Windows 服務方式啟動代理程式，則當您登出 Windows 時，代理程式會繼續執行且可以接收檔案傳送。

關於這項作業

在 Windows 上，當您從指令行啟動代理程式時，代理程式程序會使用您用來登入 Windows 的使用者名稱來執行。當您登出系統時，代理程式程序即會停止。若要防止代理程式停止，您可以配置代理程式以 Windows 服務方式執行。以 Windows 服務方式執行，也可以讓您配置在 Windows 環境啟動或重新啟動時自動啟動代理程式。

請完成下列步驟，來啟動以 Windows 服務方式執行的代理程式。您必須在其中一個支援的 Windows 版本上執行 Managed File Transfer，才能以 Windows 服務方式執行該代理程式。如需支援的環境清單，請參閱 IBM MQ 系統需求。

確切的步驟視您是否已建立代理程式或正在建立代理程式而定。下列步驟說明這兩個選項。

程序

1. 如果您要建立 Managed File Transfer 代理程式，請使用 **fteCreateAgent**、**fteCreateCDAgent** 或 **fteCreateBridgeAgent** 指令。請指定 **-s** 參數，表示要以 Windows 服務方式執行代理程式。在下列範例中，建立了代理程式 AGENT1，它具有代理程式佇列管理程式 QMGR1。Windows 服務會使用使用者名稱 fteuser 來執行，其具有相關聯的密碼 ftepassword。

```
fteCreateAgent -agentName AGENT1 -agentQMGr QMGR1 -s -su fteuser -sp ftepassword
```

您可以在 **-s** 參數之後，選擇性地指定服務的名稱。如果未指定名稱，則此服務會命名為 mqmftAgentAGENTQMGR，其中 AGENT 是您指定的代理程式名稱，而 QMGR 是代理程式佇列管理程式名稱。在此範例中，服務的預設名稱是 mqmftAgentAGENT1QMGR1。

註：您使用 **-su** 參數指定的 Windows 使用者帳戶必須具有 **Log on as a service** 權限。如需如何配置的相關資訊，請參閱第 273 頁的『以 Windows 服務方式執行 MFT 代理程式或日誌程式的指引』。

如需相關資訊，請參閱第 326 頁的『fteCreateAgent (建立 MFT 代理程式)』、第 339 頁的『fteCreateCDAgent (建立 Connect:Direct 橋接器代理程式)』或第 330 頁的『fteCreateBridgeAgent (建立及配置 MFT 通訊協定橋接器代理程式)』。

2. 如果您遵循前一個步驟來建立代理程式，請執行 **fteCreateAgent**、**fteCreateCDAgent** 或 **fteCreateBridgeAgent** 指令所產生的 MQSC 指令。這些指令會建立代理程式所需的 IBM MQ 佇列。例如，若為代理程式名為 AGENT1、代理程式佇列管理程式名為 QMGR1 及協調佇列管理程式名為 COORDQMGR1，請執行下列指令：

```
runmqsc QMGR1 MQ_DATA_PATH\mqft\config\COORDQMGR1\agents\AGENT1\AGENT1_create.mqsc
```

3. 如果您未遵循先前的步驟建立代理程式，而是想要配置現有的代理程式以 Windows 服務方式執行，請先停止執行中的代理程式，然後修改其配置。
 - a) 下列範例使用代理程式名為 AGENT1。請執行下列指令：

```
fteStopAgent AGENT1
```

- b) 使用 **fteModifyAgent** 指令，將代理程式配置成以 Windows 服務方式執行：

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

如需相關資訊，請參閱第 413 頁的『fteModify 代理程式 (以 Windows 服務方式執行 MFT 代理程式)』。

4. 使用 **fteStartAgent** 指令來啟動代理程式。或者，您也可以從 Windows 桌面開始功能表選取「控制台」中的「系統管理工具」，使用 Windows 「服務」工具來啟動該服務。

```
fteStartAgent AGENT1
```

即使您登出 Windows，該服務也會繼續執行。為了確保關機之後 Windows 重新啟動時該服務也會重新啟動，Windows「服務」工具中的**啟動類型**欄位預設會設為**自動**。如果您不想要該服務在 Windows 重新啟動時重新啟動，請將此設定變更為**手動**。

5. 選擇性的: 若要停止代理程式，請使用 `fteStopAgent` 指令，或使用 Windows「服務」工具。例如，從指令行執行下列指令：

```
fteStopAgent AGENT1
```

- 當您以服務方式執行 **fteStopAgent** 指令時，不論您是否指定 **-i** 參數，該指令一律使用此參數執行。**-i** 參數會立即停止代理程式，不會完成進行中的任何傳送。這是因為 Windows 服務的限制所致。

下一步

如果您在啟動 Windows 服務時發生問題，請參閱第 273 頁的『[以 Windows 服務方式執行 MFT 代理程式或日誌程式的指引](#)』。這個主題也說明 Windows 服務日誌檔的位置。

Linux

UNIX

在 UNIX 及 Linux 系統啟動時啟動 MFT 代理程式

在 UNIX 上，可將 Managed File Transfer Agent 配置為在系統啟動時啟動。當您登出 UNIX 系統時，代理程式會繼續執行且可以接收檔案傳送。

當您使用下列其中一個 Managed File Transfer 指令 (**fteCreateAgent**、**fteCreateCDAgent**、或 **fteCreateBridgeAgent**) 來建立及配置代理程式時，您可以使用僅執行下列指令的 Script 檔，將它配置為在 UNIX 機器上重新開機期間自動啟動：

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

其中 `mq_install_root` 是必要的 Managed File Transfer 安裝架構的根目錄，預設值為: `/opt/mqm`，而 `agent_name` 是要啟動之 Managed File Transfer Agent 的名稱。這個 Script 檔的用法視特定的 UNIX 作業系統而不同。

Solaris

Solaris

在 Solaris 上，請遵循下列步驟：

1. 執行下列指令，並追蹤傳回的路徑：

```
which sh
```

例如，此路徑可能是 `/usr/bin/sh`

2. 以系統 root 使用者 ID 建立您專屬的 `/etc/init.d/startmqmft` 檔案。
3. 編輯此檔案並對它新增 Script 指令行，並使用步驟 1 傳回的路徑作為 Script 的第一行：

```
#!/usr/bin/sh
su mqmft_user mq_install_root/bin/fteStartAgent agent_name
```

4. 使該檔案成為執行檔，例如：

```
chmod 755 /etc/init.d/startmqmft
```

5. 以符號鏈結方式將檔案鏈結到 `rc3.d` 目錄:

```
ln -s /etc/init.d/startmqmft /etc/rc3.d/S98startmqmft
```

字首 S 表示 Solaris 的預設狀態「已啟動」。98 是序號。字尾是來自 `init.d` 的檔名

HP-UX

HP-UX 在 HP-UX 上, 請遵循下列步驟:

1. 建立一個稱為 `/sbin/init.d/mqmft` 的檔案, 其內容如下:

```
#!/bin/sh
su -l mqmft_user -c mq_install_root/bin/fteStartAgent agent_name
```

2. 建立一個稱為 `/etc/rc.config.d/mqmft` 的檔案, 其內容如下:

```
MQMFT=1
```

3. 以符號鏈結方式將檔案鏈結到 `rc3.d` 目錄, 並啟動代理程式:

```
ln -s /sbin/init.d/mqmft /sbin/rc3.d/S84mqmft
```

Linux

Linux

若是 Linux 系統, 您有多種方式可以在系統開機程序期間啟動應用程式。一般而言, 請考慮執行下列步驟:

1. 建立一個稱為 `/etc/rc.mqmft` 的檔案, 其內容如下:

```
#!/bin/sh
su -l mqmft_user -c mq_install_root/bin/fteStartAgent agent_name"
```

其中 `mqmft_user` 是用來執行代理程式程序的使用者 ID。此使用者 ID 必須是 `mqm` 群組的成員。

2. 使該檔案成為執行檔, 例如:

```
chmod 755 /etc/rc.mqmft
```

3. 接下來, 將下面這一行新增至 `/etc/inittab`:

```
mqmft:5:boot:/etc/rc.mqmft
```

在 Linux 上開機期間啟動代理程式的其他方式包括將 `Script` 行新增至 `/etc/rc.d/rc.local` 檔案, 或在 Linux SuSe 上將 `Script` 行新增至 `/etc/init.d/boot.local` 檔案。您應該選取最適合您環境的方法。以下是在支援的特定 Linux 發行套件啟動期間, 啟動代理程式的其他方式的一些相關資訊:

SLES 10 及 11

若是 SUSE Linux Enterprise Server (SLES) 10 及 11 系統, 請遵循下列步驟:

1. 以系統 `root` 使用者 ID 建立您專屬的 `/etc/init.d/rc.rclocal` 檔案。
2. 將下列這幾行新增至 `rc.rclocal` 檔案:

```
#!/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. 執行下列指令：

```
chmod 755 rc.rclocal
chkconfig --add rc.rclocal
```

在 z/OS 上啟動 MFT 代理程式

在 z/OS 上，除了從「UNIX 系統服務」階段作業中執行 **fteStartAgent** 指令之外，您還可以從 JCL 以啟動作業的方式來啟動代理程式，而不需要互動式階段作業。

使用啟動的作業是因為它是以特定的使用者 ID 執行，且不受使用者登出影響。

註：  通常，據以執行啟動作業的管理使用者可能沒有登入專用權，並因此無法以執行代理程式的使用者身分登入 z/OS 系統。無法對該代理程式發出 **fteStartAgent**、**fteStopAgent** 和 **fteSetAgentTraceLevel** 指令以及已指定 **-d** 參數的 **fteShowAgentDetails** 指令。

從 IBM MQ 9.0.2 和 IBM MQ 9.0.0 Fix Pack 1 開始，代理程式內容 **adminGroup** 可與 z/OS 上的 Managed File Transfer 代理程式搭配使用。您可以定義安全管理員群組（例如 MFTADMIN），然後新增已啟動作業使用者 ID 和管理者 TSO ID 至此群組。編輯代理程式內容檔，並將 **adminGroup** 內容設為此安全管理員群組的名稱。

```
adminGroup=MFTADMIN
```

此群組的成員隨後可以對正在當作已啟動作業執行的代理程式發出 **fteStartAgent**、**fteStopAgent** 和 **fteSetAgentTraceLevel** 以及已指定 **-d** 參數的 **fteShowAgentDetails** 指令。

如需相關資訊，請參閱第 463 頁的『MFT agent.properties 檔案』中的 **adminGroup** 內容。

作為 Java 應用程式的代理程式是一個「UNIX 系統服務」應用程式，您可以透過使用 BFGAGSTP 成員從 JCL 執行、從代理程式已產生的 Managed File Transfer 指令 PDSE 程式庫資料集執行。如需如何建立 MFT 指令 PDSE 程式庫資料集，以及針對必要的代理程式自訂該資料集的相關資訊，請參閱第 54 頁的『建立 MFT 代理程式或日誌程式指令資料集』。

僅啟用代理程式與 IBM MQ Advanced for z/OS, Value Unit Edition 的遠端連線

z/OS 上在 IBM MQ Advanced for z/OS VUE 產品 ID (PID) 下執行的 MFT 代理程式可以使用用戶端連線連接至 z/OS 上的佇列管理程式。

當代理程式啟動時，它會將 BFGPR0137I 訊息寫入其事件日誌 (output0.log)，以顯示其執行所在的 PID。此訊息的範例如下：

```
BFGPR0137I: 已針對產品 'MQ z/OS MFT'、產品 ID '5655-MF9' 啟動產品使用情形資料記錄。
```

如需 IBM MQ 產品、其相關聯 PID 值及匯出分類的詳細資料，請參閱 [IBM MQ 產品 ID 及匯出資訊](#)。

z/OS 上以任何其他 PID 執行的 MFT 代理程式，都只能使用連結連線來連接至本端佇列管理程式。

z/OS 上的 MFT 代理程式只能連接至也在 z/OS 上執行的佇列管理程式，而不論 MFT PID 為何。

如果 IBM MQ Advanced for z/OS VUE 代理程式嘗試連接未在 z/OS 上執行的佇列管理程式，則會發出下列 [BFGMQ1044E](#) 訊息，並結束代理程式啟動：

```
BFGMQ1044E: z/OS 上的代理程式用戶端連線必須連接至 z/OS 上的佇列管理程式
```

相關參考

第 177 頁的『在 z/OS 上停止 MFT 代理程式』

如果您從 JCL 以已啟動作業的方式在 z/OS 上執行 Managed File Transfer Agent，則除了 **fteStopAgent** 指令之外，代理程式還接受 z/OS 操作員指令 **MODIFY** 及 **STOP**。

第 463 頁的『MFT agent.properties 檔案』

每一個 Managed File Transfer Agent 都有自己的內容檔 agent.properties，其中必須包含代理程式用來連接至其佇列管理程式的資訊。agent.properties 檔案也可以包含變更代理程式行為的內容。

啟動新的檔案傳送

您可以從「IBM MQ Explorer」或從指令行啟動新的檔案傳送，也可以選擇傳送單一檔案或以一個群組傳送多個檔案。

關於這項作業

若要從指令行啟動新的檔案傳送，請參閱 [fteCreateTransfer](#) 指令。

若要使用 IBM MQ Explorer 中的 **建立新的受管理檔案傳送** 精靈來啟動新的檔案傳送，請使用下列步驟：

程序

1. 在「導覽器」視圖中，按一下 **Managed File Transfer**。**Managed File Transfer 中心**即會顯示在「內容」視圖中。
2. 所有協調佇列管理程式都會顯示在「導覽器」視圖中。展開您要用於傳送的代理程式所登錄的協調佇列管理程式的名稱。如果目前所連接的不是您想要用於傳送的協調佇列管理程式，請在「導覽器」視圖中的協調佇列管理程式名稱上按一下滑鼠右鍵，然後按一下**斷線**。接著，用滑鼠右鍵按一下要使用的協調佇列管理程式名稱，並按一下**連接**。
3. 使用下列一種方法來啟動**建立新的 Managed File Transfer** 精靈：
 - a) 在「導覽器」視圖中，在下列任何節點名稱上按一下滑鼠右鍵：相關的協調佇列管理程式、**傳送範本**、**傳送日誌**或**擱置傳送**。然後，按一下**新建傳送**，以啟動精靈。
 - b) 按一下**檔案 > 新建 > 其他 > Managed File Transfer 精靈 > 新建傳送精靈**
4. 遵循精靈畫面上的指示操作。每一個畫面也提供上下文相關說明。若要存取 Windows 上的上下文相關說明，請按 F1 鍵。在 Linux 上，按 Ctrl+F1 或 Shift+F1。

使用傳送定義檔

您可以指定傳送定義檔，它可用來建立檔案傳送。此傳送定義檔是 XML 檔，它定義建立傳送需要的部分或全部資訊。

當您想要在單一傳送作業中指定多個來源檔案及多個目的地檔案時，傳送定義檔會很有幫助。您可以使用傳送定義檔來提交複式檔案傳送。您可以重複使用及共用傳送定義檔。

傳送定義檔可以使用兩種格式，這些格式大同小異，且都符合 FileTransfer.xsd 綱目。您可以在 Managed File Transfer 安裝架構的 samples\schema 目錄中找到這個綱目。

下列是兩種支援的傳送定義檔格式：

- 傳送的來源和目的地檔案的定義。此定義使用 <transferSpecifications> 元素作為根元素。
- 整個傳送的定義，包括來源和目的地檔案，以及來源和目的地代理程式。此定義使用 <request> 元素作為根元素。
 - 此格式的檔案可以透過使用 **-gt** 參數的 **fteCreateTransfer** 指令產生。

下列範例顯示的傳送定義檔格式只指定傳送的來源和目的地檔案：

```
<?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>
```

若要提交此格式的傳送定義檔，您必須在指令行上指定來源和目的地代理程式：

```
fteCreateTransfer -sa AGENT1 -sm agent1qm -da AGENT2 -dm agent2qm -td
c:\definitions\example1.xml
```

下列範例是一個傳送定義檔格式，它指定傳送需要的全部資訊：

```
<?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** 指令上使用 **-gt** 參數，以產生此格式的檔案。當您提交此格式的傳送定義檔時，不需要在指令行上指定任何其他項目：

```
fteCreateTransfer -td c:\definitions\example2.xml
```

除了傳送定義檔之外，您還可以傳入一般參數，以置換指令行的來源和目的地代理程式相關資訊。例如：

```
fteCreateTransfer -da AGENT9 -dm agent9qm -td c:\definitions\example2.xml
```

此範例使用指令行選項，將傳送定義檔內定義的目的地代理程式置換成 **AGENT9**，並將傳送定義檔內定義的目的地佇列管理程式置換成 **agent9qm**。

上述兩種格式都包含一個以上的 `<item>` 元素。For further information about the `<item>` element, see 檔案傳送要求訊息格式。這些傳送項目各自定義一對來源和目的地檔案，以及控制傳送行為的其他屬性。例如，您可以指定下列行為：

- 傳送是否使用總和檢查
- 傳送是文字還是二進位
- 傳送完成之後是否刪除來源檔案
- 是否改寫目的地檔案（如果該檔案存在的話）

使用傳送定義檔的優點是，您可以指定指令行無法提供的其他選項。例如，當您執行「訊息至檔案」的傳送時，可使用傳送定義檔來指定 `groupId` 屬性。此屬性指定從佇列中讀取的訊息的 IBM MQ 群組 ID。傳送定義檔的另一個優點是，您可以為每一對檔案指定不同的選項。例如，您可以指定是否使用總和檢查，或者是以文字模式還是以二進位模式逐一傳送檔案。如果您使用指令行，則每一個傳送檔案都套用相同的選項。

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>

```

z/OS

您可以使用項目，將檔案從分散式系統傳送至 z/OS 系統：

z/OS

```

<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>

```

z/OS

此範例是以文字模式，將 `textTransferTest.txt` 檔從來源代理程式傳送至目的地代理程式的資料集 `//TEXT.TRANS.TEST`。此傳送將來源資料從來源代理程式的預設編碼（未指定任何來源編碼屬性）轉換成字碼頁：IBM-1047。

建立排定的檔案傳送

您可以從「IBM MQ Explorer 探險家」或從指令行排定新的檔案傳送。排定的傳送可以包含群組中的單一檔案或多個檔案。您可以執行排定的檔案傳送一次，或多次重複傳送。

關於這項作業

您可以設定執行檔案傳送排程一次，或按下列間隔執行：

- 每分鐘
- 每小時
- 每天一次
- 每週一次
- 每月
- 每年

然後，您可以指定在下列情況下停止執行：

- 在定義的時間及日期
- 在定義的執行次數之後

或者，您可以指定不斷地繼續執行。

V 9.0.0.9 如果排定的傳送每天在同一時間執行，請使用代理程式內容檔中的 **adjustScheduleTimeForDaylightSaving** 屬性，來調整時鐘變更時排程發生的時間。如需相關資訊，請參閱 `MFT.agent.properties` 檔案。

若要使用指令行建立新的排定檔案傳送，請針對 `fteCreateTransfer` 指令使用排程參數 (`-tb`、`-ss`、`-oi`、`-of`、`-oc` 及 `-es`)。

若要使用 IBM MQ Explorer 中的 **建立新的受管理檔案傳送** 精靈來建立新的排定檔案傳送，請使用下列步驟：

程序

1. 在「導覽器」視圖中，按一下 **Managed File Transfer**。**Managed File Transfer 中心**即會顯示在「內容」視圖中。
2. 所有協調佇列管理程式都會顯示在「導覽器」視圖中。展開您要用於傳送的代理程式所登錄的協調佇列管理程式的名稱。如果目前所連接的不是您想要用於傳送的協調佇列管理程式，請在「導覽器」視圖中的協調佇列管理程式名稱上按一下滑鼠右鍵，然後按一下**斷線**。接著，用滑鼠右鍵按一下要使用的協調佇列管理程式名稱，並按一下**連接**。
3. 使用下列其中一種方法啟動 **建立新的 Managed File Transfer** 精靈：
 - a) 在「導覽器」視圖中，在下列任何節點名稱上按一下滑鼠右鍵：相關的協調佇列管理程式、**傳送範本**、**傳送日誌**或**擱置傳送**。然後，按一下**新建傳送**，以啟動精靈。
 - b) 按一下**檔案 > 新建 > 其他 > Managed File Transfer 精靈 > 新建傳送精靈**
4. 遵循精靈畫面上的指示操作。請確保已選取**啟用排程傳送**勾選框，然後在**排程**標籤上輸入您的排程明細。如果沒有可能會影響傳送的問題，排定的檔案傳送會在排程開始時間的一分鐘內啟動。例如，網路或代理程式可能發生問題，造成排定的傳送無法開始。每一個畫面都提供有上下文相關說明。若要存取 Windows 上的上下文相關說明，請按 F1 鍵。在 Linux 上，按 Ctrl+F1 或 Shift+F1。

結果

如需排定的檔案傳送所涉及訊息的相關資訊，請參閱排定的傳送的訊息格式。

處理擱置中檔案傳送

您可以透過「IBM MQ Explorer」來檢視擱置的排定檔案傳送。**擱置傳送**視窗會顯示所有已向您目前所連接的協調佇列管理程式進行登錄的擱置傳送。

關於這項作業


若要檢視尚未開始的排定檔案傳送的狀態，請使用下列步驟：

程序

1. 在「導覽器」視圖中展開 **Managed File Transfer**。**Managed File Transfer 中心**即會顯示在「內容」視圖中。
2. 所有協調佇列管理程式都會顯示在「導覽器」視圖中。展開您用於排定傳送的協調佇列管理程式的名稱。如果您想要變更所連接的協調佇列管理程式，請在「導覽器」視圖中您想要使用的協調佇列管理程式名稱上按一下滑鼠右鍵，然後再按一下**連接**。
3. 按一下**擱置傳送**。「**擱置傳送**」視窗即會顯示在「內容」視圖中。
4. 「**擱置傳送**」視窗會顯示下列關於排定檔案傳送的詳細資料：
 - a) **名稱** 排定檔案傳送的號碼。會自動指派此號碼。
 - b) **來源** 來源代理程式的名稱。
 - c) **來源檔案** 要在其主機系統上傳送的檔案名稱。
 - d) **目的地** 目的地代理程式的名稱。
 - e) **目的地檔案** 檔案傳送至目的地系統之後的檔案名稱。
 - f) **排定的開始時間（選取的時區）** 檔案傳送排定開始的時間和日期，以管理者選取的時區表示。若要變更顯示的時區，請按一下 **視窗 > 喜好設定 > IBM MQ Explorer > Managed File Transfer**，然後從**時區**：清單中選取替代時區。按一下**確定**。
 - g) **重複間隔** 如果您選擇要重複排定的傳送，此即為您要重複執行傳送的指定間隔，以數字表示。
 - h) **重複類型** 如果您選擇要重複排定的傳送，此即為您為檔案傳送指定的重複間隔類型。類型可以是下列其中一個值：**分鐘**、**小時**、**日**、**週**、**月**或**年**。

- i) **重複截止** 如果您選擇要重複排定的傳送，此即為您想要結束重複檔案傳送的詳細時間。例如，指定的日期和時間，或在指定的次數之後。

結果

若要重新整理**擱置傳送**視窗所顯示的內容，請按一下「內容」視圖工具列上的「重新整理」按鈕 。

若要取消擱置的檔案傳送，請在特定傳送按上按一下滑鼠右鍵，然後按一下**取消**。取消傳送後將會完全捨棄檔案傳送要求。

觸發檔案傳送

您可以設定檔案傳送的特定觸發條件，必須符合這些條件才會進行傳送。如果不符合觸發條件，則不會進行檔案傳送，而且會選擇性地提交日誌訊息，以記錄未進行傳送的事實。然後，會捨棄檔案傳送要求。例如，您可以設定只有當來源代理程式所在系統上之指定檔案超出指定的大小時，或特定指定檔案存在於來源代理程式所在的系統上時，才會進行檔案傳送。您可以從 IBM MQ Explorer 或從指令行設定觸發的檔案傳送。

關於這項作業

您可以持續監視資源直到滿足觸發條件。如需資源監視的進一步相關資訊，請參閱：[第 141 頁的『監視 MFT 資源』](#)。

您可以設定三個不同觸發條件。這些條件如下所示：

- 如果特定檔案存在於與來源代理程式相同的系統上
- 如果特定檔案不存在於與來源代理程式相同的系統上
- 如果特定檔案超出來源代理程式所在系統上的特定大小（此大小可以用位元組、KB、MB 或 GB 表示）。這些度量單位使用 2^{10} 慣例，例如，1KB 等於 1024 個位元組，1MB 等於 1024KB。

之前清單中的觸發類型有兩種結合方式：

- 若為單一條件，您可以在來源代理程式所在系統上指定多個檔案。如果任何一個指定的檔案符合該條件（布林運算子 OR），就會觸發傳送。
- 您可以指定多個條件。只有在符合所有條件時（布林運算子 AND），才會觸發傳送。

您也可以結合觸發的傳送與排定的傳送。如需相關資訊，請參閱建立排定的檔案傳送。在此情況下，會在排程應該開始時評估觸發條件，若為重複排程，則在每次排程應該開始時評估觸發條件。

在通訊協定橋接器代理程式上，並不支援觸發的傳送。

若要使用指令行建立觸發檔案傳送，請在 `fteCreateTransfer` 指令上使用 `-tr` 參數。

若要使用 IBM MQ Explorer 中的 **建立新的受管理檔案傳送** 精靈來建立排定的檔案傳送，請使用下列步驟：

程序

1. 在「導覽器」視圖中，按一下 **Managed File Transfer**。**Managed File Transfer 中心**即會顯示在「內容」視圖中。
2. 所有協調佇列管理程式都會顯示在「導覽器」視圖中。展開您用於排定傳送的協調佇列管理程式的名稱。如果您想要變更所連接的協調佇列管理程式，請在「導覽器」視圖中您想要使用的協調佇列管理程式名稱上按一下滑鼠右鍵，然後再按一下**連接**。
3. 使用下列一種方法來啟動**建立新的 Managed File Transfer** 精靈：
 - a) 在「導覽器」視圖中，在下列任何節點名稱上按一下滑鼠右鍵：相關的協調佇列管理程式、**傳送範本**、**傳送日誌**或**擱置傳送**。然後，按一下**新建傳送**，以開啟精靈。
 - b) 按一下**檔案 > 新建 > 其他 > Managed File Transfer 精靈 > 新建傳送精靈**
4. 遵循精靈畫面上的指示操作。確定您在**觸發**標籤上選取**啟用已觸發的傳送**勾選框，並完成該標籤上的欄位以設定觸發。每一個畫面都提供有上下文相關說明。若要存取 Windows 上的上下文相關說明，請按 F1 鍵。在 Linux 上，按 **Ctrl+F1** 或 **Shift+F1**。

監視正在進行的檔案傳送

您可以使用 IBM MQ Explorer 中的 **受管理檔案傳送-目前傳送進度** 標籤來監視進行中的檔案傳送。這可以是從「IBM MQ Explorer」或指令行啟動的檔案傳送。此標籤也會在排定的傳送開始時顯示該傳送的進度。

關於這項作業

如果您要使用「IBM MQ Explorer」來監視與遠端系統上協調佇列管理程式相關聯之傳送，請遵循第 139 頁的『[配置 IBM MQ Explorer 以監視遠端協調佇列管理程式](#)』主題中的指示。

在您停止並重新啟動「IBM MQ Explorer」之後，先前的檔案傳送資訊並不會保留下來。在重新啟動時，即會從**目前傳送進度**標籤中清除舊傳送項目的相關資訊。當 IBM MQ Explorer 開啟時，您可以隨時使用 **移除已完成的傳送** 來清除已完成的傳送。

程序

在使用「IBM MQ Explorer」或指令行啟動新的檔案傳送之後，您可以在**目前傳送進度**標籤中監視傳送進度。每一個進行中的傳送都會顯示下列資訊：

- a) **來源**。用來從來源系統傳送檔案的代理程式名稱。
- b) **目的地**。用來在目的地系統上接收檔案的代理程式名稱。
- c) **現行檔案**。目前所傳送的檔案名稱。個別檔案中已傳送的部分會以 B、KiB、MiB 顯示。GiB 或 TiB 連同檔案大小總計放在括弧內。顯示的測量單位視檔案大小而定。
B 是指每秒的位元組數。KiB/s 是每秒的 kibibyte，1 kibibyte 等於 1024 個位元組。MiB/s 是每秒的 mebibyte，1 mebibyte 等於 1,048,576 個位元組。GiB/s 是每秒的 gibibyte，1 gibibyte 等於 1,073,741,824 個位元組。TiB/s 是每秒的 tebibyte，1 tebibyte 等於 1,099,511,627,776 個位元組。
- d) **檔案數**。如果您傳送多個檔案，則此數代表整個檔案群組的傳送已進行至何處。
- e) **進度**。進度列會以百分比顯示目前的檔案傳送已完成多少。
- f) **速率**。以 KiB/s 表示檔案傳送的速率（每秒的 kibibyte，1 kibibyte 等於 1024 個位元組。）
- g) **開始時間（選取的時區）**。檔案傳送開始的時間，以管理者所選取的時區顯示。若要變更顯示的時區，請按一下 **視窗 > 喜好設定 > IBM MQ Explorer > Managed File Transfer**，然後從**時區**：清單中選取替代時區。按一下**確定**。

如果在傳送檔案的過程中傳送進入回復狀態，則開始時間將會更新，以反映檔案傳送回復的時間。

結果

此標籤會自動定期重新整理其資訊，但若要強制重新整理**目前傳送進度**標籤中的顯示內容，請按一下「內容」視圖工具列上的**重新整理**。

若要刪除**目前傳送進度**標籤中的檔案傳送，請按一下「內容」視圖工具列上的**移除已完成的傳送**。按此按鈕只會從標籤中移除檔案傳送詳細資料，並不會停止或取消現行或排定的傳送。

如果您想要在關閉**目前傳送進度**標籤後返回該處，只要按一下**視窗 > 顯示視圖 > 其他 > 其他 > Managed File Transfer - 目前傳送進度**，即可顯示該標籤。按一下**確定**。

下一步

此外，可以開發用於自訂檔案監視的應用程式。這可以透過建立適當 Managed File Transfer 管理主題的訂閱來達成（以程式化方式或系統管理方式），這樣監視器應用程式主可以接收該主題上發佈的 Managed File Transfer 檔案傳送活動。如需訂閱主題及發佈訊息格式的相關資訊，請參閱第 637 頁的『[檔案傳送進度訊息範例](#)』。

相關工作

第 139 頁的『[配置 IBM MQ Explorer 以監視遠端協調佇列管理程式](#)』

使用「IBM MQ Explorer」，可監視與遠端系統上所執行之協調佇列管理程式相關聯的檔案傳送。在 IBM WebSphere MQ 7.5 版或更新版本中，您需要能夠執行「IBM MQ Explorer」的系統。您必須安裝「IBM MQ Explorer」元件，才能連接遠端協調佇列管理程式。

第 139 頁的『在傳送日誌中檢視檔案傳送的狀態』

您可以使用 IBM MQ Explorer 中的 **傳送日誌** 來檢視檔案傳送的詳細資料。這些可能是從指令行或 IBM MQ Explorer 啟動的傳送。您也可以自訂在 **傳送日誌** 中顯示的內容。

配置 IBM MQ Explorer 以監視遠端協調佇列管理程式

使用「IBM MQ Explorer」，可監視與遠端系統上所執行之協調佇列管理程式相關聯的檔案傳送。在 IBM WebSphere MQ 7.5 版或更新版本中，您需要能夠執行「IBM MQ Explorer」的系統。您必須安裝「IBM MQ Explorer」元件，才能連接遠端協調佇列管理程式。

關於這項作業

假設：有權透過配置佇列管理程式以允許遠端連線，來連接至遠端協調佇列管理程式。

如需如何配置的相關資訊，請參閱第 45 頁的『在用戶端模式下使用通道鑑別連接至佇列管理程式』及第 295 頁的『管理 MFT 特定資源的權限』。

若要監視未執行 Windows 或 Linux 的系統上代理程式間的佇列管理程式及檔案傳送，請使用下列步驟將「IBM MQ Explorer」配置為連接至遠端系統：

程序

1. 啟動本端的「IBM MQ Explorer」。
2. 當「IBM MQ Explorer」載入後，請用滑鼠右鍵按一下 **Managed File Transfer** 資料夾，然後選取**新建配置**。
3. 繼續執行精靈，接著選取「協調佇列管理程式」與「指令佇列管理程式」，然後定義配置的名稱。
4. 按一下**完成**，以完成定義。
5. 定義完成時，用滑鼠右鍵按一下定義，然後選取**連接**。

結果

現在啟動「IBM MQ Explorer」，並使用它來監視與協調佇列管理程式相關聯之 Managed File Transfer 網路的傳送活動。

相關工作

第 138 頁的『監視正在進行的檔案傳送』

您可以使用 IBM MQ Explorer 中的 **受管理檔案傳送-目前傳送進度** 標籤來監視進行中的檔案傳送。這可以是從「IBM MQ Explorer」或指令行啟動的檔案傳送。此標籤也會在排定的傳送開始時顯示該傳送的進度。

第 139 頁的『在傳送日誌中檢視檔案傳送的狀態』


您可以使用 IBM MQ Explorer 中的 **傳送日誌** 來檢視檔案傳送的詳細資料。這些可能是從指令行或 IBM MQ Explorer 啟動的傳送。您也可以自訂在 **傳送日誌** 中顯示的內容。

在傳送日誌中檢視檔案傳送的狀態

您可以使用 IBM MQ Explorer 中的 **傳送日誌** 來檢視檔案傳送的詳細資料。這些可能是從指令行或 IBM MQ Explorer 啟動的傳送。您也可以自訂在 **傳送日誌** 中顯示的內容。

程序



1. 在「導覽器」視圖中展開 **Managed File Transfer**，然後展開您想要檢視傳送日誌的協調佇列管理程式的名稱。
2. 在「導覽器」視圖中按一下**傳送日誌**。「**傳送日誌**」即會顯示在「內容」視圖中。
3. 「**傳送日誌**」視窗會顯示下列關於檔案傳送的詳細資料：
 - a) **來源** 來源檔案所在系統上的代理程式名稱。
 - b) **目的地** 您想要傳送檔案至其中的目的地系統上的代理程式名稱。
 - c) **完成狀態** 檔案傳送的狀態。其狀態可以是下列其中一個值：「已啟動」、「進行中」、「成功」、「部分成功」、「已取消」或「失敗」。

- d) 擁有者 主機上提交傳送要求的使用者 ID。
- e) 開始時間 (選取的時區) Managed File Transfer 代理程式接受檔案傳送要求的時間和日期，以管理者選取的時區表示。若要變更顯示的時區，請按一下視窗 > 喜好設定 > IBM MQ Explorer > Managed File Transfer，然後從時區：清單中選取替代時區。按一下確定。
- f) 狀態記錄 (選取的時區) (依預設，不顯示此直欄。您可以使用配置傳送日誌直欄  視窗來選擇要顯示此直欄。) 記錄完成狀態的日期和時間，以管理者選取的時區表示。
- g) 工作名稱 使用者使用 `fteCreateTransfer` 的 `-jn` 參數或在 Ant Script 中指定的 ID
- h) 傳送 ID 檔案傳送的唯一 ID。
 - i) **Connect: Direct** 會列出有關程序號碼、程序名稱、主要節點、次要節點、來源類型和目的地類型的詳細資料。

結果

註：在 APAR IC99545 中，IBM MQ 8.0.0 Fix Pack 1 已變更「傳送日誌」的內部格式。因此，如果 IBM MQ Explorer 升級至 8.0.0.1 或更新版本，然後還原至 8.0.0.0，則對於 IBM MQ Explorer 在 8.0.0.1 時發生的傳送，不會顯示任何審核 XML。在「內容」視窗中，這些傳送的 XML 畫面只包含空文字框。

若要檢視已完成傳送的進一步詳細資料，請按一下加號 (+) 來展開您感興趣的傳送。然後，您可以查看該傳送中包含的所有來源及目的地檔案名稱。不過，如果傳送目前正在進行，而且是由許多檔案組成，則您只能檢視到目前為止已傳送的檔案。

若要重新整理傳送日誌中顯示的內容，請按一下「內容」視圖工具列上的重新整理按鈕 。在您停止並重新啟動 IBM MQ Explorer 之後，「傳送日誌」中的檔案傳送資訊會保留在日誌中。如果您想要刪除日誌中所有已完成的檔案傳送，請按一下「內容」視圖工具列上的移除已完成的傳送 .

若要從日誌中刪除個別的已完成檔案傳送，請在該傳送上按一下滑鼠右鍵，然後按一下刪除。如果您刪除傳送，它不會停止或取消進行中或已排定的傳送；您只會刪除已儲存的歷程資料。

若要將傳送的唯一 ID 複製到剪貼簿，請在該傳送上按一下滑鼠右鍵，然後按一下複製 ID。

利用蹦現功能表的內容動作，即可存取傳送的 meta 資料及完整的審核 XML。

相關概念

第 206 頁的『回復中檔案傳送的逾時選項』

您可以設定時間量 (以秒為單位)，在此期間內，來源代理程式會一直嘗試回復已停止的檔案傳送。如果代理程式達到重試間隔的逾時值時傳送未順利完成，則傳送失敗。

相關工作

第 138 頁的『監視正在進行的檔案傳送』

您可以使用 IBM MQ Explorer 中的受管理檔案傳送-目前傳送進度 標籤來監視進行中的檔案傳送。這可以是從「IBM MQ Explorer」或指令行啟動的檔案傳送。此標籤也會在排定的傳送開始時顯示該傳送的進度。

第 140 頁的『配置傳送日誌』

您可以在 IBM MQ Explorer 中配置要顯示哪些資訊，以及如何在 傳送日誌 中顯示資訊。


配置傳送日誌

您可以在 IBM MQ Explorer 中配置要顯示哪些資訊，以及如何在 傳送日誌 中顯示資訊。


關於這項作業

若要重新排列傳送日誌中直欄的順序，請按一下要移動的直欄標題，然後將直欄拖曳至新位置。新直欄順序僅保留到您下一次停止並重新啟動「IBM MQ Explorer」為止。

若要過濾傳送日誌中的項目，請在過濾顯示的日誌項目欄位中輸入字串。若要將所有項目還原回日誌，請刪除您在欄位中輸入的字串。您可以在此欄位中使用任何有效的 Java 正規表示式。如需相關資訊，請參閱第 554 頁的『MFT 所使用的正規表示式』。

若要自訂「傳送日誌」中顯示的直欄，請使用配置傳送日誌直欄 。按照下列步驟啟動並使用配置傳送日誌直欄視窗。

程序

1. 確保您已在「內容」視圖中開啟**傳送日誌**。按一下「內容」視圖工具列上的**配置傳送日誌直欄** 。配置**傳送日誌直欄**視窗即會開啟。
2. 若要自訂**傳送日誌**的視圖，請針對您要顯示或隱藏的直欄選取或清除個別勾選框。您可以按一下**全選**，然後按一下**確定**以選取所有勾選框，或按一下**取消全選**，然後按一下**確定**以清除所有勾選框。

相關工作

第 138 頁的『[監視正在進行的檔案傳送](#)』

您可以使用 IBM MQ Explorer 中的 **受管理檔案傳送-目前傳送進度** 標籤來監視進行中的檔案傳送。這可以是從「IBM MQ Explorer」或指令行啟動的檔案傳送。此標籤也會在排定的傳送開始時顯示該傳送的進度。

第 139 頁的『[在傳送日誌中檢視檔案傳送的狀態](#)』

您可以使用 IBM MQ Explorer 中的 **傳送日誌** 來檢視檔案傳送的詳細資料。這些可能是從指令行或 IBM MQ Explorer 啟動的傳送。您也可以自訂在**傳送日誌**中顯示的內容。

監視 MFT 資源

您可以監視 Managed File Transfer 資源；例如，佇列或目錄。滿足此資源的條件時，資源監視器會啟動作業，例如檔案傳送。您可以在 IBM MQ Explorer 的 Managed File Transfer 外掛程式中使用 **ftCreateMonitor** 指令或「**監視器**」視圖來建立資源監視器。

關於這項作業

Managed File Transfer 資源監視使用下列術語：

monitor

程序，依預先定義的固定間隔來輪詢資源（例如目錄或佇列），以查看是否變更資源內容。如果發現變更，則比較內容與此監視器的條件組。如果符合，則開始執行此監視器的作業。

資源

監視器在每次輪詢間隔檢查的系統資源，會與觸發條件相互比較。受監視資源包括佇列、目錄或巢狀目錄結構。

條件 (condition)

評估的表示式（通常根據受監視資源的內容來評估）。如果表示式評估為 true，則條件會組成整體觸發條件。

觸發條件

整體條件，在滿足所有條件時就算滿足。滿足觸發條件時，作業可繼續執行。

作業

滿足觸發條件或條件組時啟動的作業。支援的作業包含檔案傳送及指令呼叫。

觸發檔

放入監視目錄中的檔案，指出可開始執行作業（通常是傳送）。例如，可能指出所有要處理的檔案已送達已知位置，可進行傳送或處理。觸發檔案名稱可用來指定要傳送的檔案（使用變數替代）。如需相關資訊，請參閱 [第 149 頁的『使用變數替代來自訂 MFT 資源監視器作業』](#)。

觸發檔案也稱為備妥檔案或執行檔案。不過，在本文件中，一律稱為觸發檔案。

通訊協定橋接器代理程式或 Connect:Direct 橋接器代理程式上不支援資源監視。

MFT 資源監視概念

Managed File Transfer 資源監視特性的主要概念概觀。

監視器

資源監視器與 Managed File Transfer 代理程式相關聯，只在該代理程式啟動時才會作用。監視代理程式停止時，監視器也就停止。建立監視器時，如果代理程式已啟動，監視器會立即啟動。監視代理程式也必須是監視器所起始之作業的來源代理程式。

監視器名稱在其代理程式內必須是唯一的。監視器名稱長度必須至少為一個字元，且不得包含星號(*)、百分比(%)或問號(?)字元。提供的監視器名稱不區分大小寫，監視器名稱會轉換為大寫。如果嘗試使用已存在的名稱建立監視器，則會忽略要求，且會將此嘗試記載到監視器日誌主題。

代理程式上可建立的監視器數目沒有限制，且執行的優先順序全部相同。請考慮重疊的受監視資源、衝突的觸發條件及資源輪詢頻率的含意。

重疊資源監視器可能導致：

- 來源位置/項目的可能競用。
- 相同來源項目的可能重複傳送要求。
- 由於來源項目衝突，傳送發生非預期的錯誤或失敗。

如果多個監視器掃描相同的位置，並且可以在相同的項目上觸發，則您可能會遇到兩個不同監視器針對相同項目提交受管理傳送要求的問題。

資源監視器會在每一個輪詢間隔期間之後查看資源的內容。資源的內容會與觸發條件相互比較，如果滿足這些條件，則會呼叫與監視器相關聯的作業。

作業會非同步地啟動。如果條件符合，且作業已啟動，則監視器會持續輪詢資源內容的進一步變更。因此，舉例來說，如果因為一個稱為 `reports.go` 的檔案到達監視目錄中而發現相符項目，則作業會啟動一次。在下次輪詢間隔時，即使該檔案仍存在，作業就不會再次啟動。不過，如果檔案被刪除，然後再次放入目錄中，或檔案已更新（亦即前次修改日期屬性變更），則下一次檢查觸發條件時會導致再次呼叫作業。

資源

Managed File Transfer 中的資源監視器可以輪詢下列兩種類型資源的內容：

目錄或巢狀目錄結構

一種常見的情況是監視目錄中是否有觸發檔案。外部應用程式可能處理多個檔案，並將它們放在已知的來源目錄。應用程式完成處理之後，表示可透過將觸發檔案放入受監視的位置以準備傳送或處理檔案。Managed File Transfer 資源監視器可以偵測觸發檔案，並起始將這些檔案從來源目錄傳送至另一個 Managed File Transfer Agent。

依預設，會監視指定的目錄。如果也要檢查子目錄，請在 `fteCreateTransfer` 指令中設定遞迴層次。

兩個監視目錄範例，如下所示：

- 監視觸發檔案 (例如 `trigger.file`)，然後傳送萬用字元 (例如 `*.zip`)。
- 監視 `*.zip`，然後傳送 `${FilePath}` (例如，觸發傳送的檔案)。如需變數替代的相關資訊，請參閱第 149 頁的『[使用變數替代來自訂 MFT 資源監視器作業](#)』。

註：請不要建立監視器來監視 `*.zip` 又傳送 `*.zip`。此監視器會對系統上的每個 `.zip` 檔開始進行 `*.zip` 的傳送。亦即，監視器會為 `*.zip` 產生 `*` 次傳送。

如需建立資源監視器以監視目錄的範例，請參閱 [第 148 頁的『監視目錄及使用變數替代』](#)。

IBM MQ 佇列

監視佇列的範例是外部應用程式可能正在產生訊息，並將它們放置在具有相同群組 ID 的已知佇列上。當應用程式完成將訊息放置在佇列上後，它會指出是完整群組。Managed File Transfer 資源監視器可以偵測完整的訊息群組，並起始將訊息群組從來源佇列傳送至檔案。如需建立資源監視器來監視佇列的範例，請參閱第 149 頁的『[範例：配置 MFT 資源](#)』。

註：每個佇列只能指定一個監視器。如果指定多個監視器來輪詢 IBM MQ 佇列，將發生無法預期的行為。

不支援監視資料集。

觸發條件

當資源包含的值符合其他一些字串或型樣時，即符合條件。條件可以是下列其中一項：

- 檔案名稱 (型樣) 相符
- 檔案名稱 (型樣) 不符

- 檔案大小
- 在許多次輪詢中，如果檔案大小維持相同，則符合

檔案名稱比對可以表示為：

- 字串完全相符
- [第 549 頁的『搭配使用萬用字元與 MFT』](#) 中說明的簡式萬用字元相符
- 正規表示式相符

也可以使用萬用字元或 Java 正規表示式來識別絕不相符的檔名，從檔案名稱比對作業中排除檔案名稱。

偵測到相符的檔案時，會保留前次修改時間的時間戳記。如果後續輪詢偵測到檔案已變更，則再次滿足觸發條件時，作業就會啟動。如果在檔案不存在時偵測條件，且監視目錄中沒有任何檔案符合檔案名稱型樣，則會啟動作業。如果後來新增至目錄中的檔案符合檔案名稱型樣，則只有在該檔案被刪除時才會啟動作業。

作業

Managed File Transfer 支援下列兩種作業，可讓您配置為由資源監視器啟動：

- 檔案傳送
- 指令

檔案傳送作業的定義方式與任何其他檔案傳送相同。產生監視器所需的作業 XML 最有用的作法是執行 `fteCreateTransfer` 指令並指定 `-gt` 參數。此指令會產生 XML 文件形式的作業定義，包括傳送規格。然後，您可以在 `fteCreateMonitor` 指令上傳遞作業 XML 文件的名稱作為 `-mt` 參數的值。**fteCreateMonitor** 執行時會讀取作業 XML 文件。執行 **fteCreateMonitor** 之後，監視器不會再使用作業 XML 檔案的任何變更。

指令作業可以執行 Ant Script、呼叫可執行程式或執行 JCL 工作。如需相關資訊，請參閱[配置監視器作業來呼叫指令及 Script](#)。

使用檔案傳送作業時，您可以選取要將多少觸發條件分批放入作業中。預設是一個觸發條件啟動一個作業。您可以執行 `fteCreateMonitor` 指令並搭配 `-bs` 選項，以選取要整批放入一個作業中的觸發條件數目。

備份及還原資源監視器

您可以備份已經定義的資源監視器，以便未來可重複使用它們。您可以使用不同選項，如下所示：

- 搭配使用 **fteCreateMonitor** 指令與 `-ox` 參數可將資源監視器配置匯出至 XML 檔，搭配 `-ix` 參數可透過從 XML 檔匯入資源監視器配置來還原資源監視器。
- 搭配使用 **fteListMonitors** 指令與 `-ox`，可將單一資源監視器的定義匯出至 XML 檔案。
- **V 9.0.5** 從 IBM MQ 9.0.5 中，搭配使用 **fteListMonitors** 指令與 `-od`，以將多個資源監視器定義匯出至指定的目錄。每個資源監視器定義皆會儲存至個別 XML 檔。您也可以使用 `-od` 選項，將單一資源監視器定義匯出至指定的目錄。

如需相關資訊，請參閱 [第 160 頁的『備份及還原 MFT 資源監視器』](#)。

資源監視器記載

V 9.0.3

從 IBM MQ 9.0.3 開始，Managed File Transfer 會併入資源監視器記載。如需相關資訊，請參閱 [第 159 頁的『記載 MFT 資源監視器』](#)。

相關概念

[第 149 頁的『使用變數替代來自訂 MFT 資源監視器作業』](#)

滿足作用中資源監視器的觸發條件時，會呼叫所定義的作業。除了每次以相同的目的地代理程式或相同的目的地檔案名稱呼叫傳送或指令作業之外，您還可以在執行時期修改作業定義。您可以將變數名稱插入作業定義 XML 中，來達成此目的。當監視器判斷已滿足觸發條件，且作業定義包含變數名稱時，它會以變數值替換變數名稱，然後呼叫該作業。

相關工作

第 144 頁的『[配置 MFT 監視作業以啟動指令及 Script](#)』

資源監視器不限於將檔案傳送作為其關聯的作業執行。您也可以配置監視器以從監視代理程式呼叫其他指令，包括可執行程式、Ant Script 或 JCL 工作。若要呼叫指令，請編輯監視作業定義 XML，以併入具有對應指令呼叫參數的多個指令元素，如引數及內容。

第 149 頁的『[範例：配置 MFT 資源](#)』

您可以搭配使用 **-mq** 參數與 **fteCreateMonitor** 指令，將 IBM MQ 佇列指定為要由資源監視器監視的資源。

第 154 頁的『[監視佇列及使用變數替代](#)』

您可以使用 **fteCreateMonitor** 指令來監視佇列，並將訊息從受監視佇列傳送至檔案。在第一個要從受監視佇列讀取的訊息中，任何 IBM MQ 訊息內容的值都可代入作業 XML 定義中，並用來定義傳送行為。

相關參考

第 348 頁的『[fteCreateMonitor: 建立 MFT 資源監視器](#)』

fteCreateMonitor 指令可從指令行建立並啟動新的資源監視器。您可以使用 Managed File Transfer 來監視資源（例如目錄的內容），以在符合觸發條件時，啟動指定的作業（例如檔案傳送）。

第 401 頁的『[fteListMonitors: 列出 MFT 資源監視器](#)』

使用 **fteListMonitors** 指令，可讓您透過指令行列出 Managed File Transfer 網路中的所有現有資源監視器。

第 393 頁的『[fteDeleteMonitor: 刪除 MFT 資源監視器](#)』

利用指令行，使用 **fteDeleteMonitor** 指令來停止並刪除現有 Managed File Transfer 資源監視器。針對資源監視代理程式發出此指令。

配置 MFT 監視作業以啟動指令及 Script

資源監視器不限於將檔案傳送作為其關聯的作業執行。您也可以配置監視器以從監視代理程式呼叫其他指令，包括可執行程式、Ant Script 或 JCL 工作。若要呼叫指令，請編輯監視作業定義 XML，以併入具有對應指令呼叫參數的多個指令元素，如引數及內容。

關於這項作業

您希望監視代理程式呼叫的可執行程式、Ant Script 或 JCL 工作的檔案路徑必須併入監視代理程式的 `commandPath` 中。如需指令路徑內容的相關資訊，請參閱 [第 306 頁的『commandPath MFT 內容』](#)。

您可以按下列其中一種方式建立作業定義 XML 文件：

- 根據 `FileTransfer.xsd` 綱目手動建立作業定義 XML 文件。如需相關資訊，請參閱 [第 144 頁的『根據綱目手動建立作業定義 XML』](#)。
- 編輯 **fteCreateTransfer -gt** 參數產生的 XML 文件，作為您的作業定義的基礎。如需相關資訊，請參閱 [第 146 頁的『修改產生的文件來建立作業定義文件』](#)。

無論想要傳送作業還是指令作業，作業定義必須以 `<request>` 根元素開頭。`<request>` 的子元素必須為 `<managedTransfer>` 或 `<managedCall>`。在具有要執行的單一指令或 Script 時，您通常要選擇 `<managedCall>`；如果您希望作業併入檔案傳送並選擇性地併入最多四個指令呼叫，則選擇 `<managedTransfer>`。

根據綱目手動建立作業定義 XML

關於這項作業

您可以根據綱目 `FileTransfer.xsd` 手動建立作業定義 XML 檔案。您可以在 `MQ_INSTALLATION_PATH/mqft/samples/schema` 中找到此綱目。如需此綱目的相關資訊，請參閱 [第 668 頁的『檔案傳送要求訊息格式』](#)。

範例

下列範例顯示儲存為 `cleanup.xml` 的作業定義 XML 文件範例，它使用 `<managedCall>` 元素來呼叫稱為 `RunCleanup.xml` 的 Ant Script。 `RunCleanup.xml` Ant Script 必須位於監視代理程式的 `commandPath` 上。

```
<?xml version="1.0" encoding="UTF-8"?>
<request version="4.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedCall>
    <originator>
      <hostName>hostName</hostName>
      <userID>userID</userID>
      <mqmdUserID>mqmdUserID</mqmdUserID>
    </originator>
    <agent QMgr="QM1" agent="AGENT1"/>
    <reply QMGR="QM1">reply</reply>
    <transferSet priority="1">
      <metaDataSet>
        <metaData key="name1">value1</metaData>
      </metaDataSet>
      <call>
        <command name="RunCleanup.xml" type="antscript" retryCount="2"
          retryWait="30" successRC="0">
          <target>check_exists</target>
          <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>
</managedCall>
</request>
```

`<agent>` 元素指定在其 `commandPath` 上使用具名 Ant Script 所配置的 Managed File Transfer Agent。

`<call><command>...` 結構定義您要執行的執行檔或 Script。此指令會採用選用的 `type` 屬性，此屬性可能具有下列其中一個值：

antscript

在個別 JVM 中執行 Ant Script。

executable

呼叫可執行程式。

jcl

呼叫 JCL 工作。

如果您省略 `type` 屬性，將使用預設值 `executable`。

`name` 屬性指定您要執行的 Ant Script、執行檔或 JCL 工作的名稱，而無任何路徑資訊。代理程式會在代理程式的 `agent.properties` 檔案中的 `commandPath` 內容指定的位置中，搜尋 Script 或程式。

`retrycount` 屬性會在程式未傳回成功回覆碼時，指定再次嘗試呼叫程式的次數。指派給此屬性的值不得為負數。如果未指定 `retrycount` 屬性，則會使用預設值零。

`retrywait` 屬性指定在再次嘗試呼叫程式之前等待的時間（以秒為單位）。指派給此屬性的值不得為負數。如果未指定 `retrywait` 屬性，則會使用預設值零。

`successrc` 屬性是用於決定何時順利執行程式呼叫的表示式。指令的程序回覆碼會使用此表示式進行評估。值可能由多個表示式結合垂直線 (|) 字元（表示布林 OR）或 '&' 符號字元（表示布林 AND）組成。每一個表示式可以是下列其中一種表示式類型：

- 一個數字，表示程序回覆碼與此數之間的「相等」測試。
- 一個以大於字元 (>) 為字首的數字，表示數字與程序回覆碼之間的大於測試。
- 一個以小於字元 (<) 為字首的數字，表示數字與程序回覆碼之間的小於測試。

- 一個以驚嘆號字元 (!) 為字首的數字，表示數字與程序回覆碼之間的不等於測試。例如：>2&<7&!5|0|14 將下列回覆碼解譯為成功回覆碼：0、3、4、6、14。其餘全都解譯為不成功回覆碼。

如果未指定 `successrc` 屬性，則會使用預設值零。這表示只有在回覆碼為零時，才會判定指令已順利執行。

對於 Ant Script，您通常要指定 `<target>` 及 `<property>` 元素。`<target>` 元素值必須與 Ant Script 中的目標名稱相符。

對於可執行程式，您可以指定 `<argument>` 元素。巢狀引數元素指定傳遞至作為程式呼叫一部分呼叫的程式。按遇到引數元素的順序透過引數元素指定的值建立程式引數。您可以指定零個以上引數元素作為程式呼叫的巢狀元素。

管理者使用包含 `<managedCall>` 元素的作業定義 XML 文件來照常定義及啟動監視器。例如：

```
fteCreateMonitor -ma AGENT1 -mm QM1 -md /monitored -mn MONITOR01 -mt
/tasks/cleanuptask.xml -pi 30 -pu seconds -tr match,*go
```

傳送定義 XML 文件的路徑必須在您從中執行 **fteCreateMonitor** 指令的本端檔案系統上（在此範例中為 `/tasks/cleanuptask.xml`）。`cleanuptask.xml` 文件僅用於建立資源監視器。`cleanuptask.xml` 文件參照的任何作業（Ant Scripts 或 JCL 工作）必須在監視代理程式的指令路徑中。當符合監視觸發條件時，使用監視器中的實際值替代作業定義 XML 中的任何變數。例如，在傳送至代理程式的要求訊息中，使用 `/monitored/cleanup.go` 取代 `${FilePath}`。要求訊息放在代理程式指令佇列上。指令處理器偵測到要求適用於程式呼叫，並啟動指定的程式。如果呼叫類型 `antscript` 的指令，則啟動新的 JVM，然後 Ant 作業在新的 JVM 下執行。如需使用變數替代的相關資訊，請參閱 [使用變數替代值來自訂作業](#)。

相關概念

第 149 頁的『[使用變數替代來自訂 MFT 資源監視器作業](#)』

滿足作用中資源監視器的觸發條件時，會呼叫所定義的作業。除了每次以相同的目的地代理程式或相同的目的地檔案名稱呼叫傳送或指令作業之外，您還可以在執行時期修改作業定義。您可以將變數名稱插入作業定義 XML 中，來達成此目的。當監視器判斷已滿足觸發條件，且作業定義包含變數名稱時，它會以變數值替換變數名稱，然後呼叫該作業。

相關參考

第 668 頁的『[檔案傳送要求訊息格式](#)』

檔案傳送是由送達代理程式指令佇列的 XML 訊息起始，通常是由使用者發出檔案傳送指令或使用「IBM MQ Explorer」外掛程式來進行。The transfer request XML must conform to the `FileTransfer.xsd` schema and have the `<request>` element as the root element. `FileTransfer.xsd` 綱目文件位於 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目錄中。`FileTransfer.xsd` 綱目會匯入 `fteutils.xsd`，它位於相同的目錄中。

第 306 頁的『[commandPath MFT 內容](#)』

使用 `commandPath` 內容來指定 Managed File Transfer 可以從中執行指令的位置。在設定此內容時請特別小心，因為可以從遠端用戶端系統有效地呼叫其中一個指定的 `commandPaths` 中的任何指令，從而可以將指令傳送至代理程式。

修改產生的文件來建立作業定義文件

關於這項作業

您可以透過修改 `fteCreateTransfer` 的 `-gt` 選項產生的 XML 文件，來建立監視作業定義文件。產生的文件具有 `<request>`，後面接著 `<managedTransfer>` 元素。若要將此作業定義轉換為有效的 `<managedCall>` 結構，請遵循下列步驟：

程序

1. 將 `<managedTransfer>` 的開始和結束標籤取代為 `<managedCall>` 標籤。
2. 移除任何 `<schedule>` 元素及子節點。
3. 將 `<sourceAgent>` 的開始和結束標籤取代為 `<agent>`，以與監視代理程式配置詳細資料相符。
4. 移除 `<destinationAgent>` 和 `<trigger>` 元素。

5. 移除 `<item>` 元素。
6. 在 `<transferSet>` 元素內插入新的 `<call>...</call>` 結構。此結構包含指令定義，如下列範例所示：

```
<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>
```

範例

您也可以保留 `<managedTransfer>` 元素（包括所有檔案傳送詳細資料），並插入最多四個指令呼叫。在此情況下，您可以在 `<metaDataSet>` 與 `<item>` 元素之間插入下列呼叫元素的任何選項：

preSourceCall

在啟動傳送之前，呼叫來源代理程式上的程式。

postSourceCall

在完成傳送之後，呼叫來源代理程式上的程式。

preDestinationCall

在啟動傳送之前，呼叫目的地代理程式上的程式。

postDestinationCall

在完成傳送之後，呼叫目的地代理程式上的程式。

其中每一個元素都會採用 `<command>` 元素結構，如較早範例所述。FileTransfer.xsd 綱目定義各種呼叫元素使用的類型。

下列範例顯示作業定義文件中的 `preSourceCall`、`postSourceCall`、`preDestinationCall` 及 `postDestinationCall`：

```
:
<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">
:
```

您可以在傳送中混用不同類型的指令。引數、目標及內容元素是選用的。

監視目錄及使用變數替代

您可以使用 **fteCreateMonitor** 指令來監視目錄。替代變數的值可替換至作業 XML 定義中，並用來定義傳送行為。

關於這項作業

在此範例中，來源代理程式稱為 AGENT_HOP。AGENT_HOP 所監視的目錄稱為 /test/monitored。代理程式每隔 5 分鐘會輪詢目錄一次。

在 .zip 檔寫入至目錄後，將檔案寫入目錄的應用程式會在相同的目錄中撰寫一個觸發檔案。觸發檔案的名稱與 .zip 檔相同，但副檔名不同。例如，在 file1.zip 檔寫入至目錄後，即會在該目錄中撰寫 file1.go 檔。資源監視器會監視該目錄中符合型樣 *.go 的檔案，然後使用變數替代要求傳送關聯的 .zip 檔。

程序

1. 建立作業 XML，以定義監視器被觸發後所執行的作業。

```
<?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>
```

取代為與觸發檔案相關聯值的變數，會以**粗體**強調顯示。此作業 XML 會儲存至 /home/USER1/task.xml 檔。

2. 建立用來監視目錄 /test/monitored 的資源監視器。

提交下列指令：

```
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. 使用者或程式會將 jump.zip 檔寫入 /test/monitored 目錄，然後再將 jump.go 檔寫入該目錄。
4. jump.go 檔的存在會觸發監視器。代理程式會將觸發檔案的相關資訊代入作業 XML 中。

這會導致作業 XML 轉換成：

```
<?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>
```

結果

執行作業 XML 所定義的傳送。AGENT_HOP 會從 /test/monitored 目錄中讀取 jump.zip 檔，且該檔案會傳送至 AGENT_SKIP 執行所在系統上稱為 /out/jump.zip 的檔案。

相關概念

第 149 頁的『[使用變數替代來自訂 MFT 資源監視器作業](#)』

滿足作用中資源監視器的觸發條件時，會呼叫所定義的作業。除了每次以相同的目的地代理程式或相同的目的地檔案名稱呼叫傳送或指令作業之外，您還可以在執行時期修改作業定義。您可以將變數名稱插入作業定義 XML 中，來達成此目的。當監視器判斷已滿足觸發條件，且作業定義包含變數名稱時，它會以變數值替換變數名稱，然後呼叫該作業。

相關工作

第 144 頁的『[配置 MFT 監視作業以啟動指令及 Script](#)』

資源監視器不限於將檔案傳送作為其關聯的作業執行。您也可以配置監視器以從監視代理程式呼叫其他指令，包括可執行程式、Ant Script 或 JCL 工作。若要呼叫指令，請編輯監視作業定義 XML，以併入具有對應指令呼叫參數的多個指令元素，如引數及內容。

相關參考

第 348 頁的『[fteCreateMonitor: 建立 MFT 資源監視器](#)』

fteCreateMonitor 指令可從指令行建立並啟動新的資源監視器。您可以使用 Managed File Transfer 來監視資源（例如目錄的內容），以在符合觸發條件時，啟動指定的作業（例如檔案傳送）。

範例：配置 MFT 資源

您可以搭配使用 **-mq** 參數與 **fteCreateMonitor** 指令，將 IBM MQ 佇列指定為要由資源監視器監視的資源。

關於這項作業

在此範例中，要監視的資源是佇列 **MONITORED_QUEUE**。此佇列必須位於監視代理程式的佇列管理程式 **QM_NEPTUNE** 上。監視佇列的條件是有完整訊息群組存在。符合條件時所要執行的作業，定義於 **task.xml** 檔中。

註：請不要建立多個資源監視器來監視個別佇列。這麼做可能會導致無法預期的行為。

程序

請鍵入下列指令：

```
fteCreateMonitor -ma AGENT_NEPTUNE -mn myMonitor -mm QM_NEPTUNE -mq MONITORED_QUEUE
-mt task.xml -tr completeGroups -pi 5 -pu minutes
```

監視器每五分鐘會檢查佇列一次，以查看條件 **completeGroups** 是否成立。如果佇列上有一個以上完整群組存在，監視器即會逐一針對各個完整群組執行 **task.xml** 檔中定義的作業。

使用變數替代來自訂 MFT 資源監視器作業

滿足作用中資源監視器的觸發條件時，會呼叫所定義的作業。除了每次以相同的目的地代理程式或相同的目的地檔案名稱呼叫傳送或指令作業之外，您還可以在執行時期修改作業定義。您可以將變數名稱插入作業定義 XML 中，來達成此目的。當監視器判斷已滿足觸發條件，且作業定義包含變數名稱時，它會以變數值替換變數名稱，然後呼叫該作業。



小心: 變數名稱不區分大小寫。

用於替代的變數只可用於正數觸發條件。只有 `match` 及 `fileSize` 觸發條件會替換變數。如果使用 `noMatch` 條件，且作業定義中有替代變數名稱，則不會呼叫作業，且監視器會產生回覆碼 110 及錯誤訊息 BFGDM0060E。

如果受監視資源是佇列

在第一個要從受監視佇列讀取的訊息中，任何 IBM MQ 訊息內容的值都可代入作業 XML 定義中。

使用者定義訊息內容的字首為 `usr.`，但在變數名稱中不包括此字首。變數名稱必須以大括弧 `{}` 括住，且前面必須是錢幣符號 (`$`) 字元。

例如，`${destFileName}` 取代為要從來源佇列中讀取的第一個訊息的 `usr.destFileName` 訊息內容的值。如需相關資訊，請參閱第 571 頁的『MFT 從來源佇列上的訊息讀取的 IBM MQ 訊息內容』及第 154 頁的『監視佇列及使用變數替代』。

如果變數未定義為訊息內容，監視器會報告 BFGDM0060E 錯誤並傳回回覆碼 110 (監視器作業變數替代失敗)。此外，代理程式還會將下列錯誤訊息寫入其事件日誌 (`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.
```

如果針對監視器啟用中等或詳細資源監視器記載，則監視器會將下列訊息寫入代理程式的資源監視器事件日誌 (`resmoneventN.log`):

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

如需資源監視器記載的相關資訊，請參閱第 159 頁的『記載 MFT 資源監視器』。

下表顯示依預設會提供的替代變數。例如，`${AGENTNAME}` 取代為資源監視器代理程式的名稱。

變數	說明
AGENTNAME	資源監視器代理程式的名稱。
QUEUENAME	所監視的佇列名稱。
ENCODING	佇列上第一個訊息或群組中第一個訊息的字元編碼。
MESSAGEID	佇列上的第一個訊息或群組中的第一個訊息的 IBM MQ 訊息 ID。
GROUPID	群組的 IBM MQ 群組 ID，如果只找到一個訊息，則為訊息 ID。只有在您是監視完整群組時，才需要設定此變數。
CurrentTimeStamp	以監視器觸發時的當地時間為基礎的時間戳記。代理程式的時間戳記值是唯一的。
CurrentTimeStamp UTC	以監視器觸發時的世界標準時間時區的時間為基礎的時間戳記。代理程式的時間戳記值是唯一的。

如果受監視資源是目錄

下表顯示作業 XML 定義中可替換的變數名稱集。

變數	說明
FilePath	觸發檔案的完整路徑名稱。
FileName	觸發程式的檔案名稱部分。

表 19: 可替換的變數 (繼續)

變數	說明
LastModifiedTime	觸發檔案前次修改時間。此時間以代理程式執行所在之時區的當地時間表示，並且格式化為 ISO 8601 時間。
LastModifiedDate	觸發檔案前次修改日期。此日期以代理程式執行所在之時區的當地日期表示，並且格式化為 ISO 8601 日期。
LastModifiedTimeUTC	觸發檔案前次修改時間。此時間以轉換為世界標準時間時區的當地時間表示，並且格式化為 ISO 8601 時間。
LastModifiedDateUTC	觸發檔案前次修改日期。此日期以轉換為世界標準時間時區的當地日期表示，並且格式化為 ISO 8601 日期。
AgentName	資源監視器代理程式的名稱。
CurrentTimeStamp	以監視器觸發時的當地時間為基礎的時間戳記。代理程式的時間戳記值是唯一的。
CurrentTimeStampUTC	以監視器觸發時的世界標準時間時區的時間為基礎的時間戳記。代理程式的時間戳記值是唯一的。

如果受監視資源是觸發檔案

下表顯示當資源監視器使用觸發檔案的內容來決定需要傳送的檔案時，可以替代的變數名稱集。

表 20: 使用觸發檔案時可以替代的變數

變數	說明
contentSource	原始檔的完整路徑名稱。
contentDestination	目的地檔案的完整路徑名稱。

變數名稱必須以錢幣符號 (\$) 字元開頭，並以大括弧 {} 括住。例如，`${FilePath}` 會取代為相符觸發檔案的完整檔案路徑。

有兩個特殊關鍵字可套用至變數名稱，以進一步精準化。它們是：

記號

要替換的記號索引（左邊從 1 開始，右邊從 -1 開始）

separator

將變數值記號化的單一字元。預設值是 UNIX 平台上的正斜線字元 (/)，或 Windows 平台上的反斜線字元 (\)，但分隔字元可以是任何可出現在變數值中的有效字元。

如果 separator 關鍵字指定在變數名稱中，變數名稱會根據分隔字元分割成多個記號。

指派給 token 關鍵字的值則用作索引，選取要用來取代變數名稱的記號。記號索引相對於變數中的第一個字元，從 1 開始。如果未指定記號關鍵字，則會插入整個變數。

在訊息 XML 中，代入代理程式名稱的任何值是以不區分大小寫的方式來處理。所有 Managed File Transfer Agent 名稱都是大寫。如果在訊息 XML 中，將 Paris 值代入代理程式屬性，此值會解譯為對代理程式 PARIS 的參照。

相關概念

[第 152 頁的『範例: 資源監視器定義的變數替代』](#)

使用 XML 和 IBM MQ Explorer 來替代資源監視器定義的變數範例。

相關參考

[第 277 頁的『變數替代導致多個檔案傳送至單一檔名時應該如何處理』](#)

對於 Managed File Transfer，如果您正在監視目錄並將多個檔案從來源傳送至目的地位置，而且正在使用 `${FileName}` 變數替代，則必須測試變數替代結果。需要測試結果是因為使用變數替代可能會導致呼叫非預期的檔案傳送指令組合。

範例: 資源監視器定義的變數替代

使用 XML 和 IBM MQ Explorer 來替代資源監視器定義的變數範例。

顯示變數替代如何運作的範例

假設相符觸發程式檔案的檔案路徑在 Windows 上是 `c:\MONITOR\REPORTS\Paris\Report2009.doc`，在 UNIX 平台上是 `/MONITOR/REPORTS/Paris/Report2009.doc`，則會替換變數，如下表所示。

變數規格	在變數替代之後
<code>\${FilePath}</code>	Windows : <code>c:\MONITOR\REPORTS\Paris\Report2009.doc</code> UNIX : <code>/MONITOR/REPORTS/Paris/Report2009.doc</code>
<code>\${FilePath{token=1}{separator=.}}</code>	Windows : <code>c:\MONITOR\REPORTS\Paris\Report2009</code> UNIX : <code>/MONITOR/REPORTS/Paris/Report2009</code>
<code>\${FilePath{token=2}{separator=.}}</code>	Windows : <code>doc</code> UNIX : <code>doc</code>
<code>\${FilePath{token=3}}</code>	Windows : <code>REPORTS</code> UNIX : <code>巴黎</code>

您也可以指定負數記號索引，來選取相對於變數最後字元的記號（如下表中所示）。表格中的範例使用相同的變數值: `c:\MONITOR\REPORTS\Paris\Report2009.doc` (在 Windows 上) 和 `/MONITOR/REPORTS/Paris/Report2009.doc` (在 UNIX 平台上)。

變數規格	在變數替代之後
<code>\${FilePath}</code>	Windows : <code>c:\MONITOR\REPORTS\Paris\Report2009.doc</code> UNIX : <code>/MONITOR/REPORTS/Paris/Report2009.doc</code>
<code>\${FilePath{token=-2}{separator=.}}</code>	Windows : <code>c:\MONITOR\REPORTS\Paris\Report2009</code> UNIX : <code>/MONITOR/REPORTS/Paris/Report2009</code>
<code>\${FilePath{token=-2}{separator=\\}}</code>	Windows : <code>巴黎</code> UNIX : <code>巴黎</code>
<code>\${FilePath{token=-4}}</code>	Windows : <code>MONITOR</code> UNIX : <code>MONITOR</code>

用於替代的變數僅適用於下列正觸發條件及 noSizeChange 選項，這是正觸發條件規則的例外情況：

- match
- fileSize
- noSize 變更

如果使用 noMatch 條件，且作業定義中有替代變數名稱，則不會呼叫作業，且監視器會產生回覆碼 110 及錯誤訊息 BFGDM0060E。

使用 XML 的範例

下列範例作業定義 XML 使用監視器代理程式名稱作為傳送的來源代理程式 (Paris)，使用檔案路徑中倒數第二個目錄名稱作為傳送的目的地代理程式名稱 (Report2009)，並將傳送的檔案重新命名為觸發檔案名稱的根再加上副檔名 .rpt。

```
<?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>
```

這會導致作業 XML 轉換成：

```
<?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> 元素之 agent 屬性中的變數 {FilePath}{token=-2} 會取代為 Paris 一值。此值會以不區分大小寫的方式來處理，並解譯為對代理程式 PARIS 的參照。

使用 IBM MQ Explorer 的範例

透過「IBM MQ Explorer」建立資源監視器時，一旦指定監視器內容及觸發條件，即可將傳送項目新增至監視器。下列範例示範如何在「新增傳送項目畫面」中使用 {FilePath} 和 {FileName} 變數，來自訂資源監視器相符項所產生的傳送。

範例 1

若要在符合觸發條件時將來源檔案簡單地傳送至另一個位置，則可使用 `{FilePath}` 變數：

- 將來源檔名設定為 `{FilePath}`。
- 從目的地的類型下拉功能表中選取目錄。
- 將目的地檔名設定為您要將來源檔案傳送至其中的位置，例如，這可以是 `C:\MFT\out\`。

範例 2

若要將來源檔案傳送至另一個位置並變更檔案的副檔名，則可將 `{FileName}` 變數與 `{FilePath}` 變數一起使用：

在下列範例中，假設來源檔案的檔案路徑等於 `C:\MONITOR\REPORTS\Paris\Report2009.doc`：

- 將來源檔名設定為 `{FilePath}`。
- 將目的地檔名設定為您要將來源檔案傳送至其中的位置，後接 `{FileName}{token=1}{separator=.}`，再接檔案的新副檔名。例如，這可以是 `C:\MFT\out\{FileName}{token=1}{separator=.}.rpt`，它將等於包含來源檔案名稱的 `C:\MFT\out\Report2009.rpt`。

範例 3

若要使用來源檔案的部分檔案路徑來決定傳送目的地，則可將 `{FilePath}` 變數與記號及分隔字元規格一起使用。

在下列範例中，假設來源檔案的檔案路徑等於 `C:\MONITOR\REPORTS\Paris\Report2009.doc`。

您可以使用部分來源檔案路徑來決定檔案目的地。使用檔案路徑範例

`C:\MONITOR\REPORTS\Paris\Report2009.doc`，如果要根據來源檔案的位置（即本範例中的 Paris）將檔案傳送至某個資料夾，則可執行下列動作：

- 將來源檔名設定為 `{FilePath}`。
- 將目的地檔名設定為每個位置的資料夾所處目的地，然後附加檔案路徑的目的地部分及檔名。例如，這可以是 `C:\MFT\out\{FilePath}{token=-2}{separator=} \ {FileName}`，它將等於包含來源檔案名稱的 `C:\MFT\out\Paris\Report2009.doc`。

相關概念

第 149 頁的『使用變數替代來自訂 MFT 資源監視器作業』

滿足作用中資源監視器的觸發條件時，會呼叫所定義的作業。除了每次以相同的目的地代理程式或相同的目的地檔案名稱呼叫傳送或指令作業之外，您還可以在執行時期修改作業定義。您可以將變數名稱插入作業定義 XML 中，來達成此目的。當監視器判斷已滿足觸發條件，且作業定義包含變數名稱時，它會以變數值替換變數名稱，然後呼叫該作業。

相關參考

第 277 頁的『變數替代導致多個檔案傳送至單一檔名時應該如何處理』

對於 Managed File Transfer，如果您正在監視目錄並將多個檔案從來源傳送至目的地位置，而且正在使用 `{FileName}` 變數替代，則必須測試變數替代結果。需要測試結果是因為使用變數替代可能會導致呼叫非預期的檔案傳送指令組合。

監視佇列及使用變數替代

您可以使用 `fteCreateMonitor` 指令來監視佇列，並將訊息從受監視佇列傳送至檔案。在第一個要從受監視佇列讀取的訊息中，任何 IBM MQ 訊息內容的值都可代入作業 XML 定義中，並用來定義傳送行為。

關於這項作業

在此範例中，來源代理程式稱為 `AGENT_VENUS`，其連接至 `QM_VENUS`。`AGENT_VENUS` 所監視的佇列稱為 `START_QUEUE`，位於 `QM_VENUS` 上。代理程式每隔 30 分鐘會輪詢佇列一次。

當完整訊息群組寫入佇列時，監視器作業即會將該訊息群組傳送至眾多目的地代理程式之一（全部都連接至佇列管理程式 `QM_MARS`）的檔案上。訊息群組傳送目的地的檔案名稱，由群組中第一個訊息上的 IBM MQ 訊息內容 `usr.fileName` 所定義。訊息群組傳送目的地的代理程式名稱，由群組中第一個訊息上的 IBM

MQ 訊息內容 `usr.toAgent` 所定義。如果未設定 `usr.toAgent` 標頭，目的地代理程式將會使用預設值 `AGENT_MAGENTA`。

當您指定 `useGroups="true"` 時，如果未同時指定 `groupId="{GROUPID}"`，則傳送只會取得佇列上的第一則訊息。比方說，例如您使用變數替代產生 `fileName`，`a.txt` 的內容因此有可能不正確。這是因為 `fileName` 是由監視器所產生，但傳送實際上取用的並不是應產生 `fileName` 檔案的訊息。

程序

1. 建立作業 XML，以定義監視器被觸發後所執行的作業。

```
<?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>
```

取代為 IBM MQ 訊息標頭值的變數，會以**粗體**強調顯示。此作業 XML 會儲存至 `/home/USER1/task.xml` 檔。

2. 建立用來監視佇列 `START_QUEUE` 的資源監視器。

提交下列指令：

```
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. 使用者或程式將訊息群組寫入佇列 `START_QUEUE`。

此群組中的第一個訊息具有下列 IBM MQ 訊息內容集：

```
usr.fileName=larmer
usr.toAgent=AGENT_VIOLET
```

4. 在寫入完整群組時，會觸發監視器。代理程式將 IBM MQ 訊息內容代入作業 XML 中。這會導致作業 XML 轉換成：

```
<?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>

```

結果

即會執行作業 XML 所定義的傳送。AGENT_VENUS 從 START_QUEUE 讀取的完整訊息群組，會寫入 AGENT_VIOLET 執行時所在系統上稱為 /reports/larmer.rpt 的檔案。

下一步

將每一個訊息傳送至個別檔案

如果要監視佇列並讓每則訊息傳送至個別檔案，您可以使用與本主題中先前所述方式相似的技术。

1. 依照先前說明來建立監視器，並在 **fteCreateMonitor** 指令上指定 **-tr completeGroups** 參數。
2. 在作業 XML 中指定下列內容：

```
<queue useGroups="true" groupId="${GROUPID}">START_QUEUE</queue>
```

不過，當您將訊息放入來源佇列時，請不要將它們放置在 IBM MQ 群組中。為每一個訊息新增 IBM MQ 訊息內容。例如，您可以為每一個訊息指定具有唯一檔案名稱值的 **usr.filename** 內容。這實際上會讓 Managed File Transfer Agent 將來源佇列上的每一則訊息都視為個別群組。

「訊息轉為檔案」傳送的監視器重試行為

如果資源監視器所觸發的「訊息轉為檔案」傳送失敗，而將觸發監視器的訊息群組保留在佇列上，該傳送將會根據後續的輪詢間隔進行重新提交。重新提交傳送的次數受限於監視代理程式的 **monitorGroupRetryLimit** 內容。

「訊息轉為檔案」傳送的已觸發次數，由群組中第一個訊息的 MQMD 取消計數來決定。

每次觸發新的「訊息轉為檔案」傳送時，都會為傳送作業產生新的傳送 ID。

如果代理程式重新啟動，則即使傳送的已觸發次數超出 **monitorGroupRetryLimit** 的值，監視器仍會再次觸發傳送。如果此傳送嘗試導致傳送的已觸發次數超出 **monitorGroupRetryLimit** 的值，則代理程式會將錯誤寫入其事件日誌中。

單一訊息會被視為單一群組，且只要訊息仍在佇列上，或是傳送的已觸發次數小於 **monitorGroupRetryLimit** 的值，就會在每一個輪詢間隔重新觸發傳送。

設定 monitorGroupRetryLimit 內容

monitorGroupRetryLimit 內容的值，是監視器在訊息群組仍存在於佇列上的情況下，再次觸發「訊息轉為檔案」傳送的次數上限。此內容的預設值是 10。此內容值可設為任何正整數值或 -1。如果將此內容值指定為 -1，監視器就會不限次數地重新觸發傳送，直到不符合觸發條件為止。

若要設定監視代理程式的 **monitorGroupRetryLimit** 內容，請執行下列步驟：

1. 使用 **fteStopAgent** 指令來停止監視代理程式。
2. 編輯監視代理程式 **agent.properties** 檔，以併入 **monitorGroupRetryLimit=number_of_retries** 這一行。**agent.properties** 檔案位於 **MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/monitoring_agent_name** 目錄中。
3. 使用 **fteStartAgent** 指令來啟動監視代理程式。

相關工作

第 149 頁的『範例：配置 MFT 資源』

您可以搭配使用 **-mq** 參數與 **fteCreateMonitor** 指令，將 IBM MQ 佇列指定為要由資源監視器監視的資源。

相關參考

第 463 頁的『MFT agent.properties 檔案』

每一個 Managed File Transfer Agent 都有自己的內容檔 `agent.properties`，其中必須包含代理程式用來連接至其佇列管理程式的資訊。`agent.properties` 檔案也可以包含變更代理程式行為的內容。

使用觸發檔

您可以在資源監視器中使用觸發檔案的內容，定義要在單一傳送要求中傳送的檔案集。每次偵測到相符的觸發檔案時，系統都會剖析其內容，以取得來源檔案路徑，並選擇性地取得目的地檔案路徑。然後，這些檔案路徑會用來定義您指定的作業傳送 XML 檔（以單一傳送要求形式提交至代理程式）中的檔案項目。資源監視器的定義會決定是否啟用觸發內容。

您可以在建立監視器時指定 **-tc**（觸發內容）參數，以啟用檔案內容觸發。此 **-tc** 參數僅適用於檔案觸發選項 `match` 及 `noSizeChange`。如需建立監視器的相關資訊，請參閱第 348 頁的『[fteCreateMonitor: 建立 MFT 資源監視器](#)』。

使用觸發程式內容檔時，每一行的預設格式為：

- 單一原始檔路徑，或
- 來源檔案路徑和目的地檔案路徑，以逗點區隔

其中空格字元會作為檔案路徑的一部分來處理。您可以在 **fteCreateMonitor** 指令上指定 **-tcr** 和 **-tcc** 參數，來變更預設行格式。如需相關資訊，請參閱第 158 頁的『[進階選項](#)』。

剖析觸發檔案後，即會產生檔案路徑清單，並將其套用至您所指定的傳送作業 XML。如同所有監視器，傳送作業 XML 的格式是以 **fteCreateTransfer** 指令產生、並定義了單一項目或檔案的完整傳送作業 XML。單一項目必須使用替代變數 `${contentSource}`（並選擇性地使用 `${contentDestination}`）來取代來源及目的地檔案路徑。監視器會展開傳送作業 XML，以在觸發檔案中併入每一行（檔案路徑）的檔案項目。

-bs 參數無法用在檔案內容觸發上，因為 **-tc** 參數表示每一個觸發檔案各有一個傳送要求。

範例

下列範例所定義的監視器會觸發以 `trig` 結尾的檔案，並讀取該檔案中的檔案路徑。

```
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 指令會針對來源檔案路徑為 `${sourceContent}` 的單一檔案，建立名為 `task.xml` 的檔案。例如：

```
<item checksumMethod="MD5" mode="binary">
  <source disposition="leave" recursive="false">
    <file>${contentSource}</file>
  </source>
</item>
```

fteCreateMonitor 指令會在 `/home/trigdir` 目錄中掃描以 `trig` 結尾的檔案，並使用這些內容，為該觸發檔案中的所有路徑建立以 `task.xml` 為基礎的單一傳送要求。觸發檔案的格式必須是一行顯示一個檔案路徑（僅限來源）、且不使用逗點分隔字元。例如：

```
/home/file/first.txt
/home/file/second.txt
/home/different/third.txt
⋮
```

所有檔案都會遞送至具有其檔名（而非其檔案路徑）的 `/file/destdir` 目錄中，亦即，`/home/file/first.txt` 會遞送至 `/file/destdir/first.txt`。

或者，如果您將 **fteCreateTransfer** 指令中的 **-dd /file/destdir** 參數變更為 **-df \$ {contentDestination}**，並將觸發檔案的內容格式變更為 *source file path,destination file path*，則可以為相同目的地代理程式定義不同的目的地路徑。 例如：

```
/home/file/first.txt,/home/other/sixth.txt
```

然後，目的地位置會變成 `/home/other/sixth.txt`。

替代變數可以記號化。例如，您可以使用 `${contentDestination{token=-1}}` 將檔名部分與提供的路徑隔開。因此，如果 **fteCreateTransfer** 目的地定義為 `-df /file/destdir/${contentDestination{token=-1}}`，則 `/home/file/first.txt` 的新目的地會是 `/file/destdir/sixth.txt`。

進階選項

您可以使用 **-tcr** *regex* 參數來變更觸發檔案內容的預設行格式。請提供符合必要行格式、並可提供一個或是兩個擷取群組的正規表示式。第一個擷取群組是來源，第二個擷取群組（選用）是目的地。例如：

- 來源與目的地路徑會以連字號區隔：

```
((?:[^\-])+)-((?:[^\-])+)
```

在此範例中，分隔字元定義於三個位置，而連字號 `-` 的三個實例全都可變更為任何其他字元。請確實跳出任何特殊字元。

- 來源與目的地路徑會以含有尾端空格的逗點區隔。以 `#` 記號表示的註解會被忽略。

```
((?:[^\,])+),((?:[^\,])+)*(?:#\.*)+
```

檔案路徑不能包含 `#` 記號。一般而言，項目如下：`/home/source/from.txt,/home/destination/to.txt # some comment`。

如果您使用 **-tcr** 參數，請確定正規表示式已經過妥善設計及測試，因此表示式能夠偵測出錯誤並正確剖析觸發檔案。

您可以使用 **-tcc destSrc** 參數將擷取順序反轉。如果您指定此參數，則第一個擷取群組會是目的地檔案路徑，第二個群組會是來源檔案路徑。

錯誤的處理方式

空的觸發檔案

如果觸發檔案是空的，則不會執行檔案傳送。亦即，監視器會建立傳送要求，但不會指定檔案項目。

觸發檔案出現錯誤

如果觸發檔案中有任何項目無法根據預期的格式進行剖析，則不會產生傳送要求。此時會發佈監視器錯誤日誌，同時將錯誤記載到事件日誌中。觸發檔案會標示為已處理，且監視器在檔案更新之前都不會再嘗試處理檔案。

傳送作業 XML 不符

傳送作業 XML 必須符合觸發檔案，亦即，如果傳送作業 XML 同時具有 `${sourceContent}` 及 `${destinationContent}`，則該監視器的所有觸發檔案都必須具有來源與目的地檔案路徑，反之亦然。在第一個案例中，如果觸發檔案僅提供來源檔案路徑，則監視器會報告 `${destinationContent}` 替代失敗。

範例

下列範例說明觸發檔案內容僅含有來源檔案路徑的基本內容觸發：

```
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 參數會定義兩個由任意字元序列組成的擷取群組，這些字元以空格字元區隔。**-tcc destSrc** 參數及選項會指出要以目的地在前、來源在後的格式處理擷取群組。

```
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 記載 MFT 資源監視器

從 IBM MQ 9.0.3 開始，Managed File Transfer 包含一種改良的方法，可用於取得關於資源監視器的診斷資訊。

關於這項作業

從 IBM MQ 9.0.3 開始，可以將記載功能用於資源監視器。您可以使用 **fteSetAgentLogLevel** 指令或 **agent.properties** 檔案來控制資源監視器記載。

請注意，現有追蹤點仍將用於擷取資訊。

資源監視器日誌會寫入名為 **resmoneventN.log** 的檔案，其中 *N* 代表數字；例如 **resmonevent0.log**。



小心：一個代理程式的所有資源監視器都會寫入相同的日誌檔。

如需 **resmoneventN.log** 檔案的範例輸出，請參閱第 265 頁的『MFT 目錄資源監視器未觸發檔案時應該如何處理』。

下表列出了資源監視器寫入日誌檔的事件類型。第三個直欄說明擷取每一個事件所需的記載層次，其中最低層次是 **INFO**，而最高層次是 **VERBOSE**。

請注意，設定較高的記載層次，也會寫入較低層次的事件。例如，將記載層次設為 **MODERATE** 也會寫入 **INFO** 層次事件，但不會寫入 **VERBOSE** 層次事件。

號碼	事件	記載層次	說明
1	建立監視器	資訊	已建立資源監視器。
2	刪除監視器	資訊	已刪除資源監視器。
3	停止監視器	資訊	已停止資源監視器。
4	啟動監視器	資訊	已啟動資源監視器。
5	監視器開始輪詢	資訊	資源監視器已開始新的輪詢週期。
6	監視器結束輪詢	資訊	資源監視器輪詢週期已結束。
7	型樣相符	VERBOSE	在觸發監視器目錄上找到與所指定型樣相符的檔案，或者在佇列中找到與所指定型樣相符的訊息。
8	型樣不符	VERBOSE	在觸發監視器目錄上找到不相符的檔案，或者在佇列中找到與所指定型樣不相符的訊息。
9	提交傳送	資訊	資源監視器已起始傳送。
10	目錄太深	VERBOSE	資源監視器所監視的目錄包含更多要輪詢的子目錄，較資源監視器配置中指定的數目大。
11	檔案已鎖定	MODERATE	資源監視器所監視的觸發檔已由其他程序鎖定。
12	檔案太小	MODERATE	觸發檔小於資源監視器配置中指定的大小。

號碼	事件	記載層次	說明
13	檔案大小不穩定	MODERATE	變更觸發檔的頻率超出資源監視器配置的預期。
14	輪詢次數太多	MODERATE	資源監視器輪詢不穩定觸發檔的次數太多。
15	相符的項目數	資訊	在資源監視器所輪詢的目錄中找到的觸發檔總數。
16	傳送中的項目數	資訊	傳送要求中的項目總數。
17	FDC 或異常狀況	MODERATE	資源監視器已產生異常狀況。
18	傳送要求	資訊	由資源監視器提交的傳送要求。
19	啟動監視器失敗	MODERATE	資源監視器啟動失敗。

程序

- 若要使用 `fteSetAgentLogLevel` 來開啟及關閉資源監視器記載，請參閱 [fteSetAgentLogLevel](#) 以取得 `logMonitor` 參數的說明，及不同選項之用法範例。
- 若要使用 `agent.properties` 檔案來控制資源監視器記載，請參閱 第 463 頁的『MFT `agent.properties` 檔案』，以取得容許您執行下列記載活動的其他內容說明：
 - 開啟或關閉記載功能
 - 限制每個日誌檔的大小
 - 限制資源監視器可以產生的日誌數

範例

下列訊息範例會在佇列管理程式 MFTDEMO 上，為代理程式 HA2 設定 `verbose` 層次記載功能：

```
<?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>
```

相關參考

[fteSetAgentLogLevel](#) 指令

使用 `fteSetAgentLogLevel` 指令，針對 Managed File Transfer 通訊協定橋接器代理程式與 FTP/SFTP/FTPS 檔案伺服器之間的檔案傳送，啟用或停用診斷資訊記載。

第 463 頁的『MFT `agent.properties` 檔案』

每一個 Managed File Transfer Agent 都有自己的內容檔 `agent.properties`，其中必須包含代理程式用來連接至其佇列管理程式的資訊。`agent.properties` 檔案也可以包含變更代理程式行為的內容。

備份及還原 MFT 資源監視器

您可以透過下列方式來備份要供未來使用的資源監視器：將其定義匯出至 XML 檔，隨後即可匯入該檔案以從備份建立新的資源監視器。

關於這項作業

您可能需要備份先前已經定義的資源監視器，以便未來可重複使用其定義，例如，在不同基礎架構中或在資源監視器因佇列管理程式問題而需要重建時重建資源監視器。

使用 **fteCreateMonitor** 指令或包含 **-ox** 參數的 **fteListMonitors** 指令，即可備份單一資源管理程式定義。在這兩種情況下，皆可透過將資源管理程式定義匯出至 XML 檔來備份資源管理程式定義。然後即可使用 **fteCreateMonitor** 指令的 **-ix** 參數，從 XML 檔匯入定義以建立新的資源管理程式。

使用 **-ox** 參數時，一次只能備份一個資源監視器定義。

V 9.0.5 從 IBM MQ 9.0.5 開始，**-od** 參數已新增至 **fteListMonitors** 指令。透過指定此參數，即可將多個資源監視器定義成批匯出至指定目錄，以便一次備份多個資源監視器。每一個資源監視器定義都會儲存至個別 XML 檔案，並具有格式為 *agent name.monitor name.xml* 的名稱。

V 9.0.5 如果您具有大量要備份的資源監視器，則 **-od** 參數特別有用，因為您僅需執行 **fteListMonitors -od** 指令一次，而不必針對每個資源定義個別地執行 **fteListMonitors -ox** 指令，亦不必針對每個資源監視器使用個別 Script 來執行 **fteListMonitors -ox** 指令。

程序

- 若要將一個資源監視器的定義匯出至 XML 檔以對其進行備份，請使用下列任一指令：
 - 包含 **-ox** 參數的 **fteCreateMonitor** 指令。
 - 包含 **-ox** 參數的 **fteListMonitors** 指令。

使用 **-ox** 參數時，您還必須指定 **-ma** 及 **-mn** 參數，如下列範例中所示：

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

- V 9.0.5** 若要透過將多個資源監視器定義匯出至指定目錄中的 XML 檔案來備份它們，請搭配使用 **fteListMonitors** 指令與 **-od** 參數，如下列範例所示：

```
fteListMonitors -od /usr/mft/resmonbackup
```

成批備份資源監視器時，您必須指定有效的目標目錄。不指定目標路徑會導致下列範例中所示的錯誤訊息：

BFGCL0762E: Output directory not specified. 請指定有效的路徑來重新執行指令。

-od 參數不得與 **-ox** 參數結合使用，否則會顯示下列錯誤訊息：

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

您可以定義要包含在備份中的一組特定資源監視器。例如，透過使用 **-ma** 參數來指定代理程式名稱，即可備份該代理程式的所有資源監視器，如下列範例中所示：

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

您亦可透過在定義用於比對代理程式名稱及/或監視器名稱的型樣時納入星號字元 (*) 來使用萬用字元比對。下列範例會備份名稱符合指定型樣，且所處代理程式名稱符合指定型樣的所有資源監視器：

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

當此指令正在執行時，它會顯示下列進度報告訊息：

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

如果您使用的是詳細選項，則仍會顯示執行中總計，但並非顯示

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

此指令會顯示正在儲存的監視器定義的名稱，例如：

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

V 9.0.5

- 若要透過將特定代理程式匯出至指定目錄中的 XML 檔案，來備份特定代理程式的一個資源監視器，請使用 **fteListMonitors** 指令並指定 **-od** 參數：

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

除了輸出檔名的格式為 *agent name.monitor name.xml* 以外，使用 **-od** 參數來備份單一資源監視器，類似於使用 **-ox** 參數。

- 若要從備份還原資源監視器定義，請使用包含 **-ix** 參數的 **fteCreateMonitor** 指令，如下列範例中所示：

```
fteCreateMonitor -ix file name
```

如需如何使用 **-od** 參數的其他範例，請參閱第 401 頁的『[fteListMonitors: 列出 MFT 資源監視器](#)』。

相關參考

第 348 頁的『[fteCreateMonitor: 建立 MFT 資源監視器](#)』

fteCreateMonitor 指令可從指令行建立並啟動新的資源監視器。您可以使用 Managed File Transfer 來監視資源（例如目錄的內容），以在符合觸發條件時，啟動指定的作業（例如檔案傳送）。

第 401 頁的『[fteListMonitors: 列出 MFT 資源監視器](#)』

使用 **fteListMonitors** 指令，可讓您透過指令行列出 Managed File Transfer 網路中的所有現有資源監視器。

使用檔案傳送範本

您可以使用檔案傳送範本來儲存一般檔案傳送設定，以進行重複或複式傳送。請使用

fteCreateTemplate 指令，從指令行建立傳送範本，或者使用 IBM MQ Explorer，利用為受管理檔案傳送建立新範本精靈建立傳送範本，或選取將傳送設定儲存為範本勾選框，在建立檔案傳送時儲存範本。傳送範本視窗會顯示您在 Managed File Transfer 網路中建立的所有傳送範本。

關於這項作業

若要從指令行建立傳送範本，請使用 [fteCreateTemplate](#) 指令。然後，當您想要提交您在指令行上建立的傳送範本時，請按一下 IBM MQ Explorer 中的 **提交**。


若要在 IBM MQ Explorer 中檢視傳送範本，請使用下列步驟：

程序

1. 在「導覽器」視圖中展開 **Managed File Transfer**。**Managed File Transfer 中心**即會顯示在「內容」視圖中。
2. 所有協調佇列管理程式都會列在「導覽器」視圖中。展開您用於排定傳送的協調佇列管理程式的名稱。如果您想要變更所連接的協調佇列管理程式，請在「導覽器」視圖中您想要使用的協調佇列管理程式名稱上按一下滑鼠右鍵，然後再按一下**連接**。
3. 按一下**傳送範本**。「**傳送範本**」視窗即會顯示在「內容」視圖中。
4. 「**傳送範本**」視窗列出關於檔案傳送的下列詳細資料：
 - a) **名稱** 檔案傳送範本的名稱。
 - b) **來源** 用來從來源系統傳送檔案的代理程式名稱。
 - c) **來源檔案** 要在其主機系統上傳送的檔案名稱。
展開傳送範本資訊，以檢視此欄位。
 - d) **目的地** 用來在目的地系統上接收檔案的代理程式名稱。
 - e) **目的地檔案** 檔案傳送至目的地系統之後的檔案名稱。
展開傳送範本資訊，以檢視此欄位。

- f) **排定的開始時間（選取的時區）** 排定檔案傳送開始的時間和日期，以管理者使用的時區表示。若要變更顯示的時區，請按一下**視窗 > 喜好設定 > IBM MQ Explorer > Managed File Transfer**，然後從**時區**：清單中選取替代時區。按一下**確定**。
- g) **觸發事件** 觸發檔案傳送開始的事件類型。類型可以是下列其中一值：**存在、不存在或超出**。

結果

若要重新整理**傳送範本**視窗所顯示的內容，請按一下「內容」視圖工具列上的「重新整理」按鈕。

若要提交傳送範本並開始在範本中定義的傳送，請在範本名稱上按一下滑鼠右鍵，然後按一下**提交**。

若要變更傳送範本，請在範本名稱上按一下滑鼠右鍵，然後按一下**編輯**。原始範本中包括的所有檔案都會列為傳送群組的一部分，即使它們並未併入為原始範本的群組的一部分。如果您想要移除範本中的某個檔案，必須從群組中選取檔案規格，然後按一下**移除已選取的項目**。如果您想要將檔案規格新增至範本中，請使用範本畫面中的欄位，然後按一下**新增至群組**按鈕。完成編輯時，系統會提示您為編輯的範本指定新名稱。

若要從傳送範本中建立檔案傳送，請在範本名稱上按一下滑鼠右鍵，然後按一下**編輯為新建傳送**。

若要建立傳送範本的副本，請在範本名稱上按一下滑鼠右鍵，然後按一下**複製**。複製的傳送範本會使用與原始範本相同的名稱自動儲存，並加上 "(copy)" 字樣。

若要刪除傳送範本，請在範本名稱上按一下滑鼠右鍵，然後按一下**刪除**。

相關工作

第 163 頁的『[使用 IBM MQ Explorer 建立檔案傳送範本](#)』

您可以從「IBM MQ Explorer」或從指令行建立檔案傳送範本。然後，您可以使用該範本，利用範本詳細資料建立新的檔案傳送，或提交範本以啟動檔案傳送。

相關參考

第 354 頁的『[fteCreateTemplate: 建立新的檔案傳送範本](#)』

fteCreateTemplate 指令可建立檔案傳送範本，以備將來使用。唯一必要的參數是 **-tn** (*template_name*) 參數。其他所有參數是選用的，雖然如果指定來源檔案規格，您還必須提供目的地檔案。同樣地，如果指定目的地檔案，您還必須提供來源檔案規格。

第 405 頁的『[fteListTemplates](#)』

使用 **fteListTemplates** 指令，可列出協調佇列管理程式上的可用 Managed File Transfer 傳送範本。

第 395 頁的『[fteDeleteTemplates](#)』

使用 **fteDeleteTemplates** 指令可從協調佇列管理程式刪除現有 Managed File Transfer 範本。

使用 IBM MQ Explorer 建立檔案傳送範本

您可以從「IBM MQ Explorer」或從指令行建立檔案傳送範本。然後，您可以使用該範本，利用範本詳細資料建立新的檔案傳送，或提交範本以啟動檔案傳送。

關於這項作業

若要從指令行建立檔案傳送範本，請使用 [fteCreateTemplate](#) 指令。

若要使用 IBM MQ Explorer 中的 **建立受管理檔案傳送的新範本** 精靈來建立檔案傳送範本，請使用下列步驟：

程序

1. 在「導覽器」視圖中，按一下 **Managed File Transfer**。**Managed File Transfer 中心** 即會顯示在「內容」視圖中。
2. 所有協調佇列管理程式都會顯示在「導覽器」視圖中。展開您用於排定傳送的協調佇列管理程式的名稱。如果您想要變更所連接的協調佇列管理程式，請在「導覽器」視圖中您想要使用的協調佇列管理程式名稱上按一下滑鼠右鍵，然後再按一下**連接**。
3. 啟動建立 **Managed File Transfer** 的新範本精靈，方法是用滑鼠右鍵按一下**傳送範本**，然後按一下**新建範本**。

4. 遵循精靈畫面上的指示操作。每一個畫面都提供有上下文相關說明。若要存取 Windows 上的上下文相關說明，請按 F1 鍵。在 Linux 上，按 Ctrl+F1 或 Shift+F1。

如果您已建立包含所有必要傳送詳細資料的範本，請確保已選取**傳送摘要**頁面上的**將傳送設定儲存為範本**勾選框（如果尚未選取此勾選框）。亦請在「名稱」欄位中輸入範本的名稱。如果您建立的範本尚未包含所有必要的傳送詳細資料，系統會自動勾選**將傳送設定儲存為範本**勾選框。

相關工作

第 162 頁的『使用檔案傳送範本』

您可以使用檔案傳送範本來儲存一般檔案傳送設定，以進行重複或複式傳送。請使用

fteCreateTemplate 指令，從指令行建立傳送範本，或者使用 IBM MQ Explorer，利用**為受管理檔案傳送建立新範本**精靈建立傳送範本，或選取**將傳送設定儲存為範本**勾選框，在建立檔案傳送時儲存範本。傳送範本視窗會顯示您在 Managed File Transfer 網路中建立的所有傳送範本。

相關參考

第 354 頁的『fteCreateTemplate: 建立新的檔案傳送範本』

fteCreateTemplate 指令可建立檔案傳送範本，以備將來使用。唯一必要的參數是 **-tn** (*template_name*) 參數。其他所有參數是選用的，雖然如果指定來源檔案規格，您還必須提供目的地檔案。同樣地，如果指定目的地檔案，您還必須提供來源檔案規格。

第 405 頁的『fteListTemplates』

使用 **fteListTemplates** 指令，可列出協調佇列管理程式上的可用 Managed File Transfer 傳送範本。

第 395 頁的『fteDeleteTemplates』

使用 **fteDeleteTemplates** 指令可從協調佇列管理程式刪除現有 Managed File Transfer 範本。

將資料從檔案傳送至訊息

您可以使用 Managed File Transfer 的「檔案轉為訊息」特性，將資料從一個檔案傳送至 IBM MQ 佇列上的單一訊息或多則訊息。

若要執行「檔案轉為訊息」和「訊息轉為檔案」的傳送，傳送的來源和目的地代理程式都必須為 IBM WebSphere MQ 7.5 或更新版本，或是 IBM WebSphere MQ File Transfer Edition 7.0.3 或更新版本。如需「訊息轉為檔案」傳送的相關資訊，請參閱第 171 頁的『將資料從訊息傳送至檔案』。

檔案到訊息傳送的目的地代理程式，不得為通訊協定橋接器代理程式或 Connect:Direct 橋接器代理程式。

您可以將檔案資料傳送至 IBM MQ 訊息資料。IBM MQ 訊息可由應用程式讀取及使用。支援下列類型的「檔案轉為訊息」傳送：

- 從單一檔案至單一訊息。該訊息未設定 IBM MQ 群組 ID。
- 從單一檔案至多則訊息，方法是將檔案分割為給定長度的訊息。這些訊息都具有相同的 IBM MQ 群組 ID。
- 從單一檔案至多則訊息，方法是使用 Java 正規表示式定界字元來分割文字檔。這些訊息都具有相同的 IBM MQ 群組 ID。
- 從單一檔案至多則訊息，方法是使用十六進位定界字元來分割二進位檔。這些訊息都具有相同的 IBM MQ 群組 ID。

如果您要以位元組序列作為定界字元來分割二進位檔，請使用 **fteCreateTransfer** 指令的 **-sqdb** 參數。如需相關資訊，請參閱 **-sqdb** 參數。

依預設，透過「檔案轉為訊息」傳送所建立的訊息都是持續性訊息。這些訊息可以設定為非持續性訊息，或者設定為使用目的地佇列所定義的持續性值。

如果您指定將一個檔案分割為多則訊息，則從該檔案建立的所有訊息都將具有相同的 IBM MQ 群組 ID。如果您沒有指定將一個檔案分割為多則訊息，則只會從該檔案建立一則訊息，而且該訊息未設定 IBM MQ 群組 ID。

如果您要將檔案傳送至大型訊息，或許多小型訊息，則可能需要變更某些 IBM MQ 或 Managed File Transfer 內容。如需相關資訊，請參閱第 270 頁的『設定與訊息大小相關聯的 MQ 屬性及 MFT 內容的指引』。

註: 如果目的地佇列是叢集佇列或叢集佇列的別名, 而且未將代理程式內容 `enableClusterQueueInputOutput` 設為 `true`, 則在將檔案傳送至佇列時, 您會收到一則錯誤訊息。如需相關資訊, 請參閱第 267 頁的『目的地佇列是叢集佇列或叢集佇列的別名時應該如何處理』

配置代理程式以執行「檔案轉為訊息」傳送

依預設, 代理程式無法執行檔案轉為訊息或「訊息轉為檔案」傳送。若要啟用此功能, 必須將代理程式內容 `enableQueueInputOutput` 設定為 `true`。為了能夠寫入 IBM MQ 叢集佇列, 您還必須將代理程式內容 `enableClusterQueueInputOutput` 設為 `true`。

關於這項作業

如果嘗試對未將 `enableQueueInputOutput` 內容設為 `true` 的目的地代理程式執行檔案到訊息傳送, 則傳送就會失敗。發佈至協調佇列管理程式的傳送日誌訊息包含下列訊息:

```
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.
```

若要讓代理程式能夠對佇列進行寫入及讀取, 請執行下列步驟:

程序

1. 使用 `fteStopAgent` 指令停止目的地代理程式。
2. 編輯 `agent.properties` 檔案, 以包含 `enableQueueInputOutput=true` 行。
`agent.properties` 檔案位於 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/destination_agent_name` 目錄中。
3. 選擇性的: 編輯 `agent.properties` 檔案, 以包含 `enableClusterQueueInputOutput=true` 行。
`agent.properties` 檔案位於 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/destination_agent_name` 目錄中。
4. 使用 `fteStartAgent` 指令啟動目的地代理程式。

範例: 將單一檔案傳送至單一訊息

透過將 `-dq` 參數與 `fteCreateTransfer` 指令搭配使用, 您可以將佇列指定為檔案傳送的目的地。來源檔案必須小於目的地佇列上設定的訊息長度上限。目的地佇列所在的佇列管理程式與目的地佇列所連接的佇列管理程式不必相同, 但這兩個佇列管理程式必須能夠進行通訊。

關於這項作業

來源檔案名為 `/tmp/single_record.txt`, 與來源代理程式 `AGENT_NEPTUNE` 位於同一系統。來源代理程式 `AGENT_NEPTUNE` 使用佇列管理程式 `QM_NEPTUNE`。目的地代理程式為 `AGENT_VENUS`, 並且此代理程式連接至佇列管理程式 `QM_VENUS`。目的地佇列 `RECEIVING_QUEUE` 位於佇列管理程式 `QM_MERCURY` 上。`QM_MERCURY` 與佇列管理程式 `QM_VENUS` 位於相同的 IBM MQ 網路, 而且 `QM_VENUS` 可以存取 `QM_MERCURY`。

程序

請鍵入下列指令:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -dm QM_VENUS  
-dq RECEIVING_QUEUE@QM_MERCURY /tmp/single_record.txt
```

如果目的地佇列所在的佇列管理程式不同於目的地佇列所使用的佇列管理程式, 則必須以下列格式指定 `-dq` 參數的值: `queue_name@queue_manager_name`。如果未在該值中指定 `@queue_manager_name`, 則目的地代理程式會假設目的地佇列是位於目的地代理程式佇列管理程式上。`enableClusterQueueInputOutput` 代

理程式內容設為 true 屬異常狀況。在此情況下，目的地代理程式將使用標準 IBM MQ 解析程序來判定放置佇列的位置。

來源代理程式 AGENT_NEPTUNE 會從 /tmp/single_record.txt 檔案讀取資料，然後將該資料傳送至目的地代理程式 AGENT_VENUS。目的地代理程式 AGENT_VENUS 會將該資料傳送至 RECEIVING_QUEUE@QM_MERCURY 佇列上的持續性訊息。該訊息未設定 IBM MQ 群組 ID。

範例：依長度將單一檔案分割為多則訊息

您可以使用 **fteCreateTransfer** 指令的 **-qs** 參數，將檔案分割成多則 IBM MQ 訊息。檔案會被分割為多個固定長度的區段，每個區段會寫入個別訊息。

關於這項作業

來源檔案名為 /tmp/source.file，大小為 36KB。該來源檔案與來源代理程式 AGENT_NEPTUNE 位於同一系統。來源代理程式 AGENT_NEPTUNE 連接至佇列管理程式 QM_NEPTUNE。目的地代理程式為 AGENT_MERCURY，此代理程式連接至佇列管理程式 QM_MERCURY。目的地佇列 RECEIVING_QUEUE 還位於佇列管理程式 QM_MERCURY 上。傳送會將該來源檔案分割為大小為 1KB 的區段，然後將每個區段寫入 RECEIVING_QUEUE 上的訊息。

程序

請鍵入下列指令：

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY  
-dq RECEIVING_QUEUE -qs 1K /tmp/source.file
```

來源代理程式 AGENT_NEPTUNE 會從 /tmp/source.file 檔案讀取資料，然後將該資料傳送至目的地代理程式 AGENT_MERCURY。目的地代理程式 AGENT_MERCURY 會將資料寫入佇列 RECEIVING_QUEUE@QM_MERCURY 上的 36 個 1KB 持續性訊息。這些訊息都具有相同的 IBM MQ 群組 ID，而且群組中的最後一則訊息已設定 IBM MQ LAST_MSG_IN_GROUP 旗標。

範例：使用正規表示式定界字元將一個文字檔分割為多則訊息

透過在給定的 Java 正規表示式的每個相符項之處分割單一文字檔，可以將該檔案傳送至多則訊息。若要這樣做，需使用 **fteCreateTransfer** 指令的 **-dqdt** 參數。

關於這項作業

檔案會被分割為多個可變長度的區段，每個區段會寫入個別訊息。對於文字檔，會在檔案中文字符合給定正規表示式的每個點來分割檔案。來源檔案名為 /tmp/names.text，並且具有下列內容：

```
Jenny Jones,John Smith,Jane Brown
```

指定該檔案分割點的正規表示式為逗點字元 (,)。

該來源檔案與連接至佇列管理程式 QM_NEPTUNE 的來源代理程式 AGENT_NEPTUNE 位於同一系統。目的地佇列 RECEIVING_QUEUE 位於佇列管理程式 QM_MERCURY 上。QM_MERCURY 也是目的地代理程式 AGENT_MERCURY 所使用的佇列管理程式。傳送會將該來源檔案分割為多個區段，並將每個區段寫入 RECEIVING_QUEUE 上的訊息。

程序

請鍵入下列指令：

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY  
-dq RECEIVING_QUEUE -t text -dqdp postfix -dqdt "," /tmp/names.text
```

來源代理程式 AGENT_NEPTUNE 會從 /tmp/names.text 檔案讀取資料，然後將該資料傳送至目的地代理程式 AGENT_MERCURY。目的地代理程式 AGENT_MERCURY 會將資料寫入 *RECEIVING_QUEUE* 佇列上的三個持續訊息。這些訊息都具有相同的 IBM MQ 群組 ID，而且群組中的最後一則訊息已設定 IBM MQ *LAST_MSG_IN_GROUP* 旗標。

訊息中的資料如下所示。

- 第一則訊息：

```
Jenny Jones
```

- 第二則訊息：

```
John Smith
```

- 第三則訊息：

```
Jane Brown
```

範例：使用正規表示式定界字元分割文字檔，並在訊息中包含定界字元

透過在給定的 Java 正規表示式的每個相符項之處分割單一文字檔，可以將該檔案傳送至多則訊息，並在產生的訊息中包含正規表示式相符項。若要這樣做，需使用 `fteCreateTransfer` 指令的 `-dqdt` 及 `-qi` 參數。

關於這項作業

將單一文字檔傳送至佇列上的多則訊息。檔案會被分割為多個可變長度的區段，每個區段會寫入個別訊息。對於文字檔，會在檔案中文字符合給定正規表示式的每個點來分割檔案。來源檔案名為 /tmp/customers.text，並且具有下列內容：

```
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
```

指定該檔案分割點的正規表示式為 `Customer\snumber:\s\d+`，相符項為 "Customer number:" 後面接任何位數的文字。在指令行上指定的正規表示式必須以雙引號括住，以避免指令 Shell 評估正規表示式。該正規表示式是當成 Java 正規表示式進行評估。如需相關資訊，請參閱第 554 頁的『MFT 所使用的正規表示式』。

依預設，正規表示式可以比對的字元數為五個。此範例中所使用的正規表示式會比對大於五個字元的字串。若要啟用大於五個字元的比對，請編輯代理程式內容檔，以包含 `maxDelimiterMatchLength` 內容。

依預設，訊息中不會包含符合正規表示式的文字。若要在訊息中包含符合正規表示式的文字（如此範例中所示），請使用 `-qi` 參數。來源檔案與連接至佇列管理程式 QM_NEPTUNE 的來源代理程式 AGENT_NEPTUNE 位於同一系統。目的地佇列 *RECEIVING_QUEUE* 位於佇列管理程式 QM_MERCURY 上。QM_MERCURY 也是目的地代理程式 AGENT_MERCURY 所使用的佇列管理程式。傳送會將該來源檔案分割為多個區段，並將每個區段寫入 *RECEIVING_QUEUE* 上的訊息。

程序

1. 使用以下指令停止目的地代理程式：

```
fteStopAgent AGENT_MERCURY
```

2. 將以下行新增至 AGENT_MERCURY 的代理程式內容檔：

```
maxDelimiterMatchLength=25
```

註：增加 **maxDelimiterMatchLength** 值可能會降低效能。

3. 使用以下指令啟動目的地代理程式：

```
fteStartAgent AGENT_MERCURY
```

4. 請鍵入下列指令：

```
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
```

來源代理程式 AGENT_NEPTUNE 會從 /tmp/customers.text 檔案讀取資料，然後將該資料傳送至目的地代理程式 AGENT_MERCURY。目的地代理程式 AGENT_MERCURY 會將資料寫入 RECEIVING_QUEUE 佇列上的三個持續性訊息。這些訊息都具有相同的 IBM MQ 群組 ID，而且群組中的最後一則訊息已設定 IBM MQ LAST_MSG_IN_GROUP 旗標。

訊息中的資料如下所示。

- 第一則訊息：

```
Customer name: John Smith  
Customer contact details: john@example.net  
Customer number: 314
```

- 第二則訊息：

```
Customer name: Jane Brown  
Customer contact details: jane@example.com  
Customer number: 42
```

- 第三則訊息：

```
Customer name: James Jones  
Customer contact details: jjones@example.net  
Customer number: 26
```

範例：設定有關「檔案轉為訊息」傳送的 IBM MQ 訊息內容

您可以在 **fteCreateTransfer** 指令上使用 **-qmp** 參數，以指定是否在傳送寫入目的地佇列的第一則訊息上設定 IBM MQ 訊息內容。IBM MQ 訊息內容可讓應用程式選取要處理的訊息，或在不存取 IBM MQ 訊息描述子 (MQMD) 或 MQRFH2 標頭的情況下擷取訊息的相關資訊。

關於這項作業

在 **fteCreateTransfer** 指令中包含 **-qmp true** 參數。在此範例中，提交該指令的使用者的 MQMD 使用者 ID 為 larmer。

程序

請鍵入下列指令：

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq MY_QUEUE@MyQM -qmp true  
-t text /tmp/source_file.txt
```


由目的地代理程式 AGENT_SATURN 寫入佇列 MY_QUEUE(位於佇列管理程式 MyQM 上) 的第一則訊息的 IBM MQ 訊息內容設為下列值:

```
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.WMQFTECmdUser=larmer
```

範例：設定有關「檔案轉為訊息」傳送的使用者定義內容

使用者定義 meta 資料設定為由傳送寫入目的地佇列的第一則訊息的 IBM MQ 訊息內容。IBM MQ 訊息內容可讓應用程式選取要處理的訊息，或在未存取 IBM MQ 訊息描述子 (MQMD) 或 MQRFH2 標頭的情況下擷取訊息的相關資訊。

關於這項作業

在 `fteCreateTransfer` 指令中包含參數 `-qmp true` 及 `-md account=123456`，以將 RFH2 標頭中的 `usr.account` 內容設為 123456。

程序

請鍵入下列指令：

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq MY_QUEUE@MyQM
-qmp true -md account=123456 /tmp/source_file.txt
```

除了 IBM MQ 訊息內容標準集外，由目的地代理程式 AGENT_SATURN 寫入佇列 MY_QUEUE（位於佇列管理程式 MyQM 上）的第一則訊息的訊息標頭中還會設定使用者定義內容。標頭設定為下列值：

```
usr.account=123456
```

字首 `usr` 會新增至使用者定義 meta 資料名稱的開頭。

範例：新增「檔案轉為訊息」傳送的使用者定義訊息內容

將 Managed File Transfer 用於訊息轉為檔案的受管理傳送時，您可以針對產生的訊息併入使用者定義的訊息內容。

關於這項作業

您可以使用下列任何方法來定義自訂訊息內容：

- 指定傳送要求的 `-md` 參數。如需相關資訊，請參閱 [第 169 頁的『範例：設定有關「檔案轉為訊息」傳送的使用者定義內容』](#)。
- 使用 Ant 作業；您可以使用 `fte:filecopy` 或 `fte:filemove`。下列範例說明 `fte:filecopy` 作業：

```
<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>

```

- 使用資源監視器及變數替代。下列範例顯示部分傳送作業 XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<monitor:monitor
xmlns:monitor="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="5.00"
xsi:schemaLocation="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition ./Monitor.xsd">
  <name>METADATA</name>
  <pollInterval units="minutes">5</pollInterval>
  <batch maxSize="5"/>
  <agent>AGENT1</agent>
  <resources>
    <directory recursionLevel="0">e:\temp</directory>
  </resources>
  <triggerMatch>
    <conditions>
      <allOf>
        <condition>
          <fileMatch>
            <pattern>*.txt</pattern>
          </fileMatch>
        </condition>
      </allOf>
    </conditions>
  </triggerMatch>
  <tasks>
    <task>
      <name/>
      <transfer>
        <request version="5.00"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
          <managedTransfer>
            <originator>
              <hostName>mqjason.raleigh.ibm.com.</hostName>
              <userID>administrator</userID>
            </originator>
            <sourceAgent QMgr="AGENTQM" agent="AGENT1"/>
            <destinationAgent QMgr="AGENTQM" agent="AGENT2"/>
            <transferSet priority="0">
              <metaDataSet>
                <metaData key="FileName">{FileName}</metaData>
              </metaDataSet>
              <item checksumMethod="MD5" mode="text">
                <source disposition="delete" recursive="false">
                  <file>{FilePath}</file>
                </source>
                <destination type="queue">
                  <queue persistent="true"
setMqProps="true">TEST.QUEUE@AGENTQM</queue>
                </destination>
              </item>
            </transferSet>
          </managedTransfer>
        </request>
      </transfer>
    </task>
  </tasks>
</originator>

```

```
<hostName>mqjason.raleigh.ibm.com.</hostName>
<userID>administrator</userID>
</originator>
</monitor:monitor>
```

相關工作

第 168 頁的『[範例：設定有關「檔案轉為訊息」傳送的 IBM MQ 訊息內容](#)』

您可以在 **fteCreateTransfer** 指令上使用 **-qmp** 參數，以指定是否在傳送寫入目的地佇列的第一則訊息上設定 IBM MQ 訊息內容。IBM MQ 訊息內容可讓應用程式選取要處理的訊息，或在不存取 IBM MQ 訊息描述子 (MQMD) 或 MQRFH2 標頭的情況下擷取訊息的相關資訊。

相關參考

第 735 頁的『[fte:filecopy Ant 作業](#)』

fte:filecopy 作業可在 Managed File Transfer 代理程式之間複製檔案。不會從來源代理程式刪除檔案。

第 739 頁的『[fte:filemove Ant 作業](#)』

fte:filemove 作業可在 Managed File Transfer 代理程式之間移動檔案。將檔案從來源代理程式成功傳送至目的地代理程式後，會從來源代理程式刪除檔案。

檔案轉為訊息傳送失敗

在代理程式已啟動將檔案資料寫入目的地佇列作業後，如果「檔案轉為訊息」傳送失敗，代理程式會將一則訊息寫入佇列，以指示使用訊息的應用程式發生失敗。

如果發生失敗，已寫入目的地佇列的訊息將：

- 為空白
- 具有與代理程式先前寫入目的地佇列的訊息相同的 IBM MQ 群組 ID
- 已設定 IBM MQ LAST_MSG_IN_GROUP 旗標
- 包含其他 IBM MQ 訊息內容（如果已啟用訊息內容）。如需相關資訊，請參閱 [第 570 頁的『失敗內容』](#) 主題。

範例

透過執行以下指令來要求傳送：

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq RECEIVING_QUEUE
-qmp true -qs 1K /tmp/source1.txt
```

source1.txt 檔案為 48KB。傳送會將此檔案分割為 1KB 的訊息，並將這些訊息寫入目的地佇列 RECEIVING_QUEUE。

在傳送進行時，代理程式將 16 則訊息寫入 RECEIVING_QUEUE 後，來源代理程式發生失敗。

代理程式會將空白訊息寫入 RECEIVING_QUEUE。除了訊息內容標準集外，空白訊息還具有以下訊息內容集：

```
usr.WMQFTEResultCode = 40
usr.WMQFTESupplement = BFGTR0036I: The transfer failed to complete successfully.
```

將資料從訊息傳送至檔案

Managed File Transfer 的「訊息轉為檔案」特性，可讓您從 IBM MQ 佇列的一個以上訊息中，將資料傳送至檔案、資料集（在 z/OS 上）或使用者檔案空間。如果您具有可建立或處理 IBM MQ 訊息的應用程式，則可以使用 Managed File Transfer 的「訊息轉為檔案」功能，將這些訊息傳送至您 Managed File Transfer 網路中任何系統上的檔案。

如需「檔案轉為訊息」傳送的相關資訊，請參閱 [第 164 頁的『將資料從檔案傳送至訊息』](#)。



小心：「訊息轉為檔案」傳送的來源代理程式不可以是通訊協定橋接器代理程式或 Connect:Direct 橋接器代理程式。

您可以將 IBM MQ 訊息資料傳送至檔案。下列是受支援的「訊息轉為檔案」傳送類型：

- 從單一訊息至單一檔案
- 從多個訊息至單一檔案
- 從具有相同 IBM MQ 群組 ID 的多個訊息至單一檔案。
- 從多個訊息至單一檔案，包括寫入至檔案之每一個訊息中的資料之間的文字或二進位定界字元。

如果您要從大型訊息或許多小型訊息傳送檔案，則可能必須變更某些 IBM MQ 或 Managed File Transfer 內容。如需相關資訊，請參閱第 270 頁的『設定與訊息大小相關聯的 MQ 屬性及 MFT 內容的指引』。

V 9.0.1 從 IBM MQ 9.0.1，在「訊息轉為檔案」傳送中，來源代理程式會瀏覽來自來源佇列的訊息，與 IBM MQ 9.0.1 之前版本的破壞性 GET 不同。瀏覽完所有訊息（如果使用了訊息分組，則為群組中）並將資料寫入目的地檔案之後，就會從來源佇列中移除訊息。這可在萬一傳送失敗或取消時，使訊息仍保留在來源佇列中。由於這項變更，因此必須隨 GET 權限一起提供 BROWSE 權限，才能執行「訊息轉為檔案」的傳送。

V 9.0.0.2 從 IBM MQ 9.0.0 Fix Pack 2 開始，將會更新 Managed File Transfer 以還原先透過 APAR IT18213 移除，並且針對傳送要求 XML 內容中的傳送 ID 及 groupId 屬性值所進行的比較檢查。如果這兩個 ID 相同，則在「訊息轉為檔案」傳送的輸入佇列中第一次進行 MQGET 嘗試時，來源代理程式會使用此 ID 作為訊息 ID 比對選項（相對於群組 ID 比對選項）。

配置代理程式以執行訊息轉為檔案傳送

依預設，代理程式無法執行訊息與檔案之間的來回傳送。若要啟用此功能，必須將代理程式內容 enableQueueInputOutput 設為 true。

關於這項作業

如果您嘗試從 enableQueueInputOutput 內容未設為 true 的來源代理程式中執行「訊息轉為檔案」的傳送，傳送將會失敗。發佈至協調佇列管理程式的傳送日誌訊息包含下列訊息：

```
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.
```

若要讓代理程式能夠對佇列進行寫入及讀取，請執行下列步驟：

程序

1. 使用 **fteStopAgent** 指令來停止來源代理程式。
2. 編輯 agent.properties 檔案，以包含 enableQueueInputOutput=true 行。
agent.properties 檔案位於 MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/source_agent_name 目錄中。
3. 使用 **fteStartAgent** 指令來啟動來源代理程式。

範例：從佇列傳送至單一檔案

您可以搭配使用 **-sq** 參數與 **fteCreateTransfer** 指令，將 IBM MQ 佇列指定為檔案傳送的來源。

關於這項作業

來源資料內含在佇列 START_QUEUE 上的三個訊息中。此佇列必須位於來源代理程式的佇列管理程式 QM_NEPTUNE 上。

程序

請鍵入下列指令：

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

佇列 `START_QUEUE` 上的訊息中資料，會被寫入 `AGENT_VENUS` 執行所在系統上的 `/out/three_to_one.txt` 檔。

範例：將訊息群組從佇列傳送至單一檔案

您可以搭配使用 `-sq` 及 `-sqgi` 參數與 `fteCreateTransfer` 指令，在 IBM MQ 佇列上指定單一完整群組作為檔案傳送的來源。

關於這項作業

在此範例中，佇列 `START_QUEUE` 上有十個訊息。此佇列必須位於來源代理程式的佇列管理程式 `QM_NEPTUNE` 上。前三個訊息屬於 IBM MQ 群組 ID 為 `41424b3ef3a22020202020202020202020202020202020201111` 的群組；此群組並不是完整群組。接下來的五個訊息屬於 IBM MQ 群組 ID 為 `41424b3ef3a22020202020202020202020202020202020202222` 的群組；此群組是完整群組。其餘的兩個訊息屬於 IBM MQ 群組 ID 為 `41424b3ef3a22020202020202020202020202020202020203333` 的群組；此群組是完整群組。

程序

請鍵入下列指令：

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

訊息中屬於佇列 `START_QUEUE` 上第一個完整群組 (具有 IBM MQ 群組 ID `41424b3ef3a22020202020202020202020202020202020202222` 的群組) 的資料會寫入 `AGENT_VENUS` 執行所在系統上的 `/out/group.txt` 檔案。

範例：在每一個訊息中的資料前面插入文字定界字元

在文字模式下從來源佇列傳送至檔案時，您可以搭配使用 `fteCreateTransfer` 指令與 `-sq`、`-sqdt` 及 `-sqdp` 參數，以指定在個別訊息中的資料前面插入文字定界字元。

關於這項作業

在此範例中，佇列 `START_QUEUE` 上有四個訊息。此佇列位於來源代理程式的佇列管理程式 `QM_NEPTUNE` 上。要在每一則訊息的資料前面插入的文字定界字元，可以用 Java 文字字串來表示，例如：
`\n\u002D\u002D\u002D\n`。

程序

請鍵入下列指令：

```
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
```

來源代理程式 `AGENT_NEPTUNE` 會分別為 `START_QUEUE` 上四個訊息中的資料加上前置的文字定界字元。這項資料會寫入目的地檔案 `/out/output.txt` 中。

範例：在每一個訊息中的資料後面插入二進位定界字元

以二進位模式從來源佇列傳送至檔案時，您可以在 **fteCreateTransfer** 指令中使用 **-sq**、**-sqdb** 及 **-sqdp** 參數，指定在個別訊息中的資料之後插入二進位定界字元。

關於這項作業

在此範例中，佇列 **START_QUEUE** 上有三個訊息。此佇列位於來源代理程式的佇列管理程式 **QM_NEPTUNE** 上。要在每一個訊息中的資料後面插入的二進位定界字元，必須使用以逗點區隔的十六進位位元組清單表示，例如：**x34,xE7,xAE**。

程序

請鍵入下列指令：

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -df /out/binary.file
                  -sqdp postfix -sqdb x34,xE7,xAE -sq START_QUEUE
```

來源代理程式 **AGENT_NEPTUNE** 會分別為 **START_QUEUE** 上三個訊息中的資料加上二進位定界字元。此資料會寫入目的地檔案 **/out/binary.file**。

監視佇列及使用變數替代

您可以使用 **fteCreateMonitor** 指令來監視佇列，並將訊息從受監視佇列傳送至檔案。在第一個要從受監視佇列讀取的訊息中，任何 IBM MQ 訊息內容的值都可代入作業 XML 定義中，並用來定義傳送行為。

關於這項作業

在此範例中，來源代理程式稱為 **AGENT_VENUS**，其連接至 **QM_VENUS**。**AGENT_VENUS** 所監視的佇列稱為 **START_QUEUE**，位於 **QM_VENUS** 上。代理程式每隔 30 分鐘會輪詢佇列一次。

當完整訊息群組寫入佇列時，監視器作業即會將該訊息群組傳送至眾多目的地代理程式之一（全部都連接至佇列管理程式 **QM_MARS**）的檔案上。訊息群組傳送目的地的檔案名稱，由群組中第一個訊息上的 IBM MQ 訊息內容 **usr.fileName** 所定義。訊息群組傳送目的地的代理程式名稱，由群組中第一個訊息上的 IBM MQ 訊息內容 **usr.toAgent** 所定義。如果未設定 **usr.toAgent** 標頭，目的地代理程式將會使用預設值 **AGENT_MAGENTA**。

當您指定 **useGroups="true"** 時，如果未同時指定 **groupId="\${GROUPID}"**，則傳送只會取得佇列上的第一則訊息。比方說，例如您使用變數替代產生 **fileName**，**a.txt** 的內容因此有可能不正確。這是因為 **fileName** 是由監視器所產生，但傳送實際上取用的並不是應產生 **fileName** 檔案的訊息。

程序

1. 建立作業 XML，以定義監視器被觸發後所執行的作業。

```
<?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>
```

```

    </item>
  </transferSet>
</managedTransfer>
</request>

```

取代為 IBM MQ 訊息標頭值的變數，會以**粗體**強調顯示。此作業 XML 會儲存至 /home/USER1/task.xml 檔。

2. 建立用來監視佇列 START_QUEUE 的資源監視器。

提交下列指令：

```

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. 使用者或程式將訊息群組寫入佇列 START_QUEUE。

此群組中的第一個訊息具有下列 IBM MQ 訊息內容集：

```

usr.fileName=larmer
usr.toAgent=AGENT_VIOLET

```

4. 在寫入完整群組時，會觸發監視器。代理程式將 IBM MQ 訊息內容代入作業 XML 中。這會導致作業 XML 轉換成：

```

<?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>

```

結果

即會執行作業 XML 所定義的傳送。AGENT_VENUS 從 START_QUEUE 讀取的完整訊息群組，會寫入 AGENT_VIOLET 執行時所在系統上稱為 /reports/larmer.rpt 的檔案。

下一步

將每一個訊息傳送至個別檔案

如果要監視佇列並讓每則訊息傳送至個別檔案，您可以使用與本主題中先前所述方式相似的技术。

1. 依照先前說明來建立監視器，並在 **fteCreateMonitor** 指令上指定 **-tr completeGroups** 參數。
2. 在作業 XML 中指定下列內容：

```

<queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>

```

不過，當您將訊息放入來源佇列時，請不要將它們放置在 IBM MQ 群組中。為每一個訊息新增 IBM MQ 訊息內容。例如，您可以為每一個訊息指定具有唯一檔案名稱值的 **usr.fileName** 內容。這實際上會讓 Managed File Transfer Agent 將來源佇列上的每一則訊息都視為個別群組。

範例：使用 IBM MQ 訊息內容阻止訊息轉為檔案傳送

您可以將 `usr.UserReturnCode` IBM MQ 訊息內容設定為非零值，讓「訊息轉為檔案」的傳送無法執行。您也可以設定 `usr.UserSupplement` IBM MQ 訊息內容，指定無法執行之原因的增補資訊。

關於這項作業

在此範例中，佇列 `INPUT_QUEUE` 與 `/home/user/output.file` 檔之間正在進行傳送。

使用者會建立訊息，並將其放置於佇列 `INPUT_QUEUE` 上。來源代理程式採用佇列 `INPUT_QUEUE` 中的訊息，並將傳送資料傳送至目的地代理程式。目的地代理程式則將此資料寫入 `/home/user/output.file` 檔中。

將訊息寫入佇列 `INPUT_QUEUE` 的使用者想要停止進行中的傳送，並且刪除任何已寫入目的地檔案的資料。

程序

1. 使用者將訊息寫入具有下列 IBM MQ 訊息內容集的佇列 `INPUT_QUEUE`:

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

2. 來源代理程式讀取這些 IBM MQ 訊息內容，並停止處理來自該佇列的訊息。目的地代理程式刪除任何已寫入目的地目錄的檔案資料。
3. 來源代理程式將傳送日誌訊息傳送至協調佇列管理程式，回報傳送失敗。
訊息包含下列資訊：

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
  ID="414d5120514d3120202020202020202020202020207e970d4920008702" 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 代理程式

您可以使用指令行或 IBM MQ Explorer 來列出向特定佇列管理程式登錄的 Managed File Transfer 代理程式。

關於這項作業

若要使用指令行列出代理程式，請參閱 [fteListAgents](#) 指令。

若要使用 IBM MQ Explorer 來列出代理程式，請在「導覽器」視圖中按一下協調佇列管理程式名稱下的代理程式。

如果代理程式未由 **fteListAgents** 指令列出或未顯示在 IBM MQ Explorer 中，請使用下列主題中的診斷流程圖來尋找並修正問題：[如果您的代理程式未由 fteListAgents 指令列出。](#)

停止 MFT 代理程式

您可以從指令行停止 Managed File Transfer 代理程式。停止代理程式時，代理程式開始靜止，但允許代理程式在停止之前完成現行檔案傳送。您也可以指令行指定 **-i** 參數，以立即停止代理程式。代理程式停止之後，一直到重新啟動之前，您都無法使用該代理程式來傳送檔案。

開始之前

如果您想要檢查與佇列管理程式相關聯的代理程式名稱，您可以使用 IBM MQ Explorer 或指令行來列出代理程式，請參閱 [fteListAgents](#) 指令。

關於這項作業



若要從指令行停止代理程式，請參閱 [fteStopAgent](#)。

如果您已配置代理程式以 Windows 服務方式執行，則執行 **fteStopAgent** 指令也會停止 Windows 服務。或者，您也可以使用 Windows「服務」工具來停止服務，以停止代理程式。如需相關資訊，請參閱第 129 頁的『[以 Windows 服務方式啟動 MFT 代理程式](#)』主題。

在 z/OS 上停止 MFT 代理程式

如果您從 JCL 以已啟動作業的方式在 z/OS 上執行 Managed File Transfer Agent，則除了 **fteStopAgent** 指令之外，代理程式還接受 z/OS 操作員指令 **MODIFY** 及 **STOP**。

使用啟動的作業是因為它是以特定的使用者 ID 執行，且不受使用者登出影響。

註：  通常，據以執行啟動作業的管理使用者可能沒有登入專用權，並因此無法以執行代理程式的使用者身分登入 z/OS 系統。無法對該代理程式發出 **fteStartAgent**、**fteStopAgent** 和 **fteSetAgentTraceLevel** 指令以及已指定 **-d** 參數的 **fteShowAgentDetails** 指令。

從 IBM MQ 9.0.2 和 IBM MQ 9.0.0 Fix Pack 1 開始，代理程式內容 **adminGroup** 可與 z/OS 上的 Managed File Transfer 代理程式搭配使用。您可以定義安全管理員群組（例如 MFTADMIN），然後新增已啟動作業使用者 ID 和管理者 TSO ID 至此群組。編輯代理程式內容檔，並將 **adminGroup** 內容設為此安全管理員群組的名稱。

```
adminGroup=MFTADMIN
```

此群組的成員隨後可以對正在當作已啟動作業執行的代理程式發出 **fteStartAgent**、**fteStopAgent** 和 **fteSetAgentTraceLevel** 以及已指定 **-d** 參數的 **fteShowAgentDetails** 指令。

如需相關資訊，請參閱第 463 頁的『[MFT agent.properties 檔案](#)』中的 **adminGroup** 內容。

使用 z/OS MODIFY 指令 (F) 執行受管制的代理程式關閉

MODIFY 指令可讓您以受管制的方式停止代理程式，這是 **fteStopAgent** 指令的替代方案。代理程式會完成目前正在進行的任何傳送，但代理程式不會啟動任何新的傳送。

例如：

```
F job_name,APPL=STOP
```

其中 *job_name* 是執行代理程式程序的工作。

使用 z/OS STOP 指令 (P) 執行立即的代理程式關閉

STOP 指令相等於使用 **fteStopAgent** 指令搭配 **-i** 參數執行的立即停止。即使代理程式目前正在傳送檔案，代理程式也會立即停止。

例如：

```
P job_name
```

其中 *job_name* 是執行代理程式程序的工作。

相關參考

第 132 頁的『[在 z/OS 上啟動 MFT 代理程式](#)』

在 z/OS 上，除了從「UNIX 系統服務」階段作業中執行 **fteStartAgent** 指令之外，您還可以從 JCL 以啟動作業的方式來啟動代理程式，而不需要互動式階段作業。

第 463 頁的『[MFT agent.properties 檔案](#)』

每一個 Managed File Transfer Agent 都有自己的內容檔 `agent.properties`，其中必須包含代理程式用來連接至其佇列管理程式的資訊。`agent.properties` 檔案也可以包含變更代理程式行為的內容。

通訊協定橋接器

通訊協定橋接器可供 Managed File Transfer (MFT) 網路存取儲存在 MFT 網路之外的檔案伺服器上的檔案（在本端網域或遠端位置）。此檔案伺服器可使用 FTP、FTPS 或 SFTP 網路通訊協定。每一個檔案伺服器需要至少一個專用的代理程式。專用代理程式即所謂的通訊協定橋接器代理程式。橋接器代理程式可以與多個檔案伺服器互動。

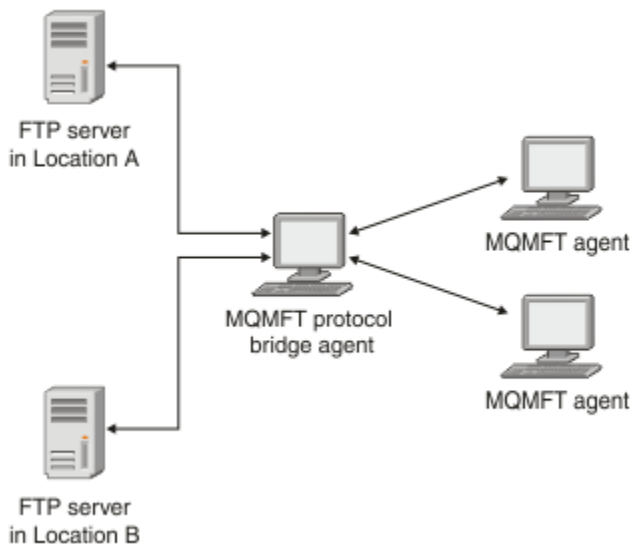
通訊協定橋接器可提供作為 Managed File Transfer 的服務元件的一部分。在連接至不同檔案伺服器的單一 MFT 系統上，您可以有多個專用的代理程式。

您可以使用通訊協定橋接器代理程式，將檔案同時傳送至多個端點。MFT 提供稱為 `ProtocolBridgeProperties.xml` 的檔案，您可以編輯此檔案，定義您想要將檔案傳送至哪些不同的通訊協定檔案伺服器。**fteCreateBridgeAgent** 指令為您將預設通訊協定檔案伺服器的詳細資料新增至 `ProtocolBridgeProperties.xml`。此檔案詳述於第 484 頁的『[通訊協定橋接器內容檔格式](#)』中。

您可以使用通訊協定橋接器代理程式來執行下列動作：

- 使用 FTP、FTPS 或 SFTP 從 MFT 網路將檔案上傳至遠端伺服器。
- 使用 FTP、FTPS 或 SFTP 從遠端伺服器將檔案下載至 MFT 網路

註：通訊協定橋接器代理程式僅支援允許以絕對檔案路徑存取檔案的 FTP、FTPS 或 SFTP 伺服器。如果在傳送要求中指定了相對檔案路徑，則通訊協定橋接器代理程式將會嘗試根據用來登入通訊協定伺服器的起始目錄，將相對路徑轉換成絕對檔案路徑。通訊協定橋接器代理程式不支援只能根據現行目錄來存取檔案的通訊協定伺服器。



此圖顯示不同位置上的兩個 FTP 伺服器。這些 FTP 伺服器用來與 Managed File Transfer 代理程式交換檔案。通訊協定橋接器代理程式位於 FTP 伺服器與 MFT 網路的其餘部分之間，且配置為與這兩個 FTP 伺服器進行通訊。

除了通訊協定橋接器代理程式之外，請確定您在 MFT 網路中還有另一個代理程式。通訊協定橋接器代理程式只是 FTP、FTPS 或 SFTP 伺服器的橋接器，不會將傳送的檔案寫入本端磁碟。如果要對 FTP、FTPS 或 SFTP 伺服器來回傳送檔案，您必須使用通訊協定橋接器代理程式作為檔案傳送的目的地或來源（代表 FTP、FTPS 或 SFTP 伺服器），並以另一個標準代理程式作為對應的來源或目的地。

使用通訊協定橋接器傳送檔案時，橋接器必須有權讀取您要傳送的檔案所在的來源或目的地目錄。比方說，如果您想要從只具有執行權限 (d--x--x--x) 的目錄 /home/fte/bridge 傳送檔案，則您嘗試從這個目錄執行的任何傳送都會失敗，並產生下列錯誤訊息：

```
BFGBR0032E: Attempt to read filename from the protocol file server
has failed with server error 550. Failed to open file.
```

配置通訊協定橋接器代理程式

通訊協定橋接器代理程式如同標準 MFT 代理程式。您可以使用 **fteCreateBridgeAgent** 指令來建立通訊協定橋接器代理程式。您可以使用 `ProtocolBridgeProperties.xml` 檔案來配置通訊協定橋接器代理程式，如第 484 頁的『通訊協定橋接器內容檔格式』中所述。如果您是使用舊版，請使用進階代理程式內容中所述的特定通訊協定橋接器內容來配置代理程式。對於所有版本，您也可以如第 186 頁的『對映檔案伺服器的認證』中所述配置認證對映。為特定通訊協定檔案伺服器配置通訊協定橋接器代理程式之後，該代理程式即可為此用途專用。

通訊協定橋接器回復

如果因為檔案伺服器無法使用，導致通訊協定橋接器代理程式無法連接至檔案伺服器，在檔案伺服器變成可用之前，所有檔案傳送要求都會排入佇列中。如果因為代理程式使用錯誤認證，導致通訊協定橋接器代理程式無法連接至檔案伺服器，則傳送會失敗，且傳送日誌訊息會反映此錯誤。如果通訊協定橋接器代理程式由於任何原因而結束，則所有要求的檔案傳送都會保留，並於通訊協定橋接器重新啟動之後繼續進行。

在檔案傳送期間，檔案通常以暫存檔形式寫入目的地，然後在傳送完成時重新命名。但是，如果傳送目的地是配置為限制寫入的通訊協定檔案伺服器（使用者可以將檔案上傳至通訊協定檔案伺服器，但無法以任何方式變更這些已上傳的檔案；實際上，使用者只能寫入一次），已傳送的檔案將直接寫入目的地。這表示如果傳送期間發生問題，目的地通訊協定檔案伺服器上會留下未完整寫入的檔案，Managed File Transfer 無法刪除或編輯這些檔案。在此情況下，傳送會失敗。

使用 ProtocolBridgeProperties.xml 檔定義通訊協定檔案伺服器的內容

使用 Managed File Transfer 在代理程式配置目錄中提供的 ProtocolBridgeProperties.xml 檔案，定義您想要來回傳送檔案的一個以上通訊協定檔案伺服器的內容。

關於這項作業

ftcCreateBridgeAgent 指令在代理程式配置目錄 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` 中建立 ProtocolBridgeProperties.xml 檔案。如果執行此指令時指定預設通訊協定檔案伺服器，則此指令也會在檔案中建立此伺服器的項目。

BFGCL0392I 訊息會提供 ProtocolBridgeProperties.xml 檔案的位置。

```
<?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\FTPSServerKeyStore.jks"
trustStorePassword="password">

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

</tns:ftpServer>

<tns:sftpServer name="mySFTPSServer" 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://wmgfte.ibm.com/ProtocolBridgeProperties"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://wmgfte.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>

```

此指令可以產生下列訊息：BFGCL0532I:

為了讓代理程式運作，必須手動建立其他認證檔。

By default this file is called ProtocolBridgeCredentials.xml and is located in the home directory of the user who starts the agent. For example, if this user started the agent the location would be: \$HOME/ProtocolBridgeCredentials.xml

如果您使用認證檔：

1. 請參閱下列文字以取得有關如何建立認證檔的進一步資訊。
2. 認證檔必須位於權限受限的目錄中。例如，其他使用者必須沒有讀取權。
3. 在 \$HOME 環境變數中為已啟動代理程式的使用者 ID 指定認證檔目錄位置，或編輯 ProtocolBridgeProperties.xml 檔案並在下列行中指定位置：

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

如果您想要新增其他非預設通訊協定伺服器，請編輯該檔案來定義其內容。此範例新增額外一部 FTP 伺服器。

註：通訊協定橋接器代理程式不支援檔案鎖定。這是因為 Managed File Transfer 不支援檔案伺服器上的檔案鎖定機制。

程序

1. 在檔案中插入下列幾行來定義通訊協定檔案伺服器作為 <tns:serverProperties> 的子元素：

```
<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. 然後，變更屬性的值：

- name 是通訊協定檔案伺服器的名稱
- host 是通訊協定檔案伺服器的主機名稱或 IP 位址
- port 是通訊協定檔案伺服器的埠號
- platform 是執行通訊協定檔案伺服器的平台
- timeZone 是執行通訊協定檔案伺服器的時區
- locale 是通訊協定檔案伺服器上使用的語言
- fileEncoding 是通訊協定檔案伺服器的字元編碼
- listFormat 是通訊協定檔案伺服器所傳回的檔案清單格式
- limitedWrite 決定寫入檔案伺服器時是否遵循預設模式，亦即建立暫存檔，然後在傳送完成時重新命名該檔案。若為配置為唯寫的檔案伺服器，則會直接以其最終名稱建立檔案。此內容值可以是 true 或 false。如果是通訊協定橋接器代理程式，則 limitedWrite 屬性會與 doNotUseTempOutputFile 代理程式內容一起使用。如果您想要使用暫存檔，則不得設定 doNotUseTempOutputFile 的值，而且必須將 limitedWrite 的值設為 false。其他任何設定組合意味著不會使用暫存檔。
- maxListFileNames 是在通訊協定檔案伺服器上的目錄中掃描檔案名稱時收集的名稱數目上限。
- maxListDirectoryLevels 是在通訊協定檔案伺服器上的目錄中掃描檔案名稱時遞迴的目錄層次數目上限。

如需這些屬性的詳細資料，包括屬性是必要或選用項目及其預設值，請參閱第 484 頁的『通訊協定橋接器內容檔格式』。

相關參考

第 484 頁的『通訊協定橋接器內容檔格式』

代理程式配置目錄中的 ProtocolBridgeProperties.xml 檔案，會定義通訊協定檔案伺服器的內容。

第 554 頁的『MFT 所使用的正規表示式』

Managed File Transfer 在許多情況下會使用正規表示式。例如，正規表示式可用來比對 Connect:Direct 安全認證的使用者 ID，或將檔案分割成多則訊息（每次符合正規表示式時建立一則新訊息）。Managed File Transfer 所使用的正規表示式語法是 java.util.regex API 所支援的語法。此正規表示式類似（但不同於）Perl 語言所使用的正規表示式語法。

ProtocolBridgePropertiesExit2: 查閱通訊協定檔案伺服器內容

如果您有大量通訊協定檔案伺服器，則可以實作 `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit2` 介面來查閱在傳送時參照的通訊協定檔案伺服器內容。您可以實作此介面，這更勝於維護 `ProtocolBridgeProperties.xml` 檔。Managed File Transfer 提供一個查閱通訊協定檔案伺服器內容的使用者結束程式範例。

配置查閱通訊協定橋接器內容的使用者結束程式

關於這項作業

任何查閱通訊協定橋接器內容的使用者結束程式都必須實作 `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit2` 介面。如需相關資訊，請參閱第 786 頁的『[ProtocolBridgePropertiesExit2.java 介面](#)』。

您可以類似其他使用者結束程式的方式，將多個通訊協定伺服器內容結束程式鏈結在一起。會根據在代理程式內容檔中使用 `protocolBridgePropertiesExitClasses` 內容指定的順序來呼叫結束程式。個別傳回所有起始設定方法，如果有一個以上傳回 `false` 值，代理程式就不會啟動。此錯誤報告於代理程式事件日誌中。

對所有結束程式的 `getProtocolServerProperties` 方法，只傳回一個整體結果。如果該方法傳回內容物件作為結果碼，則此值是傳回的結果，且不會呼叫後續結束程式的 `getProtocolServerProperties` 方法。如果該方法傳回空值作為結果碼，則會呼叫下一個結束程式的 `getProtocolServerProperties` 方法。如果沒有後續的結束程式，則傳回空值結果。出現空值的整體結果碼，代表通訊協定橋接器代理程式查閱失敗。

建議您使用 `ProtocolBridgePropertiesExit2.java` 介面，但如果需要 `ProtocolBridgePropertiesExit.java` 介面的相關資訊，請參閱第 184 頁的『[ProtocolBridgePropertiesExit: 查閱通訊協定檔案伺服器內容](#)』。

若要執行結束程式，請完成下列步驟：

程序

1. 編譯通訊協定伺服器內容使用者結束程式。
2. 建立包含已編譯結束程式及其套件結構的 Java 保存檔 (JAR)。
3. 將包含結束程式類別的 JAR 檔放在通訊協定橋接器代理程式的 `exits` 目錄。此目錄位於 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` 目錄中。
4. 編輯通訊協定橋接器代理程式的內容檔，以併入 `protocolBridgePropertiesExitClasses` 內容。以此內容的值而言，請指定以逗點區隔的類別清單，這些類別實作通訊協定橋接器伺服器內容使用者結束程式。會根據此清單中指定的順序來呼叫結束程式類別。如需相關資訊，請參閱第 463 頁的『[MFT agent.properties 檔案](#)』。
5. 您可以選擇性地指定 `protocolBridgePropertiesConfiguration` 內容。您對此內容指定的值會以字串形式傳遞給 `protocolBridgePropertiesExitClasses` 指定的結束程式類別的 `initialize()` 方法。如需相關資訊，請參閱第 463 頁的『[MFT agent.properties 檔案](#)』。

使用使用者結束程式範例

關於這項作業

`MQ_INSTALLATION_PATH/mqft/samples/protocolBridge` 目錄及第 224 頁的『[通訊協定橋接器內容使用者結束程式範例](#)』主題中提供了可供查閱通訊協定橋接器內容的使用者結束程式範例。

`SamplePropertiesExit2.java` 結束程式會讀取包含通訊協定伺服器內容的內容檔。此內容檔中每一個項目的格式如下所示：

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

內容檔的位置是從通訊協定橋接器代理程式內容 `protocolBridgePropertiesConfiguration` 取得。

若要執行使用者結束程式範例，請完成下列步驟：

程序

1. 編譯 `SamplePropertiesExit2.java` 檔案。
2. 建立包含已編譯結束程式及其套件結構的 JAR 檔。
3. 將 JAR 檔放在 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/exits` 目錄中。
4. 編輯 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` 檔案以包含此行：

```
protocolBridgePropertiesExitClasses=SamplePropertiesExit2
```

5. 在 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent` 目錄中建立通訊協定橋接器內容檔，例如 `protocol_bridge_properties.properties`。編輯此檔案，以併入此格式的項目：

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

6. 編輯 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/agent.properties` 檔案以包含此行：

```
protocolBridgePropertiesConfiguration=MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/protocol_bridge_properties.properties
```

您必須使用 `protocol_bridge_properties.properties` 檔的絕對路徑。

7. 使用 **`fteStartAgent`** 指令來啟動通訊協定橋接器代理程式。

相關概念

[第 178 頁的『通訊協定橋接器』](#)

通訊協定橋接器可供 Managed File Transfer (MFT) 網路存取儲存在 MFT 網路之外的檔案伺服器上的檔案（在本端網域或遠端位置）。此檔案伺服器可使用 FTP、FTPS 或 SFTP 網路通訊協定。每一個檔案伺服器需要至少一個專用的代理程式。專用代理程式即所謂的通訊協定橋接器代理程式。橋接器代理程式可以與多個檔案伺服器互動。

相關參考

[第 786 頁的『ProtocolBridgePropertiesExit2.java 介面』](#)

[第 224 頁的『通訊協定橋接器內容使用者結束程式範例』](#)

[第 463 頁的『MFT agent.properties 檔案』](#)

每一個 Managed File Transfer Agent 都有自己的內容檔 `agent.properties`，其中必須包含代理程式用來連接至其佇列管理程式的資訊。`agent.properties` 檔案也可以包含變更代理程式行為的內容。

[第 330 頁的『fteCreateBridgeAgent（建立及配置 MFT 通訊協定橋接器代理程式）』](#)

`fteCreateBridgeAgent` 指令會建立 Managed File Transfer 通訊協定橋接器代理程式及其相關聯的配置。為您要傳送檔案至其中及從中接收檔案的每一台檔案伺服器建立通訊協定橋接器代理程式。

ProtocolBridgePropertiesExit：查閱通訊協定檔案伺服器內容

如果您有大量通訊協定檔案伺服器，則可以實作

`com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` 介面來查閱在傳送時參照的通訊協定檔案伺服器內容。您可以實作此介面，這更勝於維護 `ProtocolBridgeProperties.xml` 檔。建議您使用 `ProtocolBridgePropertiesExit2.java` 介面，但同時也支援

`ProtocolBridgePropertiesExit.java` 介面。如果您已從 IBM WebSphere MQ File Transfer Edition 實作 `ProtocolBridgePropertiesExit.java` 介面，則可以在 IBM WebSphere MQ 7.5 或更新版本中使用該介面。

`ProtocolBridgePropertiesExit2.java` 中的新 `getCredentialLocation` 方法會使用 `ProtocolBridgeCredentials.xml` 檔案的預設位置，即您的起始目錄。

配置查閱通訊協定橋接器內容的使用者結束程式

任何查閱通訊協定橋接器內容的使用者結束程式都必須實作 `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` 介面。如需相關資訊，請參閱 [第 185 頁的『ProtocolBridgePropertiesExit.java 介面』](#)。

您可以類似其他使用者結束程式的方式，將多個通訊協定伺服器內容結束程式鏈結在一起。會根據在代理程式內容檔中使用 `protocolBridgePropertiesExitClasses` 內容指定的順序來呼叫結束程式。個別傳回所有起始設定方法，如果有一個以上傳回 `false` 值，代理程式就不會啟動。此錯誤報告於代理程式事件日誌中。

對所有結束程式的 `getProtocolServerProperties` 方法，只傳回一個整體結果。如果該方法傳回內容物件作為結果碼，則此值是傳回的結果，且不會呼叫後續結束程式的 `getProtocolServerProperties` 方法。如果該方法傳回空值作為結果碼，則會呼叫下一個結束程式的 `getProtocolServerProperties` 方法。如果沒有後續的結束程式，則傳回空值結果。出現空值的整體結果碼，代表通訊協定橋接器代理程式查閱失敗。

若要執行結束程式，請完成下列步驟：

1. 編譯通訊協定伺服器內容使用者結束程式。
2. 建立包含已編譯結束程式及其套件結構的 Java 保存檔 (JAR)。
3. 將包含結束程式類別的 JAR 檔放在通訊協定橋接器代理程式的 `exits` 目錄。此目錄位於 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` 目錄中。
4. 編輯通訊協定橋接器代理程式的內容檔，以併入 `protocolBridgePropertiesExitClasses` 內容。以此內容的值而言，請指定以逗點區隔的類別清單，這些類別實作通訊協定橋接器伺服器內容使用者結束程式。會根據此清單中指定的順序來呼叫結束程式類別。如需相關資訊，請參閱 [第 463 頁的『MFT agent.properties 檔案』](#)。
5. 您可以選擇性地指定 `protocolBridgePropertiesConfiguration` 內容。您對此內容指定的值會以字串形式傳遞給 `protocolBridgePropertiesExitClasses` 指定的結束程式類別的 `initialize()` 方法。如需相關資訊，請參閱 [第 463 頁的『MFT agent.properties 檔案』](#)。

ProtocolBridgePropertiesExit.java 介面

```
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);
}

```

對映檔案伺服器的認證

使用通訊協定橋接器代理程式的預設認證對映功能，或是撰寫您專屬的使用者結束程式，將 Managed File Transfer 中的使用者認證對映至檔案伺服器上的使用者認證。Managed File Transfer 提供一個使用者結束程式範例，可執行使用者認證對映。

使用 *ProtocolBridgeCredentials.xml* 檔來對映檔案伺服器的認證

使用通訊協定橋接器代理程式的預設認證對映功能，將 Managed File Transfer 中的使用者認證對映至檔案伺服器上的使用者認證。Managed File Transfer 提供 XML 檔案，您可以編輯此檔案以包括自己的認證資訊。

關於這項作業

ProtocolBridgeCredentials.xml 檔必須由使用者手動建立。依預設，此檔案的位置是啟動通訊協定橋接器代理程式之使用者的起始目錄，但此檔案可儲存於檔案系統上可供代理程式存取的任何位置上。To specify a different location, add the <credentialsFile> element to the ProtocolBridgeProperties.xml file. 例如：

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

請先編輯該檔案，併入主機、使用者及認證資訊，以設定認證對映，然後才能使用通訊協定橋接器代理程式。如需相關資訊及範例，請參閱第 481 頁的『通訊協定橋接器認證檔案格式』。

z/OS 當您使用 IBM WebSphere MQ 7.5 或更早版本在 z/OS 平台上建立 ProtocolBridgeCredentials.xml 檔案時，必須在編輯檔案之前設定檔案標籤。執行下列指令，將檔案標示為具有 ASCII 內容：

```
chtag -t -c IS08859-1 ProtocolBridgeCredentials.xml
```

註：**z/OS** 在 z/OS 上，您可以將通訊協定橋接器認證檔儲存在資料集上，使用者可在此資料集中指定 .xml 檔案的名稱。

程序

1. 編輯 <tns:server name="server name"> 這一行，將 name 屬性的值變更為 ProtocolBridgeProperties.xml 檔中的伺服器名稱。

為 IBM WebSphere MQ File Transfer Edition 7.0.4 及更舊版本所建立的通訊協定橋接器代理程式，沒有 ProtocolBridgeProperties.xml 檔案（或是相關的使用者結束程式），因此對 IBM

WebSphere MQ File Transfer Edition 7.0.4 Fix Pack 1 以及更新版本而言，會自動指派伺服器的主機名稱作為伺服器名稱。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 屬性來指定使用包含萬用字元或正規表示式的伺服器名稱。例如：

```
<tns:server name="serverA*" pattern="wildcard">
```

2. 將使用者 ID 及認證資訊，作為 <tns:server> 的子元素插入檔案中。

您可以在檔案中插入下列一個或多個元素：

- 如果通訊協定檔案伺服器是 FTP、FTPS 或 SFTP 伺服器，則可以使用密碼來鑑別要求傳送的使用者。將下列字行插入檔案中：

```
<tns:user name="FTE User ID"
  serverUserId="Server User ID"
  serverPassword="Server Password">
</tns:user>
```

然後，變更屬性的值。

- name 是 Java 正規表示式，用於比對與 MFT 傳送要求相關聯的 MQMD 使用者 ID
- serverUserId 是傳遞給通訊協定檔案伺服器作為登入使用者 ID 的值。如果未指定 serverUserId 屬性，則改用與 MFT 傳送要求相關聯的 MQMD 使用者 ID
- serverPassword 是與 serverUserId 相關聯的密碼。

name 屬性可以包含 Java 正規表示式。認證對映程式會試圖將 MFT 傳送要求的 MQMD 使用者 ID 與此正規表示式相比對。通訊協定橋接器代理程式會試圖將 MQMD 使用者 ID 與 <tns:user> 元素之 name 屬性中的正規表示式相比對（依元素存在於檔案中的順序進行）。找到相符的項目時，通訊協定橋接器代理程式就不會再尋找是否還有其他相符的項目。如果找到相符項目，即會將對應的 serverUserId 及 serverPassword 值傳遞給通訊協定檔案伺服器，作為登入使用者 ID 及密碼。MQMD 使用者 ID 比對須區分大小寫。

- 如果通訊協定檔案伺服器是 SFTP 伺服器，您可以使用公開和私密金鑰來鑑別要求傳送的使用者。在檔案中插入下列這幾行，並變更屬性的值。<tns:user> 元素可以包含一或多個 <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 正規表示式，用於比對與 MFT 傳送要求相關聯的 MQMD 使用者 ID
- serverUserId 是傳遞給通訊協定檔案伺服器作為登入使用者 ID 的值。如果未指定 serverUserId 屬性，則改用與 MFT 傳送要求相關聯的 MQMD 使用者 ID
- hostKey 是登入時預計會從伺服器傳回的金鑰
- key 是 serverUserId 的私密金鑰
- keyPassword 是金鑰的密碼，用來產生公開金鑰
- associationName 是用以識別追蹤及記載用途的值

name 屬性可以包含 Java 正規表示式。認證對映程式會試圖將 MFT 傳送要求的 MQMD 使用者 ID 與此正規表示式相比對。通訊協定橋接器代理程式會試圖將 MQMD 使用者 ID 與 <tns:user> 元素之 name 屬性中的正規表示式相比對（依元素存在於檔案中的順序進行）。找到相符的項目時，通訊協定橋接器代理程式就不會再尋找是否還有其他相符的項目。如果找到相符項目，則會使用對應的 serverUserId 及 key 值，向通訊協定檔案伺服器鑑別 MFT 使用者。MQMD 使用者 ID 比對須區分大小寫。

如需對通訊協定橋接器代理程式使用私密金鑰的相關資訊，請參閱第 190 頁的『[範例：如何配置通訊協定橋接器代理程式對 UNIX SFTP 伺服器使用私密金鑰認證](#)』。

註: **z/OS**

當傳送要求寫入指令佇列時，如果來源代理程式指令佇列位於 z/OS 或 IBM i 系統上，MQMD 使用者 ID 可能轉換為大寫。因此，相同原始使用者的 MQMD 使用者 ID 可能以原本的大小寫傳到認證結束程式或轉換為大寫傳到認證結束程式，端視傳送要求中指定的來源代理程式而定。預設認證對映結束程式須區分大小寫以比對提供的 MQMD 使用者 ID，而您在對映檔案中可能必須允許此行為。

使用結束類別對映檔案伺服器的認證

如果您不想要使用通訊協定橋接器代理程式的預設認證對映功能，您可以撰寫專屬的使用者結束程式，將 Managed File Transfer 中的使用者認證對映至檔案伺服器上的使用者認證。Managed File Transfer 提供一個使用者結束程式範例，可執行使用者認證對映。如果您配置認證對映使用者結束程式，它們會取代預設認證對映功能。

配置通訊協定橋接器認證使用者結束程式

關於這項作業

用來對映通訊協定橋接器認證的使用者結束程式，必須實作下列其中一個介面：

- `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit`，可讓通訊協定橋接器代理程式與一部預設通訊協定檔案伺服器來回傳送檔案
- `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit2`，可讓您與多個端點來回傳送檔案。

`com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit2` 介面包含與 `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit` 相同的函數，還包括延伸函數。如需相關資訊，請參閱第 784 頁的『[ProtocolBridgeCredentialExit.java 介面](#)』及第 785 頁的『[ProtocolBridgeCredentialExit2.java 介面](#)』。

認證結束程式可以採取類似其他使用者結束程式的方式鏈結在一起。結束程式依照它們在代理程式內容檔中使用 `protocolBridgeCredentialConfiguration` 內容指定的順序依序呼叫。個別傳回所有起始設定方法，如果有一個以上傳回 `false` 值，代理程式就不會啟動。此錯誤報告於代理程式事件日誌中。

對所有結束程式的 `mapMQUserId` 方法，只會傳回一個整體結果，如下所示：

- 如果該方法傳回 `USER_SUCCESSFULLY_MAPPED` 或 `USER_DENIED_ACCESS` 值作為結果碼，則此值是傳回的結果，且不會呼叫後續結束程式的 `mapMQUserId` 方法。
- 如果該方法傳回 `NO_MAPPING_FOUND` 值作為結果碼，則會呼叫下一個結束程式的 `mapMQUserId` 方法。
- 如果沒有後續的結束程式，則傳回 `NO_MAPPING_FOUND` 結果。
- 橋接器代理程式將整體結果碼 `USER_DENIED_ACCESS` 或 `NO_MAPPING_FOUND` 視為傳送失敗。

若要執行結束程式，請完成下列步驟：

程序

1. 編譯通訊協定橋接器認證使用者結束程式。
2. 建立包含已編譯結束程式及其套件結構的 Java 保存檔 (JAR)。
3. 將包含結束類別的 JAR 檔放在橋接器代理程式的 `exits` 目錄。此目錄位於 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` 目錄中。
4. 編輯通訊協定橋接器代理程式的內容檔，以併入 `protocolBridgeCredentialExitClasses` 內容。以此內容的值而言，請指定以逗點區隔的類別清單，這些類別實作通訊協定橋接器認證結束常式。會根據此清單中指定的順序來呼叫結束程式類別。如需相關資訊，請參閱第 463 頁的『[MFT agent.properties 檔案](#)』。
5. 編輯通訊協定橋接器代理程式的內容檔，以併入：

```
exitClassPath=IBM MQ
installation_directory\mqft\config\configuration_queue_manager\agents\protocol_bridge_agent_n
ame\exits\SampleCredentialExit.jar
```

代理程式的 `agent.properties` 檔案位於 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/bridge_agent_name` 目錄中。

如果變更 `agent.properties` 檔案，則必須重新啟動代理程式，以讓變更生效。

- 您可以選擇性地指定 `protocolBridgeCredentialConfiguration` 內容。您對此內容指定的值會以「字串」物件形式傳遞給 `protocolBridgeCredentialExitClasses` 指定的結束類別的 `initialize()` 方法。如需相關資訊，請參閱第 463 頁的『MFT `agent.properties` 檔案』。
- 使用 **`fteStartAgent`** 指令來啟動通訊協定橋接器代理程式。

使用使用者結束程式範例

關於這項作業

`MQ_INSTALLATION_PATH/mqft/samples/protocolBridge` 目錄和第 222 頁的『通訊協定橋接器認證使用者結束程式範例』主題中提供了通訊協定橋接器認證結束程式範例。此範例是基於 `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit` 介面。

`SampleCredentialExit.java` 結束程式會讀取內容檔，此內容檔將與傳送要求相關聯的 MQMD 使用者 ID 對映至伺服器使用者 ID 及伺服器密碼。此內容檔的位置是從通訊協定橋接器代理程式內容 `protocolBridgeCredentialConfiguration` 中取得。

若要執行使用者結束程式範例，請完成下列步驟：

程序

- 編譯 `SampleCredentialExit.java` 檔。
- 建立包含已編譯結束程式及其套件結構的 JAR 檔。
- 將 JAR 檔放在 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/exits` 目錄中。
- 編輯 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` 檔案以包含此行：

```
protocolBridgeCredentialExitClasses=SampleCredentialExit
```

- 編輯通訊協定橋接器代理程式的內容檔，以併入：

```
exitClassPath=IBM MQ
installation_directory\mqft\config\configuration_queue_manager\agents\protocol_bridge_agent_n
ame\exits\SampleCredentialExit.jar
```

代理程式的 `agent.properties` 檔案位於 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` 目錄中。

如果變更 `agent.properties` 檔案，則必須重新啟動代理程式，以讓變更生效。

- 在 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent` 目錄中建立認證內容檔 (`credentials.properties`)，並編輯它以包含此格式的項目：

```
mqUserId=serverUserId,serverPassword
```

- 編輯 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` 檔案以包含此行：

```
protocolBridgeCredentialConfiguration=MQ_DATA_PATH/mqft/
config/coordination_queue_manager/agents/bridge_agent_name/credentials.properties
```

您必須使用 `credentials.properties` 檔的絕對路徑。

- 使用 **`fteStartAgent`** 指令來啟動通訊協定橋接器代理程式。

範例：如何配置通訊協定橋接器代理程式對 UNIX SFTP 伺服器使用私密金鑰認證

此範例示範如何產生及配置 ProtocolBridgeCredentials.xml 檔。此範例是一般例子，詳細資料可能隨平台而不同，但原則不變。

關於這項作業

程序

1. 產生公開和私密金鑰，以用來向 SFTP 伺服器進行鑑別。

例如，在 Linux 主機系統上，您可以使用作為 'openssh' 套件一部分提供的工具 **ssh-keygen** 來建立公開/私密金鑰組。

依預設，在沒有引數的情況下，**ssh-keygen** 指令會提示輸入兩個金鑰檔的位置和通行詞組，其預設為名稱：

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



小心：如果您是從 OpenSSH 的最新版本（例如隨 RHEL 8 提供的版本）使用 **ssh-keygen** 指令，則使用的金鑰格式與通訊協定橋接器代理程式不相容，且對 SFTP 伺服器的傳送嘗試會失敗，並出現下列訊息：

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

若要使用這些較新 OpenSSH 版本建立相容的私密金鑰，請在 **ssh-keygen** 指令中使用下列引數指定金鑰格式：

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

然後，`id_rsa` 私密金鑰的內容具有下列項目的第一行和最後一行：

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

與通訊協定橋接器代理程式相容。

2. 將 `id_rsa.pub` 檔案的整個內容複製到 SFTP 伺服器上 SFTP 使用者的 `~/.ssh/authorized_keys` 檔案。

請確保已適當地設定此檔案及 `~/.ssh` 目錄上的檔案許可權，以便 SFTP 伺服器容許金鑰鑑別。這些許可權通常如下：

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

3. Managed File Transfer 需要使用 MD5 演算法產生的主機 ssh 指紋。執行下列其中一個指令，以取得 SFTP 伺服器的主機 ssh 指紋。

- 若為 Red Hat® Enterprise Linux 6.x 版及更舊版本，以及 Linux Ubuntu 14.04，請執行下列指令：

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

- 從 Red Hat Enterprise Linux 7.x 版、Linux Ubuntu 16.04 及 SuSE Linux 12.4 開始，依預設，**ssh-keygen** 指令會使用 SHA56 演算法來產生 ssh 指紋。若要使用 MD5 演算法產生 ssh 指紋，請執行下列指令：

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

指令的輸出將類似於下列範例：

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

僅擷取輸出的十六進位部分，以用作 ProtocolBridgeCredentials.xml 檔案中的 hostKey (請參閱步驟 第 191 頁的『4』)。因此，在此範例中，您將擷取 64:39:f5:49:41:10:55:d2:0b:81:42:5c:87:62:9d:27。

4. 在通訊協定橋接器代理程式系統上，編輯 ProtocolBridgeCredentials.xml 檔。將下列範例中以斜體顯示的值替換成您自己的值：

```
<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* 是通訊協定橋接器代理程式的名稱。
- *SFTP_host_name* 是 ProtocolBridgeProperties.xml 檔中顯示的 SFTP 伺服器名稱。
- *mq_User_ID* 是與傳送要求相關聯的 MQMD 使用者 ID。
- *SFTP_user_ID* 是步驟 2 中使用的 SFTP 使用者 ID。它是傳遞至 SFTP 伺服器作為登入使用者 ID 的值。
- *ssh_host_finger* 是步驟 3 所收集的指紋。
- *name* 是可指定作為追蹤及記載用途的名稱。
- *pass_phrase* 是您在步驟 1 中於 ssh-keygen 中提供的密碼詞組。
- *id_rsa* 檔案的完整內容是從步驟 1 產生之 id_rsa 檔案的完整內容。若要防止連線錯誤，請確保包含下列兩個項目：

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

您可以複製 <tns:privatekey> 元素來新增其他金鑰。

5. 啟動通訊協定橋接器代理程式（如果尚未啟動）。或者，通訊協定橋接器代理程式會定期輪詢 ProtocolBridgeCredentials.xml 檔來取得變更。

將通訊協定橋接器配置用於 FTPS 伺服器

採取您配置 FTP 伺服器時的類似作法來配置 FTPS 伺服器：為伺服器建立橋接器代理程式，定義伺服器內容，然後對映使用者認證。

關於這項作業

若要配置 FTPS 伺服器，請完成下列步驟：

程序

1. 使用 **fteCreateBridgeAgent** 指令為 FTPS 伺服器建立通訊協定橋接器代理程式。適用於 FTP 的參數也適用於 FTPS，但 FTPS 另外還有三個必要的特定參數：
 - a) **-bt** 參數。指定 FTPS 作為此參數的值。
 - b) 信任儲存庫檔案的 **-bts** 參數。此指令假設只需要伺服器鑑別，因此您必須指定信任儲存庫檔案的位置。

依預設是使用 **fteCreateBridgeAgent** 指令來配置 FTPS 通訊協定的明確格式，但您可以變更通訊協定橋接器內容檔來配置隱含格式。通訊協定橋接器一律以被動模式連接至 FTPS 伺服器。

如需 **fteCreateBridgeAgent** 指令的相關資訊，請參閱 [第 330 頁的『fteCreateBridgeAgent \(建立及配置 MFT 通訊協定橋接器代理程式\)』](#)。

如果您需要如何建立信任儲存庫檔案的相關指示，請參閱 IBM Developer 文章：[在 IBM WebSphere MQ File Transfer Edition 中配置 Secure Sockets Layer 連線功能](#)，或請參閱 [Oracle keytool 文件中的 keytool 相關資訊](#)。

2. Define the FTPS server properties within an `<ftpsServer>` element in the protocol bridge properties file: `ProtocolBridgeProperties.xml`. 如需相關資訊，請參閱 [第 180 頁的『使用 ProtocolBridgeProperties.xml 檔定義通訊協定檔案伺服器的內容』](#)。您也可以透過編輯通訊協定橋接器內容檔來啟用用戶端鑑別。如需所有配置選項的詳細資料，請參閱 [第 484 頁的『通訊協定橋接器內容檔格式』](#)。
3. 透過使用通訊協定橋接器代理程式的預設認證對映功能，或是撰寫您專屬的使用者結束程式，將 Managed File Transfer 中的使用者認證對映至 FTPS 伺服器上的使用者認證。如需相關資訊，請參閱 [第 186 頁的『對映檔案伺服器的認證』](#)。
4. 依預設，信任儲存庫檔案會配置為 JKS 格式；如果您想要變更格式，請編輯通訊協定橋接器內容檔。

範例

以下顯示通訊協定橋接器內容檔中的 FTPS 伺服器項目範例：

```
<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"/>

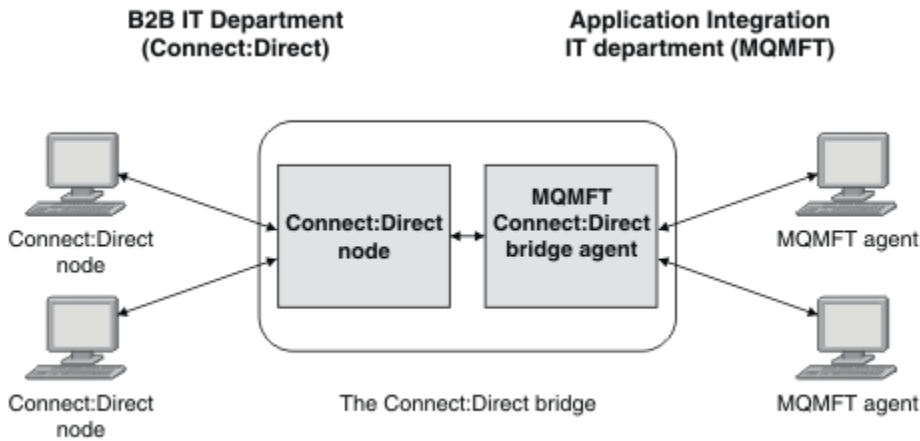
  <!-- Define servers here -->
</tns:serverProperties>
```

下一步

如需支援及不支援的 FTPS 通訊協定組件的相關資訊，請參閱 [第 558 頁的『通訊協定橋接器支援的 FTPS 伺服器』](#)。

Connect:Direct 橋接器

您可以從現有 IBM Sterling Connect:Direct 網路來回傳送檔案。使用 Connect:Direct 橋接器 (Managed File Transfer 元件)，可在 MFT 與 IBM Sterling Connect:Direct 之間傳送檔案。



此圖顯示兩個部門 (B2B IT 部門和「應用程式整合 IT」部門) 之間的 MFT Connect:Direct 橋接器。B2B IT 部門使用 Connect:Direct，從公司的事業夥伴來回傳送檔案。「應用程式整合 IT」部門使用 IBM MQ 作為其傳訊基礎架構，因此最近已選擇 Managed File Transfer 作為其檔案傳送解決方案。

透過使用 MFT Connect:Direct 橋接器，這兩個部門即可在 B2B IT 部門中的 Connect:Direct 網路與「應用程式整合 IT」部門中的 MFT 網路之間傳送檔案。Connect:Direct 橋接器是 Managed File Transfer 元件，其中包含與 Connect:Direct 節點通訊的 MFT 代理程式。MFT 代理程式專用於與 Connect:Direct 節點之間的傳送，因此也稱為 Connect:Direct 橋接器代理程式。

Connect:Direct 橋接器可提供作為 Managed File Transfer 的服務和代理程式元件的一部分，並可用於下列作業：

1. 使用 Managed File Transfer 指令起始一個或多個檔案的傳送（從 MFT 代理程式至 Connect:Direct 節點）。
2. 使用 Managed File Transfer 指令起始一個或多個檔案的傳送（從 Connect:Direct 節點至 MFT 代理程式）。
3. 使用 Managed File Transfer 指令起始會啟動使用者定義 Connect:Direct 程序的檔案傳送。
4. 使用 Connect:Direct 程序提交 MFT 檔案傳送要求。

Connect:Direct 橋接器可以將檔案僅傳送至 Connect:Direct 節點，或是僅從該節點傳送檔案。Connect:Direct 橋接器只能在 Connect:Direct 程序所提交的傳送過程中，將檔案傳送至其本端檔案系統，或從其本端檔案系統傳送檔案。

z/OS 您可以使用 Connect:Direct 橋接器，與位於 z/OS 系統上的 Connect:Direct 節點的資料集之間進行來回傳送。相較於僅涉及 Managed File Transfer 代理程式的資料集傳送，此傳送在行為上存在一些差異。如需相關資訊，請參閱第 534 頁的『從 Connect:Direct 節點來回傳送資料集』。

支援的平台

Connect:Direct 橋接器由 MFT Connect:Direct 橋接器代理程式，以及 Connect:Direct 節點組成。此代理程式在 Windows 和 Linux for System x 上受到支援。在 IBM Sterling Connect:Direct for Windows 和 IBM Sterling Connect:Direct for UNIX 支援的平台上支援此節點。如需建立 Connect:Direct 橋接器代理程式，以及配置與該代理程式通訊的 Connect:Direct 節點的相關指示，請參閱第 118 頁的『配置 Connect:Direct 橋接器』。

Connect:Direct 橋接器可以與 Connect:Direct 節點之間來回傳送檔案，這些節點是在 Connect:Direct for Windows、Connect:Direct for UNIX **z/OS** 或 Connect:Direct for z/OS 服務的安裝中執行。如需支援的 Connect:Direct 版本的詳細資料，請參閱 IBM MQ 系統需求網頁。

組成 Connect:Direct 橋接器的代理程式及節點必須在同一系統上，或可以存取同一檔案系統（例如透過共用 NFS 裝載）。此檔案系統用於在涉及 Connect:Direct 橋接器的檔案傳送期間，將檔案暫時儲存在 **cdTmpDir** 參數所定義的目錄中。Connect:Direct 橋接器代理程式及 Connect:Direct 橋接器節點，必須能使

用相同路徑名稱，對此目錄進行定址。例如，如果代理程式和節點是在不同的 Windows 系統上，則這些系統必須使用相同的磁碟機代號來裝載共用檔案系統。下列配置容許代理程式及節點使用相同路徑名稱：

- 代理程式和節點是在同一個系統上，它可能執行 Windows 或 Linux for System x
- 代理程式是在 Linux for System x 上，而節點是在 UNIX 上
- 代理程式是在一個 Windows 系統上，而節點是在另一個 Windows 系統上

下列配置不容許代理程式及節點使用相同路徑名稱：

- 代理程式是在 Linux for System x 上，而節點是在 Windows 上
- 代理程式是在 Windows 上，而節點是在 UNIX 上

規劃 Connect:Direct 橋接器安裝時，請考量此限制。

將檔案傳送至 Connect:Direct 節點

您可以使用 Connect:Direct 橋接器，將檔案從 Managed File Transfer 代理程式傳送至 Connect:Direct 節點。指定 Connect:Direct 節點作為傳送目的地，方法是指定 Connect:Direct 橋接器代理程式作為目的地代理程式，並以 `connect_direct_node_name:file_path` 格式指定目的地檔案。

開始之前

在傳送檔案之前，您必須配置屬於 Managed File Transfer 元件的 Connect:Direct 橋接器。如需相關資訊，請參閱第 118 頁的『[配置 Connect:Direct 橋接器](#)』。

關於這項作業

在此範例中，Connect:Direct 橋接器代理程式稱為 CD_BRIDGE。來源代理程式稱為 FTE_AGENT，並且可以是任何 WMQFTE 版本。目的地 Connect:Direct 節點稱為 CD_NODE1。要傳送的檔案位於 FTE_AGENT 所在系統上的 `/home/helen/file.log` 檔案路徑中。該檔案將傳送至執行 CD_NODE1 的系統上的 `/files/data.log` 檔案路徑中。

程序

1. 將 `fteCreateTransfer` 指令與 `-df`（目的地檔案）參數的值（格式為 `connect_direct_node_name:file_path`），以及指定為 Connect:Direct 橋接器代理程式名稱的 `-da`（目的地代理程式）參數的值搭配使用。

註：由 `connect_direct_node_name` 指定的 Connect:Direct 節點，是您要將檔案傳送至其中的節點，而非作為 Connect:Direct 橋接器一部分來運作的 Connect:Direct 節點。

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
                 -df CD_NODE1:/files/data.log /home/helen/file.log
```

如需相關資訊，請參閱第 367 頁的『[fteCreateTransfer：啟動新的檔案傳送](#)』。

2. 來源代理程式 FTE_AGENT 將檔案傳送至 Connect:Direct 橋接器代理程式 CD_BRIDGE。該檔案暫時儲存在執行 Connect:Direct 橋接器代理程式所在系統上，由 `cdTmpDir` 代理程式內容所定義的位置中。Connect:Direct 橋接器代理程式將該檔案傳送至 Connect:Direct 節點 CD_NODE1。

相關概念

第 192 頁的『[Connect:Direct 橋接器](#)』

您可以從現有 IBM Sterling Connect:Direct 網路來回傳送檔案。使用 Connect:Direct 橋接器（Managed File Transfer 元件），可在 MFT 與 IBM Sterling Connect:Direct 之間傳送檔案。

相關工作

第 195 頁的『[從 Connect:Direct 節點傳送檔案](#)』

您可以使用 Connect:Direct 橋接器，將檔案從 Connect:Direct 節點傳送至 Managed File Transfer Agent。您可以透過下列方式，將 Connect:Direct 節點指定為傳送來源：將 Connect:Direct 橋接器代理程式指定為來源代理程式，並以 `connect_direct_node_name:file_path` 格式指定來源規格。

相關參考

第 463 頁的『[MFT agent.properties 檔案](#)』

每一個 Managed File Transfer Agent 都有自己的內容檔 `agent.properties`，其中必須包含代理程式用來連接至其佇列管理程式的資訊。`agent.properties` 檔案也可以包含變更代理程式行為的內容。

從 Connect:Direct 節點傳送檔案

您可以使用 Connect:Direct 橋接器，將檔案從 Connect:Direct 節點傳送至 Managed File Transfer Agent。您可以透過下列方式，將 Connect:Direct 節點指定為傳送來源：將 Connect:Direct 橋接器代理程式指定為來源代理程式，並以 `connect_direct_node_name:file_path` 格式指定來源規格。

開始之前

在傳送檔案之前，您必須配置屬於 Managed File Transfer 元件的 Connect:Direct 橋接器。請參閱 [第 118 頁的『配置 Connect:Direct 橋接器』](#)。

關於這項作業

在此範例中，Connect:Direct 橋接器代理程式稱為 CD_BRIDGE。目的地代理程式稱為 FTE_AGENT，它可以是任何 Managed File Transfer 版本。來源 Connect:Direct 節點稱為 CD_NODE1。要傳送的檔案位於 CD_NODE1 所在系統上的 `/home/brian/in.file` 檔案路徑中。該檔案將傳送至執行 FTE_AGENT 的系統上的 `/files/out.file` 檔案路徑中。

程序

搭配使用 `fteCreateTransfer` 指令與來源規格的值 (格式為 `connect_direct_node_name:file_path`)，以及指定為 Connect:Direct 橋接器代理程式名稱的 `-sa` 參數值。

註：由 `connect_direct_node_name` 指定的 Connect:Direct 節點，是您要從中傳送檔案的節點，而非作為 Connect:Direct 橋接器一部分來運作的 Connect:Direct 節點。例如：

```
fteCreateTransfer -sa CD_BRIDGE -da FTE_AGENT
                  -df /files/out.file CD_NODE1:/home/brian/in.file
```

如需相關資訊，請參閱 [第 367 頁的『fteCreateTransfer: 啟動新的檔案傳送』](#)。

結果

Connect:Direct 橋接器代理程式 CD_BRIDGE 要求來自 Connect:Direct 節點 CD_NODE1 的檔案。Connect:Direct 節點將該檔案傳送至 Connect:Direct 橋接器。在從 Connect:Direct 節點傳送該檔案時，Connect:Direct 橋接器會將該檔案暫時儲存在 `cdTmpDir` 代理程式內容所定義的位置中。將該檔案從 Connect:Direct 節點傳送至 Connect:Direct 橋接器完畢時，Connect:Direct 橋接器會將該檔案傳送至目的地代理程式 FTE_AGENT，然後從暫存位置刪除該檔案。

相關概念

[第 192 頁的『Connect:Direct 橋接器』](#)

您可以從現有 IBM Sterling Connect:Direct 網路來回傳送檔案。使用 Connect:Direct 橋接器 (Managed File Transfer 元件)，可在 MFT 與 IBM Sterling Connect:Direct 之間傳送檔案。

相關參考

[第 463 頁的『MFT agent.properties 檔案』](#)

每一個 Managed File Transfer Agent 都有自己的內容檔 `agent.properties`，其中必須包含代理程式用來連接至其佇列管理程式的資訊。`agent.properties` 檔案也可以包含變更代理程式行為的內容。

將資料集傳送至 z/OS 上的 Connect:Direct 節點

您可以使用位於 Windows 或 Linux 系統上的 Connect:Direct 橋接器，從 z/OS 上的 Managed File Transfer 代理程式，將資料集傳送至 z/OS 上的 Connect:Direct 節點。

開始之前

在傳送檔案之前，您必須配置屬於 Managed File Transfer 元件的 Connect:Direct 橋接器。請參閱 [第 118 頁的『配置 Connect:Direct 橋接器』](#)。

關於這項作業

在此範例中，**-df** 參數用於指定傳送目的地。**-df** 參數適合在傳送的來源代理程式為任何版本的 Managed File Transfer 時使用。如果來源代理程式是 IBM WebSphere MQ File Transfer Edition 7.0.4 或更新版本，則您可以改用 **-ds** 參數。來源代理程式稱為 FTE_ZOS1，並且是 IBM WebSphere MQ File Transfer Edition 7.0.3 代理程式。Connect:Direct 橋接器代理程式稱為 CD_BRIDGE，它位於 Linux 系統上。目的地 Connect:Direct 節點稱為 CD_ZOS2。來源代理程式和目的地 Connect:Direct 節點都位於 z/OS 系統上。要傳送的資料集位於 FTE_ZOS1 所在系統上的 //FTEUSER.SOURCE.LIB 中。該資料集將傳送至 CD_ZOS2 所在系統上的 //CDUSER.DEST.LIB 資料集。

程序

1. 將 fteCreateTransfer 指令與 **-df** 參數的值（格式為 `connect_direct_node_name:data_set_name;attributes`），以及指定為 Connect:Direct 橋接器代理程式名稱的 **-da**（目的地代理程式）參數的值搭配使用。

由 `connect_direct_node_name` 指定的 Connect:Direct 節點，是您要將資料集傳送至其中的節點，而非作為 Connect:Direct 橋接器一部分來運作的 Connect:Direct 節點。

由 `data_set_name` 指定的資料集名稱必須是絕對（而非相對）名稱。Connect:Direct 不會以使用者名稱作為資料集名稱的字首。

```
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'
```

如需相關資訊，請參閱 [第 367 頁的『fteCreateTransfer: 啟動新的檔案傳送』](#)。

2. 來源代理程式 FTE_ZOS1 將資料集中的資料，傳送至 Connect:Direct 橋接器代理程式 CD_BRIDGE。該資料將作為純文字檔，暫時儲存在執行 Connect:Direct 橋接器代理程式的系統上，由 cdTmpDir 代理程式內容所定義的位置中。Connect:Direct 橋接器代理程式將該資料傳送至 Connect:Direct 節點 CD_ZOS2。當此傳送完成時，即會從執行 Connect:Direct 橋接器代理程式的系統刪除該純文字檔。

相關概念

[第 192 頁的『Connect:Direct 橋接器』](#)

您可以從現有 IBM Sterling Connect:Direct 網路來回傳送檔案。使用 Connect:Direct 橋接器（Managed File Transfer 元件），可在 MFT 與 IBM Sterling Connect:Direct 之間傳送檔案。

相關參考

[第 534 頁的『從 Connect:Direct 節點來回傳送資料集』](#)

您可以使用 Connect:Direct 橋接器，在 Managed File Transfer 代理程式與 IBM Sterling Connect:Direct 節點之間傳送資料集。您可以將資料集指定為傳送來源及/或傳送目的地。

[第 536 頁的『Connect:Direct 程序陳述式參數與 BPXWDYN 鍵之間的對映』](#)

當您針對來源或目的地為 Connect:Direct 節點的資料集提交傳送要求時，您提供的任何受支援的 BPXWDYN 鍵，均會轉換為 Connect:Direct 程序接受的格式。

將多個檔案傳送至 Connect:Direct 節點

您可以使用 Connect:Direct 橋接器，將多個檔案從 Managed File Transfer Agent 傳送至 Connect:Direct 節點。若要將 Connect:Direct 節點用作多個檔案傳送目的地，請將 Connect:Direct 橋接器代理程式指定為目的地代理程式，並以 `connect_direct_node_name:directory_path` 格式指定目的地目錄。

開始之前

在傳送檔案之前，您必須配置屬於 Managed File Transfer 元件的 Connect:Direct 橋接器。請參閱 [第 118 頁的『配置 Connect:Direct 橋接器』](#)。

關於這項作業

在此範例中，來源代理程式稱為 FTE_AGENT。Connect:Direct 橋接器代理程式稱為 CD_BRIDGE。目的地 Connect:Direct 節點稱為 CD_NODE1。要傳送的檔案為 FTE_AGENT 所在系統上的 /home/jack/

data.log、/logs/log1.txt 及 /results/latest。這些檔案將傳送至執行 CD_NODE1 的系統上的 /in/files 目錄。

程序

將 `fteCreateTransfer` 指令與 `-dd` (目的地目錄) 參數的值 (格式為 `connect_direct_node_name:directory_path`) 搭配使用。將 `-da` (目的地代理程式) 參數的值, 指定為 `Connect:Direct` 橋接器代理程式的名稱。

註: 由 `connect_direct_node_name` 指定的 `Connect:Direct` 節點, 是您要將檔案傳送至其中的節點, 而非作為 `Connect:Direct` 橋接器一部分來運作的 `Connect:Direct` 節點。

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
                  -dd CD_NODE1:/in/files /home/jack/data.log
                  /logs/log1.txt /results/latest
```

如需相關資訊, 請參閱 [第 367 頁的『fteCreateTransfer: 啟動新的檔案傳送』](#)。

結果

來源代理程式 `FTE_AGENT` 將第一個檔案, 傳送至 `Connect:Direct` 橋接器代理程式 `CD_BRIDGE`。`Connect:Direct` 橋接器代理程式暫時將該檔案儲存在 `cdTmpDir` 內容所定義的位置中。將該檔案從來源代理程式完全傳送至 `Connect:Direct` 橋接器後, `Connect:Direct` 橋接器代理程式會將該檔案, 傳送至 `cdNode` 代理程式內容所定義的 `Connect:Direct` 節點。此節點將該檔案傳送至目的地 `Connect:Direct` 節點 `CD_NODE1`。當兩個 `Connect:Direct` 節點之間的傳送完成時, `Connect:Direct` 橋接器代理程式會從暫存位置刪除該檔案。對於每個指定的來源檔案, 將會重複執行此程序。

相關概念

[第 192 頁的『Connect:Direct 橋接器』](#)

您可以從現有 IBM Sterling `Connect:Direct` 網路來回傳送檔案。使用 `Connect:Direct` 橋接器 (Managed File Transfer 元件), 可在 MFT 與 IBM Sterling `Connect:Direct` 之間傳送檔案。

相關工作

[第 194 頁的『將檔案傳送至 Connect:Direct 節點』](#)

您可以使用 `Connect:Direct` 橋接器, 將檔案從 Managed File Transfer 代理程式傳送至 `Connect:Direct` 節點。指定 `Connect:Direct` 節點作為傳送目的地, 方法是指定 `Connect:Direct` 橋接器代理程式作為目的地代理程式, 並以 `connect_direct_node_name:file_path` 格式指定目的地檔案。

[第 198 頁的『使用萬用字元將多個檔案傳送至 Connect:Direct』](#)

若要將多個檔案從 Managed File Transfer 代理程式傳送至 `Connect:Direct` 節點, 請使用 `Connect:Direct` 橋接器。您可以在提供給 `fteCreateTransfer` 指令的來源規格中使用萬用字元。如同涉及萬用字元的所有 Managed File Transfer 傳送一樣, 只有檔案路徑的最後部分才能包含萬用字元。例如, `/abc/def*` 是有效的檔案路徑, 而 `/abc*/def` 無效。

[第 195 頁的『從 Connect:Direct 節點傳送檔案』](#)

您可以使用 `Connect:Direct` 橋接器, 將檔案從 `Connect:Direct` 節點傳送至 Managed File Transfer Agent。您可以透過下列方式, 將 `Connect:Direct` 節點指定為傳送來源: 將 `Connect:Direct` 橋接器代理程式指定為來源代理程式, 並以 `connect_direct_node_name:file_path` 格式指定來源規格。

[第 198 頁的『從 Connect:Direct 節點傳送多個檔案』](#)

您可以使用 `Connect:Direct` 橋接器, 將多個檔案從 `Connect:Direct` 節點傳送至 Managed File Transfer Agent。可以透過下列方式將 `Connect:Direct` 節點指定為多個檔案傳送的來源: 將 `Connect:Direct` 橋接器代理程式指定為來源代理程式, 並以 `connect_direct_node_name:file_path` 格式指定一個以上的來源規格。

相關參考

[第 463 頁的『MFT agent.properties 檔案』](#)

每一個 Managed File Transfer Agent 都有自己的內容檔 `agent.properties`, 其中必須包含代理程式用來連接至其佇列管理程式的資訊。`agent.properties` 檔案也可以包含變更代理程式行為的內容。

從 Connect:Direct 節點傳送多個檔案

您可以使用 Connect:Direct 橋接器，將多個檔案從 Connect:Direct 節點傳送至 Managed File Transfer Agent。可以透過下列方式將 Connect:Direct 節點指定為多個檔案傳送的來源：將 Connect:Direct 橋接器代理程式指定為來源代理程式，並以 `connect_direct_node_name:file_path` 格式指定一個以上的來源規格。

開始之前

在傳送檔案之前，您必須配置屬於 Managed File Transfer 元件的 Connect:Direct 橋接器。請參閱第 118 頁的『[配置 Connect:Direct 橋接器](#)』。

關於這項作業

在此範例中，Connect:Direct 橋接器代理程式稱為 CD_BRIDGE。目的地代理程式稱為 FTE_Z，它是在 z/OS 系統上執行。來源 Connect:Direct 節點稱為 CD_NODE1。要傳送的檔案為 CD_NODE1 所在系統上的 `/in/file1`、`/in/file2` 及 `/in/file3` 檔案路徑中。這些檔案將傳送至執行 FTE_Z 的系統上的已分割資料集 `//OBJECT.LIB`。

程序

搭配使用 `fteCreateTransfer` 指令與來源規格的值 (格式為 `connect_direct_node_name:file_path`)，以及指定為 Connect:Direct 橋接器代理程式名稱的 `-sa` 參數值。

註：由 `connect_direct_node_name` 指定的 Connect:Direct 節點，是您要從中傳送檔案的節點，而非作為 Connect:Direct 橋接器一部分來運作的 Connect:Direct 節點。

```
fteCreateTransfer -sa CD_BRIDGE -da FTE_Z
                  -dp '//OBJECT.LIB' CD_NODE1:/in/file1
                  CD_NODE1:/in/file2 CD_NODE1:/in/file3
```

如需相關資訊，請參閱第 367 頁的『[fteCreateTransfer: 啟動新的檔案傳送](#)』。

結果

Connect:Direct 橋接器代理程式 CD_BRIDGE 要求來自 Connect:Direct 節點 CD_NODE1 的第一個檔案。Connect:Direct 節點將該檔案傳送至 Connect:Direct 橋接器。在從 Connect:Direct 節點傳送該檔案時，Connect:Direct 橋接器會將該檔案暫時儲存在 `cdTmpDir` 代理程式內容所定義的位置中。將該檔案從 Connect:Direct 節點傳送至 Connect:Direct 橋接器完畢時，Connect:Direct 橋接器會將該檔案傳送至目的地代理程式 FTE_Z，然後從暫存位置刪除該檔案。對於每個指定的來源檔案，將會重複執行此程序。

相關概念

第 192 頁的『[Connect:Direct 橋接器](#)』

您可以從現有 IBM Sterling Connect:Direct 網路來回傳送檔案。使用 Connect:Direct 橋接器 (Managed File Transfer 元件)，可在 MFT 與 IBM Sterling Connect:Direct 之間傳送檔案。

相關參考

第 463 頁的『[MFT agent.properties 檔案](#)』

每一個 Managed File Transfer Agent 都有自己的內容檔 `agent.properties`，其中必須包含代理程式用來連接至其佇列管理程式的資訊。`agent.properties` 檔案也可以包含變更代理程式行為的內容。

使用萬用字元將多個檔案傳送至 Connect:Direct

若要將多個檔案從 Managed File Transfer 代理程式傳送至 Connect:Direct 節點，請使用 Connect:Direct 橋接器。您可以在提供給 `fteCreateTransfer` 指令的來源規格中使用萬用字元。如同涉及萬用字元的所有 Managed File Transfer 傳送一樣，只有檔案路徑的最後部分才能包含萬用字元。例如，`/abc/def*` 是有效的檔案路徑，而 `/abc*/def` 無效。

開始之前

在傳送檔案之前，您必須配置屬於 Managed File Transfer 元件的 Connect:Direct 橋接器。如需相關資訊，請參閱第 118 頁的『[配置 Connect:Direct 橋接器](#)』。

關於這項作業

在此範例中，來源代理程式稱為 FTE_AGENT，Connect:Direct 橋接器代理程式稱為 CD_BRIDGE。目的地 Connect:Direct 節點稱為 CD_NODE1。要傳送的檔案為 FTE_AGENT 所在系統上的 /reports 目錄中。僅會傳送名稱以 report 開頭，後面接著兩個字元且字尾為 .log 的檔案。例如，將會傳送 /reports/report01.log 檔案，但不會傳送 /reports/report1.log 檔案。這些檔案將傳送至執行 CD_NODE1 的系統上的 /home/fred 目錄。

程序

1. 將 fteCreateTransfer 指令與 **-dd** (目的地目錄) 參數的值 (格式為 `connect_direct_node_name:directory_path`) 搭配使用。對於 **-da** (目的地代理程式) 參數，請指定 Connect:Direct 橋接器代理程式。

註: 由 `connect_direct_node_name` 指定的 Connect:Direct 節點，是您要將檔案傳送至其中的節點，而非作為 Connect:Direct 橋接器一部分來運作的 Connect:Direct 節點。

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
                  -dd CD_NODE1:/home/fred "/reports/report??.log"
```

如需相關資訊，請參閱第 367 頁的『fteCreateTransfer: 啟動新的檔案傳送』。

2. 來源代理程式 FTE_AGENT 將符合 /reports/report??.log 型樣的第一個檔案，傳送至 Connect:Direct 橋接器代理程式 CD_BRIDGE。Connect:Direct 橋接器代理程式暫時將該檔案儲存在 cdTmpDir 內容所定義的位置中。將該檔案從來源代理程式完全傳送至 Connect:Direct 橋接器後，Connect:Direct 橋接器代理程式會將該檔案，傳送至 cdNode 代理程式內容所定義的 Connect:Direct 節點。此節點將該檔案傳送至目的地 Connect:Direct 節點 CD_NODE1。當兩個 Connect:Direct 節點之間的傳送完成時，Connect:Direct 橋接器代理程式會從暫存位置刪除該檔案。對於符合 /reports/report??.log 萬用字元型樣的每個來源檔案，將會重複執行此程序。

註: 符合 /reports/report??.log 型樣的檔案清單，會因來源代理程式 FTE_AGENT 所在系統的作業系統而異。

- 如果來源代理程式位於含有 Windows 作業系統的系統上，則型樣相符將不區分大小寫。此型樣將符合 /reports 目錄中，檔名格式為 report 後面接著兩個字元且字尾為 .log (不論字母大小寫為何) 的所有檔案。例如，Report99.Log 便是相符項。
- 如果來源代理程式位於含有 Linux 或 UNIX 作業系統的系統上，則型樣相符將區分大小寫。此型樣僅符合 /reports 目錄中，檔名格式為 report 後面接著兩個字元且字尾為 .log 的檔案。例如，reportAB.log 是相符項，但 reportAB.LOG 及 Report99.Log 不是相符項。

回復及重新啟動從 Connect:Direct 節點來回進行的傳送

在傳送期間，Managed File Transfer 可能會無法連接至 IBM Sterling Connect:Direct 節點；比方說，例如該節點無法使用。Managed File Transfer 會嘗試回復傳送，否則傳送會失敗並產生錯誤訊息。

如果 Connect:Direct 節點無法使用

如果 Connect:Direct 節點無法使用 (例如，由於網路或電源中斷)，Managed File Transfer 會按下列方式回復檔案傳送：

- 在此傳送要求中，如果 Managed File Transfer 先前未順利連接至 Connect:Direct 節點，則會在 **cdMaxConnectionRetries** 及 **recoverableTransferRetryInterval properties** 的值所決定的時間長度內重試傳送。在 Connect:Direct 橋接器代理程式的 `agent.properties` 檔案中，可以指定這些內容。在失敗嘗試次數達到 **cdMaxConnectionRetries property** 的值之後，傳送失敗並產生錯誤訊息。依預設，將會以 60 秒為間隔，無限地嘗試傳送。
- 在執行此傳送要求的過程中，如果 Managed File Transfer 先前曾順利連接至 Connect:Direct 節點，則會在 **cdMaxPartialWorkConnectionRetries** 及 **recoverableTransferRetryInterval** 內容值所決定的時間長度內再次嘗試傳送。當失敗的嘗試次數達到 **cdMaxPartialWorkConnectionRetries** 內容值後，傳送會失敗並產生錯誤訊息。依預設，將會以 60 秒為間隔，無限地嘗試傳送。

- 對於某些類型的 Connect:Direct 節點故障 (例如強制停止節點)，當節點回復時，Connect:Direct 處理程序會進入 Held Due to Error (HE) 狀態。當該節點回復之後，Managed File Transfer 會自動回復與檔案傳送相關且狀態為 HE 的所有 Connect:Direct 程序。
- 如果傳送失敗，將會從管理 Connect:Direct 橋接器的系統，刪除與該傳送相關的所有暫存檔。這些暫存檔的位置由 `cdTmpDir` 內容定義。
- 如果是從 Managed File Transfer 傳送至 Connect:Direct，且將來源處置方式指定為刪除，則在傳送失敗時不會刪除來源檔案。

如果 Connect:Direct 節點使用者認證無效

如果 Managed File Transfer 因 Connect:Direct 節點拒絕使用者認證而無法連接至該節點，傳送會失敗並產生錯誤訊息。在此狀況下，請檢查您是否已提供適用於 Connect:Direct 節點的正確使用者認證。如需相關資訊，請參閱第 120 頁的『對映 Connect:Direct 的認證』。

如果 Connect:Direct 橋接器代理程式無法使用

如果 Connect:Direct 橋接器代理程式無法使用，所有進行中檔案傳送的回復方式，都與標準 Managed File Transfer 傳送相同。如需相關資訊，請參閱第 206 頁的『MFT 回復及重新啟動』。

相關概念

第 192 頁的『Connect:Direct 橋接器』

您可以從現有 IBM Sterling Connect:Direct 網路來回傳送檔案。使用 Connect:Direct 橋接器 (Managed File Transfer 元件)，可在 MFT 與 IBM Sterling Connect:Direct 之間傳送檔案。

第 206 頁的『MFT 回復及重新啟動』

如果代理程式或佇列管理程式因故無法使用，例如因為電源或網路故障，在這些情況下，Managed File Transfer 會回復，如下所示：

相關工作

第 118 頁的『配置 Connect:Direct 橋接器』

配置 Connect:Direct 橋接器，以在 Managed File Transfer 網路與 Connect:Direct 網路之間傳送檔案。

Connect:Direct 橋接器的元件為 Connect:Direct 節點，以及專門用來與該節點通訊的 Managed File Transfer 代理程式。此代理程式稱為 Connect:Direct 橋接器代理程式。

相關參考

第 463 頁的『MFT agent.properties 檔案』

每一個 Managed File Transfer Agent 都有自己的內容檔 `agent.properties`，其中必須包含代理程式用來連接至其佇列管理程式的資訊。`agent.properties` 檔案也可以包含變更代理程式行為的內容。

透過檔案傳送要求提交使用者定義的 Connect:Direct 程序

您可以提交傳送要求，以經由 Connect:Direct 橋接器代理程式進行傳送。在檔案傳送過程中，該代理程式會呼叫使用者定義的 Connect:Direct 程序。

依預設，當您提交檔案傳送要求以經由 Connect:Direct 橋接器進行傳送時，Connect:Direct 橋接器代理程式會產生 Connect:Direct 程序，用於將檔案傳送至遠端 Connect:Direct 節點，或是從該節點傳送檔案。

但是，您可以使用 `ConnectDirectProcessDefinition.xml` 檔案，將 Connect:Direct 橋接器代理程式，配置成改為呼叫使用者定義的 Connect:Direct 程序。

ConnectDirectProcessDefinition.xml 檔案

`fteCreateCDAgent` 指令在代理程式配置目錄 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name` 中建立

`ConnectDirectProcessDefinitions.xml` 檔案。您必須先編輯此檔案以設定程序定義，然後才能從 Connect:Direct 橋接器代理程式，呼叫使用者定義的 Connect:Direct 程序。

此檔案會定義一個以上的程序集，其中包括一個以上在傳送過程中呼叫的 Connect:Direct 程序的位置。每個程序集均包括許多條件。如果傳送滿足程序集的所有條件，則會使用此程序集來指定該傳送呼叫的 Connect:Direct 程序。如需相關資訊，請參閱第 125 頁的『使用 `ConnectDirectProcessDefinition.xml` 檔案指定要啟動的 Connect:Direct 程序』。

本質符號變數

您可以使用 Managed File Transfer 定義的本質符號變數，將值代入使用者定義的 Connect:Direct 程序。為了遵循 Connect:Direct 命名慣例，Managed File Transfer 使用的所有本質符號變數均會具有下列格式：
%FTE 後接五個大寫英數字元。

建立程序以將檔案從 Connect:Direct 節點傳送至 Connect:Direct 橋接器系統時，您必須在 Connect:Direct 程序中，將本質變數 %FTETFILE 用作 TO FILE 的值。建立程序以將檔案從 Connect:Direct 橋接器系統傳送至 Connect:Direct 節點時，您必須在 Connect:Direct 程序中，將本質變數 %FTEFFILE 用作 FROM FILE 的值。這些變數包含 Connect:Direct 橋接器代理程式在 Managed File Transfer 網路中進行傳入及傳出時，所用的暫存檔路徑。

如需本質符號變數的相關資訊，請參閱 Connect:Direct 產品文件。

範例 Connect:Direct 程序

Managed File Transfer 提供範例 Connect:Direct 程序。這些範例位於下列目錄中：
`MQ_INSTALLATION_PATH/mqft/samples/ConnectDirectProcessTemplates`。

使用 *ConnectDirectProcessDefinition.xml* 檔案指定要啟動的 *Connect:Direct* 程序

指定要在 Managed File Transfer 傳送過程中啟動的 Connect:Direct 程序。Managed File Transfer 提供 XML 檔案，您可以編輯此檔案以指定程序定義。

關於這項作業

`fteCreateCDAgent` 指令在代理程式配置目錄 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name` 中建立 `ConnectDirectProcessDefinitions.xml` 檔案。您必須先編輯此檔案以設定程序定義，然後才能從 Connect:Direct 橋接器代理程式，呼叫使用者定義的 Connect:Direct 程序。

對於您要指定在透過 Connect:Direct 橋接器進行傳送過程中呼叫的每個程序，請執行下列步驟：

程序

1. 定義您要讓 Connect:Direct 橋接器代理程式，在傳送過程中呼叫的 Connect:Direct 程序，然後將程序範本儲存在檔案中。
2. 在文字編輯器中開啟 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name/ConnectDirectProcessDefinitions.xml` 檔案。
3. 建立 `<processSet>` 元素。
4. 在 `<processSet>` 元素內，建立 `<condition>` 元素。
5. 在 `<condition>` 元素內，建立一或多個元素來定義傳送要求必須符合的條件，以呼叫您在步驟 1 中定義的 Connect:Direct 程序。這些元素可以是 `<match>` 元素或 `<defined>` 元素。
 - 使用 `<match>` 元素可指定變數的值必須符合某個型樣。請建立具有下列屬性的 `<match>` 元素：
 - `variable` - 要比較其值的變數名稱。此變數為本質符號。如需相關資訊，請參閱 [第 554 頁的『與使用者定義的 Connect:Direct 程序搭配使用的替代變數』](#)。
 - `value` - 要與指定變數值進行比較的型樣。
 - 選用項目：`pattern` - `value` 屬性值所用的型樣類型。此型樣類型可以是 `wildcard` 或 `regex`。此為選用屬性，預設值為 `wildcard`。
 - 使用 `<defined>` 元素可指定必須為變數定義值。請建立具有下列屬性的 `<defined>` 元素：
 - `variable` - 必須為其定義值的變數名稱。此變數為本質符號。如需相關資訊，請參閱 [第 554 頁的『與使用者定義的 Connect:Direct 程序搭配使用的替代變數』](#)。
6. 在 `<processSet>` 元素內，建立 `<process>` 元素。

7. 在 <process> 元素內，建立 <transfer> 元素。

transfer 元素可指定 Connect:Direct 橋接器代理程式，在傳送過程中呼叫的 Connect:Direct 程序。請建立具有下列屬性的 <transfer> 元素：

- process- -您在步驟 1 中定義之 Connect:Direct 處理程序的位置。這個檔案的位置是以絕對路徑或相對於 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name` 目錄來指定。

結果

在搜尋條件相符項時，Connect:Direct 橋接器代理程式會從檔案開頭開始搜尋到檔案結尾。找到的第一個相符項，即為所用的相符項。

相關工作

第 118 頁的『[配置 Connect:Direct 橋接器](#)』

配置 Connect:Direct 橋接器，以在 Managed File Transfer 網路與 Connect:Direct 網路之間傳送檔案。

Connect:Direct 橋接器的元件為 Connect:Direct 節點，以及專門用來與該節點通訊的 Managed File Transfer 代理程式。此代理程式稱為 Connect:Direct 橋接器代理程式。

相關參考

第 495 頁的『[Connect:Direct 程序定義檔案格式](#)』

Connect:Direct 橋接器代理程式配置目錄中的 `ConnectDirectProcessDefinitions.xml` 檔案，會指定要在檔案傳送過程中啟動的使用者定義的 Connect:Direct 程序。

第 339 頁的『[fteCreateCDAgent \(建立 Connect:Direct 橋接器代理程式\)](#)』

`fteCreateCDAgent` 指令可建立 Managed File Transfer Agent 及其與 Connect:Direct 橋接器搭配使用的關聯配置。

在 Managed File Transfer 呼叫的 Connect:Direct 程序中使用本質符號變數

您可以在程序定義中使用本質符號變數，從 Managed File Transfer 傳送中呼叫使用者定義的 Connect:Direct 程序，並將資訊從該傳送中傳入 Connect:Direct 程序。

關於這項作業

此範例使用本質符號變數，將資訊從 Managed File Transfer 傳送中傳入使用者定義的 Connect:Direct 程序。如需 Managed File Transfer 使用的本質符號變數相關資訊，請參閱第 554 頁的『[與使用者定義的 Connect:Direct 程序搭配使用的替代變數](#)』。

在此範例中，檔案將從 Managed File Transfer Agent 傳送至 Connect:Direct 橋接器節點。該傳送的第一部分由 Managed File Transfer 執行。該傳送的第二部分由使用者定義的 Connect:Direct 程序執行。

程序

1. 建立使用本質符號變數的 Connect:Direct 程序。

```
%FTEPNAME PROCESS
  SNODE=%FTESNODE
  PNODEID=(%FTEPUSER,%FTEPPASS)
  SNODEID=(%FTESUSER,%FTESPASS)

COPY001 COPY
  FROM (
    FILE=%FTEFFILE
    DISP=%FTEFDISP
  )
  TO (
    FILE=%FTETFILE
    DISP=%FTETDISP
  )
PEND
```

2. 將此程序儲存至位於下列位置的文字檔：`MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent/Example.cdp`
3. 編輯 `ConnectDirectProcessDefinition.xml` 檔案，以包含呼叫您在步驟 1 中建立的 Connect:Direct 程序的規則。

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:cdprocess xmlns:tns="http://wmqfte.ibm.com/ConnectDirectProcessDefinitions"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/
  ConnectDirectProcessDefinitions ConnectDirectProcessDefinitions.xsd">

  <tns:processSet>
    <tns:condition>
      <tns:match variable="%FTESNODE" value="TOBERMORY" pattern="wildcard"/>
    </tns:condition>
    <tns:process>
      <tns:transfer process="Example.cdp"/>
    </tns:process>
  </tns:processSet>
</tns:cdprocess>
```

在此範例中，如果提交至 Connect:Direct 橋接器代理程式的傳送要求，以 TOBERMORY 作為其來源或目的地 Connect:Direct 節點，則會呼叫 Example.cdp Connect:Direct 程序。

- 提交滿足您在步驟 3 中於 ConnectDirectProcessDefinition.xml 檔案內所定義條件的檔案傳送要求。

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
```

在此範例中，目的地 Connect:Direct 節點為 TOBERMORY。此節點是該傳送中的次要節點，因此 %FTESNODE 的值設定為 TOBERMORY。此指令會比對在 ConnectDirectProcessDefinition.xml 檔案中設定的條件。

- Managed File Transfer 將來源檔案傳送至與 Connect:Direct 橋接器代理程式所處相同系統上的暫存位置。
- Connect:Direct 橋接器代理程式根據傳送要求中的資訊以及配置資訊，設定本質符號變數的值。本質符號變數會設定為下列值：
 - %FTEPNAME=*process_name* - 此值是 Connect:Direct 橋接器代理程式產生的 8 字元程序名稱。
 - %FTESNODE=TOBERMORY - 此值將根據 **fteCreateTransfer** 指令的 **-df** 參數來設定。
 - %FTEPUSER=*primary_node_user* - 此資訊將從 ConnectDirectCredentials.xml 檔案取得。
 - %FTEPPASS=*primary_node_user_password* - 此資訊將從 ConnectDirectCredentials.xml 檔案取得。
 - %FTESUSER=*secondary_node_user* - 此資訊將從 ConnectDirectCredentials.xml 檔案取得。
 - %FTESPASS=*secondary_node_user_password* - 此資訊將從 ConnectDirectCredentials.xml 檔案取得。
 - %FTEFFILE=*temporary_location* - 此值是與 Connect:Direct 橋接器代理程式所處相同系統上的檔案暫存位置。
 - %FTEFDISP=leave - 此值將根據 **fteCreateTransfer** 指令的 **-sd** 參數來設定。
 - %FTETFILE=/home/bulgaria/destination.txt - 此值將根據 **fteCreateTransfer** 指令的 **-df** 參數來設定。
 - %FTETDISP=overwrite - 此值將根據 **fteCreateTransfer** 指令的 **-de** 參數來設定。
- Connect:Direct 程序在 Connect:Direct 橋接器節點上啟動。Connect:Direct 會將檔案從 Connect:Direct 橋接器系統上的暫存位置，傳送至 Connect:Direct 節點 TOBERMORY 執行所在系統上的目的地 /home/bulgaria/destination.txt。

相關概念

第 200 頁的『透過檔案傳送要求提交使用者定義的 Connect:Direct 程序』

您可以提交傳送要求，以經由 Connect:Direct 橋接器代理程式進行傳送。在檔案傳送過程中，該代理程式會呼叫使用者定義的 Connect:Direct 程序。

相關參考

第 554 頁的『與使用者定義的 Connect:Direct 程序搭配使用的替代變數』

您可以使用 Managed File Transfer 專用的本質符號變數，定義要代入使用者定義的 Connect:Direct 程序的值。

使用 Connect:Direct 程序提交 Managed File Transfer 傳送要求

您可以透過 Connect:Direct 程序，將傳送要求提交至 Connect:Direct 橋接器代理程式。Managed File Transfer 提供可在 Connect:Direct 處理程序中從 **RUN TASK** 陳述式呼叫的指令。

Managed File Transfer 提供下列可與 Connect:Direct 程序搭配使用的指令：

ftetag

在 **ftebxfer** 或 **ftecxfer** 指令之前的步驟中指定此指令，可以建立傳送所需的審核資訊。此指令會將傳送的來源規格視為參數。如需來源規格的格式相關資訊，請參閱 [第 367 頁的『fteCreateTransfer: 啟動新的檔案傳送』](#)。

ftebxfer

如果要向其提交傳送要求的佇列管理程式，與提交指令的 Connect:Direct 節點位於相同系統上，請指定此指令以建立檔案傳送要求。此指令會採用與 **fteCreateTransfer** 指令相同的參數。如需這些參數的相關資訊，請參閱 [第 367 頁的『fteCreateTransfer: 啟動新的檔案傳送』](#)。此指令還具有以下一個其他參數：

-qmgrname

必要項目。要向其提交該指令的佇列管理程式名稱。

ftecxfer

如果要向其提交傳送要求的佇列管理程式，與提交指令的 Connect:Direct 節點位於不同系統上，請指定此指令以建立檔案傳送要求。此指令會採用與 **fteCreateTransfer** 指令相同的參數。如需參數的相關資訊，請參閱 [第 367 頁的『fteCreateTransfer: 啟動新的檔案傳送』](#)。此指令還具有以下三個其他參數：

-qmgrname

必要項目。要向其提交該指令的佇列管理程式名稱。

-connname

必要項目。要向其提交該指令的佇列管理程式的主機及埠（以 IBM MQ CONNAME 格式指定）。例如，host.example.com(1337)。

-channelname

選用項目。用來連接至要向其提交該指令的佇列管理程式的通道名稱。如果未指定此參數，則會使用預設值 SYSTEM.DEF.SVRCONN。

相關工作

第 204 頁的『使用「Connect:Direct 要求端」建立及提交呼叫 Managed File Transfer 的 Connect:Direct 程序』

「Connect:Direct 要求端」是一種圖形使用者介面，您可以用來建立及提交呼叫 Managed File Transfer 的 Connect:Direct 程序。

相關參考

第 557 頁的『範例：呼叫 MFT 指令的 Connect:Direct 程序檔案』

呼叫 Managed File Transfer **ftetag** 指令及 **ftecxfer** 指令的 Connect:Direct 程序檔案範例。

使用「Connect:Direct 要求端」建立及提交呼叫 Managed File Transfer 的 Connect:Direct 程序

「Connect:Direct 要求端」是一種圖形使用者介面，您可以用來建立及提交呼叫 Managed File Transfer 的 Connect:Direct 程序。

關於這項作業

此作業說明如何建立 Connect:Direct 處理程序，以呼叫 Managed File Transfer **ftecxfer** 指令或 **ftebxfer** 指令。如果要向其提交傳送要求的佇列管理程式，與提交指令的 Connect:Direct 節點位於不同的系統上，請使用 **ftecxfer** 指令。如果要向其提交傳送要求的佇列管理程式，與提交指令的

Connect:Direct 節點位於相同系統上，請使用 **ftebxfex** 指令。**ftexcxfex** 指令會與傳送來源代理程式的代理程式佇列管理程式建立用戶端連線。在呼叫 **ftexcxfex** 指令之前，您必須呼叫 **ftetag** 指令，並為其傳遞來源規格資訊。讓您可以使用與從 Managed File Transfer 起始的傳送相同的方式，來記載及審核程序。

程序

1. 啟動「Connect:Direct 要求端」。
2. 在畫面的節點標籤中，選取用作程序主要節點的 Connect:Direct 節點。
3. 選取檔案 > 新建 > 程序。「程序內容」視窗即會開啟。
4. 在名稱：欄位中，鍵入程序的名稱。
5. 從 **Snode** > 名稱：清單中選取次要節點。
6. 從 **Snode** > 作業系統：清單中選取次要節點的作業系統。
7. 選擇性的：在此視窗中完成您所需的任何進一步資訊。
8. 按一下確定。「程序內容」視窗即會關閉。
9. 建立執行 Managed File Transfer **ftetag** 指令的陳述式。
 - a) 在「程序」視窗中，用滑鼠右鍵按一下 **End** 陳述式。
 - b) 選取插入 > 執行作業。「執行作業陳述式」視窗即會開啟。
 - c) 在標籤：欄位中，鍵入 Tag。
 - d) 在選用參數或指令 欄位中，鍵入 `pgm(MQ_INSTALLATION_PATH/bin/ftetag) args(source_specification)`。如需 *source_specification* 格式的相關資訊，請參閱第 367 頁的『fteCreateTransfer：啟動新的檔案傳送』。
 - e) 按一下確定。「執行作業陳述式」視窗即會關閉。
10. 建立執行 Managed File Transfer **ftexcxfex** 或 **ftebxfex** 指令的陳述式。
 - a) 在「程序」視窗中，用滑鼠右鍵按一下 **End** 陳述式。
 - b) 選取插入 > 執行作業。「執行作業陳述式」視窗即會開啟。
 - c) 在標籤：欄位中，鍵入 Transfer。
 - d) 在選用參數或指令 欄位中，輸入 `pgm(MQ_INSTALLATION_PATH/bin/ftexcxfex) args(parameters)` 或 `pgm(MQ_INSTALLATION_PATH/bin/ftebxfex) args(parameters)`，視您選擇的指令而定。**ftexcxfex** 和 **ftebxfex** 指令所使用的參數，與 **fteCreateTransfer** 指令所使用的參數相同，但 **ftexcxfex** 和 **ftebxfex** 另有一些專用參數。如需相關資訊，請參閱第 367 頁的『fteCreateTransfer：啟動新的檔案傳送』及第 204 頁的『使用 Connect:Direct 程序提交 Managed File Transfer 傳送要求』。
 - e) 按一下確定。「執行作業陳述式」視窗即會關閉。
11. 選擇性的：建立您需要的任何其他陳述式。
12. 提交程序。
 - a) 在「程序」視窗中按一下滑鼠右鍵。
 - b) 選取提交。「Connect:Direct 連接」視窗即會開啟。
 - c) 輸入用於執行程序的使用者名稱及密碼。
 - d) 按一下確定。

相關概念

第 204 頁的『使用 Connect:Direct 程序提交 Managed File Transfer 傳送要求』

您可以透過 Connect:Direct 程序，將傳送要求提交至 Connect:Direct 橋接器代理程式。Managed File Transfer 提供可在 Connect:Direct 處理程序中從 **RUN TASK** 陳述式呼叫的指令。

從 IBM Integration Bus 使用 MFT

您可以利用 FTEOutput 及 FTEInput 節點使用 IBM Integration Bus 的 Managed File Transfer。

- 使用 FTEInput 節點，在應用 Managed File Transfer 的網路上傳送檔案，然後在 Integration Bus 流程期間處理該檔案。
- 使用 FTEOutput 節點，將 Integration Bus 流程輸出檔案，傳輸到網路上的其他位置。

與分配管理系統代理程式來回傳送檔案的代理程式，可以是任何層次的 Managed File Transfer。

如需相關資訊，請參閱 [IBM Integration Bus 產品說明文件](#)。

MFT 回復及重新啟動

如果代理程式或佇列管理程式因故無法使用，例如因為電源或網路故障，在這些情況下，Managed File Transfer 會回復，如下所示：

- 一般而言，如果檔案在傳送時中發生問題，在解決問題後，Managed File Transfer 會回復並重新啟動該檔案傳送。
- 如果傳送過程中的檔案被刪除或變更，而代理程式或佇列管理程式在當時無法使用，則傳送會失敗，且傳送日誌中會出現一個訊息提供失敗的詳細資料。
- 如果代理程式程序在檔案傳送期間失敗，則重新啟動代理程式後，就會繼續傳送。
- 如果代理程式與其代理程式佇列管理程式的連線中斷，代理程式會一直等待嘗試重新連接至佇列管理程式。當代理程式順利重新連接至其佇列管理程式時，現行傳送會繼續進行。
- 如果代理程式因故而停止，則任何與代理程式相關聯的資源監視器都會停止輪詢。當代理程式回復時，監視器也會重新啟動，資源輪詢會回復。
- 對於來源處置為刪除的檔案傳送，如果在所有資料從來源代理程式傳送至目的地代理程式之後進行回復，則來源檔會在刪除之前解除鎖定。這個解除鎖定動作表示來源檔在檔案刪除之前可能遭到修改。因此會將刪除來源檔視為不安全的作業，而顯示下列警告：

```
BFGTR0075W: The source file has not been deleted because it is possible that the source file was modified after the source file was transferred.
```

在此情況下，請驗證來源檔的內容是否未經修改，然後再手動刪除來源檔。

您可以在「IBM MQ Explorer」中檢查傳送的狀態。如果任何傳送顯示為 Stalled，則您可能需要採取更正動作，因為停滯狀態表示代理程式或傳送所涉及的兩個代理程式之間發生問題。

相關概念

第 206 頁的『回復中檔案傳送的逾時選項』

您可以設定時間量（以秒為單位），在此期間內，來源代理程式會一直嘗試回復已停止的檔案傳送。如果代理程式達到重試間隔的逾時值時傳送未順利完成，則傳送失敗。

V 9.0.1 回復中檔案傳送的逾時選項

您可以設定時間量（以秒為單位），在此期間內，來源代理程式會一直嘗試回復已停止的檔案傳送。如果代理程式達到重試間隔的逾時值時傳送未順利完成，則傳送失敗。

在 IBM MQ 9.0.1 之前，Managed File Transfer 來源代理程式的預設行為是一直嘗試回復已停止的傳送，直到傳送成功為止。因為新參數為選用項目，您不一定要設定它，傳送會遵循預設行為。

從 IBM MQ 9.0.1 開始，您可以指定下列其中一個選項：

-1

代理程式繼續嘗試回復已停止的傳送，直到傳送順利完成為止。使用此選項相當於代理程式在未設定此內容時的預設行為。

0

一旦進入回復，代理程式即停止檔案傳送。

>0

代理程式繼續嘗試回復已停止的傳送，直至達到已指定的正整數值所設定的時間量（以秒為單位）為止。值 21600 指出代理程式從進入回復的 6 小時內會一直嘗試回復傳送。此參數的最大值為 999999999。

為一個來源代理程式的所有傳送設定回復重試逾時

若要設定適用於來源代理程式的所有傳送的回復逾時，請將此參數和值配對新增至 `agent.properties` 檔案中。

在此範例中，將傳送回復逾時值設為 `-1`，表示代理程式會繼續嘗試回復已停止的傳送，直到傳送順利完成為止：

```
transferRecoveryTimeout=-1
```

將傳送回復逾時值設為 `0`，表示代理程式會在進入回復後立即將傳送標示為失敗：

```
transferRecoveryTimeout=0
```

將傳送回復逾時值設為值 `21600`，表示將傳送標示為失敗之前的 6 小時內，代理程式會一直重試已停止的傳送：

```
transferRecoveryTimeout=21600
```

設定或置換個別傳送的回復重試逾時

在執行下列動作時，您可以為個別傳送設定回復重試逾時參數：

- 在 Managed File Transfer 指令行介面或 IBM MQ Explorer 中使用 `fteCreateTransfer` 指令來建立傳送
- 在 Managed File Transfer 指令行介面或 IBM MQ Explorer 中使用 [第 354 頁的『fteCreateTemplate: 建立新的檔案傳送範本』](#) 指令建立傳送範本
- 使用 `fteCreateMonitor` Managed File Transfer 指令行介面或 IBM MQ Explorer 來建立監視器
- 使用 `fte:filecopy` 或 `fte:filemove` Ant 作業來複製或移動檔案

將傳送的 `-rt` 值設為 `-1` 等同於預設行為，即回復會繼續執行，直到傳送順利完成或使用者手動取消為止，例如

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

如需在建立新傳送時使用傳送回復逾時參數的相關資訊，請參閱 [fteCreateTransfer](#) 指令。

將 `-rt` 參數值設為 `0`，表示如果使用此範本起始的傳送停止，則傳送將立即失敗，而且不會嘗試進行回復，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
```

如需在建立新傳送範本時使用傳送回復逾時參數的相關資訊，請參閱 [fteCreateTemplate](#) 指令。

您可以使用 IBM MQ Explorer，為傳送設定回復逾時參數和值。如需使用 IBM MQ Explorer 來配置傳送的相關資訊，請參閱 [第 133 頁的『啟動新的檔案傳送』](#) 及 [第 163 頁的『使用 IBM MQ Explorer 建立檔案傳送範本』](#)。

您還可以執行下列作業來設定回復逾時：包括 `transferRecoveryTimeout` 選項和值，並使用 `fte:filecopy` 或 `fte:filemove` 元素，以利用 Ant 作業來移動或複製檔案，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>
```

當文件複製作業已起始且傳送進入回復時，傳送將立即停止，而不嘗試進行回復。設定 `transferRecoveryTimeout` 選項來搭配 `fte:filecopy` 或 `fte:filemove` 時，會置換 `agent.properties` 檔案中設定的值。如果未設定 `transferRecoveryTimeout` 值來搭配 `fte:filecopy` 或 `fte:filemove`，將使用來自 `agent.properties` 檔案的

transferRecoveryTimeout 參數的值。如需相關資訊，請參閱 [第 735 頁的『fte:filecopy Ant 作業』](#) 及 [第 739 頁的『fte:filemove Ant 作業』](#)。

處理回復逾時優先順序

透過指令行介面引數為建立傳送、範本或監視器指令指定的傳送回復逾時值 (包括在 IBM MQ Explorer 精靈中設定選項)，或在 **fte:filespec** 巢狀元素中指定的傳送回復逾時值，其優先順序高於在來源代理程式的 `agent.properties` 檔案中為 **transferRecoveryTimeout** 參數指定的值。例如，對於指令

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

(其開頭沒有 **-rt** 參數和值配對)，來源代理程式 AGENT1 將檢查 `agent.properties` 檔案的 **transferRecoveryTimeout** 值，以決定回復逾時行為。

如果在 `agent.properties` 檔案中，**transferRecoveryTimeout** 未設定或是設為 `-1`，則代理程式會遵循預設行為，並嘗試回復傳送，直到傳送順利完成為止。

透過 Managed File Transfer 指令行介面 (例如，使用 **fteCreateTransfer** 指令) 指定回復逾時選項 **-rt** 時，此值優先於 `agent.properties` 檔案中的值，並用作傳送的設定。

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

處理回復逾時計數器

回復逾時計數器是在傳送進入回復狀態時啟動。傳送日誌訊息會透過主題字串 `Log/agent_name/transfer_ID` 發佈至 SYSTEM.FTE 主題，以指出傳送狀態已變更為回復中，以及狀態變更時的來源代理程式時鐘時間。如果在設定的重試間隔內回復傳送，但未達到回復逾時 (`counter <= recovery timeout`)，則計數器將重設為 0，並準備在傳送進入回復時重新啟動。

如果計數器達到為回復逾時設定的最大值 (`counter == recovery timeout`)，則回復傳送將停止，且來源代理程式會報告此傳送失敗。因為傳送達到回復逾時而導致傳送失敗的這種類型，會以新訊息碼 RECOVERY TIMEOUT (69) 指出。另一則傳送日誌訊息會透過主題字串 `Log/agent_name/transfer_ID` 發佈至 SYSTEM.FTE 主題，以指出傳送失敗，並包含新訊息、新回覆碼及來源代理程式的事件日誌。當回復期間發生下列任何事件時，來源代理程式的事件日誌中將更新訊息。

- 當回復逾時參數設為大於 `-1` 的值時，Managed File Transfer 會進入回復。代理程式的事件日誌將更新，以指出 **TransferId** 的回復計時器開始，以及來源代理程式在起始回復逾時處理之前等待的時間量。
- 當繼續回復 Managed File Transfer 時，來源代理程式的事件日誌中將更新訊息，以指出處於回復的 **TransferId** 會繼續回復。
- 當回復 Managed File Transfer 已逾時，來源代理程式的事件日誌會更新，以指出 **TransferId** 因回復逾時而導致回復失敗。

這些日誌訊息讓使用者 (訂閱者和日誌程式) 能夠識別因為傳送回復逾時而失敗的傳送。

回復逾時的計數器一律位於來源代理程式。但是，如果目的地代理程式無法及時接收到來源代理程式的資訊，它會向來源代理程式傳送要求，將此傳送置於回復狀態。對於設定了回復逾時選項的傳送，來源代理程式會在接收到目的地代理程式的要求時啟動回復逾時計數器。

對於不使用回復逾時選項的傳送、失敗的傳送和部分完成的傳送，仍需要手動處理。

對於向多個檔案發出單一傳送要求的傳送集，其中某些檔案已順利完成，但有一個檔案僅局部完成，仍會標示為失敗的傳送，因為傳送沒有按照預期完成。在傳送局部完成的檔案時，來源代理程式可能已逾時。

確定目的地代理程式和檔案伺服器已經準備妥當，並處於可接受檔案傳送的狀態。

您必須再次為整個集發出傳送要求，但為了避免因初次傳送嘗試後還有一些檔案留在目的地而導致有問題，發出新要求時您可以指定 **改寫現有檔案** 選項。這可確保前次傳送嘗試的不完整檔案集在新的傳送中予以清除，然後再重新寫入至目的地。

追蹤資料和訊息

會基於診斷目的而包含追蹤點。將記載回復逾時值、重試間隔開始時間、回復期間和計數器重設以及傳送是否逾時和失敗。萬一發生問題或出現非預期的行為，您可以收集來源代理程式輸出日誌和追蹤檔案，並在 IBM 支援人員要求時提供它們，以協助進行疑難排解。

當傳送進入回復狀態 (BFGTR0081I)、因為逾時而終止回復 (BFGSS0081E) 以及從回復狀態中繼續 (BFGTR0082I) 時，會有訊息通知使用者。

BFGTR0001 - BFGTR9999

BFGTR0081I

BFGTR0082I

BFGSS0001 - BFGSS9999

BFGSS0081E

為 MFT 開發應用程式

指定要使用 MFT 執行的程式

您可以在執行 Managed File Transfer Agent 的系統上執行程式。在檔案傳送要求過程中，您可以指定程式在傳送開始前或結束後執行。此外，您也可以提交受管理的呼叫要求，以啟動不在檔案傳送要求過程中的程式。

您可以指定讓程式在下列五種情況下執行：

- 在傳送要求過程中，於傳送開始前在來源代理程式上執行
- 在傳送要求過程中，於傳送開始前在目的地代理程式上執行
- 在傳送要求過程中，於傳送完成後在來源代理程式上執行
- 在傳送要求過程中，於傳送完成後在目的地代理程式上執行
- 不在傳送要求過程中。您可以向代理程式提交執行程式的要求。這種情況有時稱為受管理的呼叫。

使用者結束程式和程式呼叫的呼叫順序如下：

```
- 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.
```

附註：

1. 只有在傳送順利完成或局部順利完成時，才會執行 **DestinationTransferEndExits**。
2. 只有在傳送順利完成或局部順利完成時，才會執行 **postDestinationCall**。
3. **SourceTransferEndExits** 是針對成功、局部成功或失敗的傳送而執行。
4. 只有在下列情況下，才會呼叫 **postSourceCall**：
 - 傳送未取消。
 - 有成功或部分成功的結果。
 - 已順利執行任何後置目的地傳送程式。

您可以透過數種方式來指定想要執行的程式。這些選項如下：

使用 Apache Ant 作業

使用其中一個 **fte:filecopy**、**fte:filemove** 或 **fte:call** Ant 作業來啟動程式。使用 Ant 作業可讓您透過 **fte:presrc**、**fte:predst**、**fte:postdst**、**fte:postsrc** 及 **fte:command** 巢狀元素，在上述五種的任一情況中指定程式。如需相關資訊，請參閱第 749 頁的『程式呼叫巢狀元素』。

編輯檔案傳送要求訊息

您可以編輯傳送要求所產生的 XML。使用此方法可讓您將 **preSourceCall**、**postSourceCall**、**preDestinationCall**、**postDestinationCall** 及 **managedCall** 元素新增至 XML 檔，在上述五種的任何一種情況中執行程式。然後，這個修改過的 XML 檔即可作為新檔案傳送要求（例如使用 **fteCreateTransfer -td** 參數）的傳送定義。如需相關資訊，請參閱第 683 頁的『MFT 代理程式呼叫要求訊息範例』。

使用 **fteCreateTransfer** 指令

您可以使用 **fteCreateTransfer** 指令來指定要啟動的程式。您可以使用此指令來指定在前四種情況（在傳送要求過程中）下執行程式，但無法啟動受管理的呼叫。如需參數使用的相關資訊，請參閱第 367 頁的『fteCreateTransfer: 啟動新的檔案傳送』。如需使用此指令的範例，請參閱第 728 頁的『使用 fteCreateTransfer 啟動程式的範例』。

相關參考

第 306 頁的『commandPath MFT 內容』

使用 **commandPath** 內容來指定 Managed File Transfer 可以從中執行指令的位置。在設定此內容時請特別小心，因為可以從遠端用戶端系統有效地呼叫其中一個指定的 **commandPaths** 中的任何指令，從而可以將指令傳送至代理程式。

受管理呼叫

Managed File Transfer (MFT) 代理程式通常用來傳送檔案或訊息。這些稱為受管理傳送。代理程式也可以用來執行指令、Script 或 JCL，而不需要傳送檔案或訊息。此功能稱為受管理呼叫。

受管理呼叫要求可以透過數種方式提交給代理程式：

- 使用 **fte: call** Ant 作業。
- 使用執行指令或 Script 的作業 XML 來配置資源監視器。如需相關資訊，請參閱 [配置監視器作業以啟動指令及 Script](#)。
- 將 XML 訊息直接放入代理程式的指令佇列中。如需受管理呼叫 XML 綱目的詳細資料，請參閱 [檔案傳送要求訊息格式](#)。

對於受管理呼叫，必須在代理程式內容 **commandPath** 中指定包含正在執行之指令或 Script 的目錄。

受管理呼叫無法執行位於未在代理程式的 **commandPath** 中指定之目錄中的指令或 Script。這是為了確保代理程式不會執行任何惡意程式碼。

此外，您也可以代理程式上啟用權限檢查，以確保只容許授權使用者提交受管理呼叫要求。如需此作業的相關資訊，請參閱 [限制 MFT 代理程式動作的使用者權限](#)。

在受管理呼叫中呼叫的指令、Script 或 JCL，會以代理程式所監視的外部程序來執行。當處理程序結束時，受管理呼叫會完成，且處理程序的回覆碼可供呼叫 **fte: Ant call** Ant 作業的代理程式或 Script 使用。

如果受管理呼叫是由 **fte: call** Ant 作業啟動，您的 Ant Script 可以檢查回覆碼值，以判斷受管理呼叫是否成功。

對於所有其他類型的受管理呼叫，您可以指定應使用哪些回覆碼值來指出受管理呼叫已順利完成。當外部處理程序完成時，代理程式會將來自處理程序的回覆碼與這些回覆碼進行比較。

註：因為受管理呼叫以外部處理程序執行，所以一旦啟動就無法取消它們。

受管理呼叫及來源傳送插槽

代理程式包含許多來源傳送插槽，如代理程式內容 **maxSourceTransfers** 所指定，如 [進階代理程式內容：傳送限制](#) 中所述。

每當執行受管理呼叫或受管理傳送時，它們會佔用來源傳送插槽。當受管理呼叫或受管理轉接完成時，會釋放插槽。

當代理程式收到新的受管理呼叫或受管理傳送要求時，如果所有來源傳送時段都在使用中，則代理程式會將要求排入佇列，直到時段變成可用為止。

如果受管理呼叫啟動受管理傳送 (例如, 如果受管理呼叫執行 Ant Script, 且該 Ant Script 使用 `fte: filecopy` 或 `fte: filemove` 作業來傳送檔案), 則需要兩個來源傳送插槽:

- 一個用於受管理傳送
- 一個用於受管理呼叫

在此狀況下, 請務必注意, 如果受管理傳送需要很長時間才能完成或進入回復, 則兩個來源傳送插槽會被佔用, 直到受管理傳送完成或取消為止。

因此, 您應該考量受管理呼叫的設計, 以確保它不會長時間佔用來源傳送時段。

搭配使用 Apache Ant 與 MFT

您可以利用 Managed File Transfer 提供的作業, 將檔案傳送功能整合到 Apache Ant 工具中。

您可以使用 `fteAnt` 指令, 在您已配置的 Managed File Transfer 環境中執行 Ant 作業。您可以使用 Ant Script 中的檔案傳送 Ant 作業, 來協調直譯式 Script 語言的複雜檔案傳送作業。

如需 Apache Ant 的相關資訊, 請參閱 Apache Ant 專案網頁: <https://ant.apache.org/>

相關概念

第 211 頁的『將 Ant Script 與 MFT 搭配使用入門』

將 Ant Script 與 Managed File Transfer 搭配使用可讓您從直譯式 Scripting 語言協調複雜的檔案傳送作業。

相關參考

第 731 頁的『MFT 提供的 Ant 作業』

Managed File Transfer 提供了許多 Ant 作業, 您可用於存取檔案傳送功能。

第 318 頁的『fteAnt: 在 MFT 中執行 Ant 作業』

`fteAnt` 指令可在提供 Managed File Transfer Ant 作業的環境中執行 Ant Script。

第 212 頁的『MFT 的 Ant 作業範例』

Managed File Transfer 安裝隨附多個 Ant Script 範例。這些範例位於 `MQ_INSTALLATION_PATH/mqft/samples/fteant` 目錄中。每一個範例 Script 包含一個 `init` 目標, 請編輯 `init` 目標中的內容集, 以搭配您的配置來執行這些 Script。

將 Ant Script 與 MFT 搭配使用入門

將 Ant Script 與 Managed File Transfer 搭配使用可讓您從直譯式 Scripting 語言協調複雜的檔案傳送作業。

Ant Script

Ant Script (或建置檔) 是可定義一個以上目標的 XML 文件。這些目標包含要執行的作業元素。Managed File Transfer 提供了您可用於將檔案傳送功能整合至 Apache Ant 的作業。如果要瞭解 Ant Script, 請參閱 Apache Ant 專案網頁: <https://ant.apache.org/>

產品安裝的目錄 `MQ_INSTALLATION_PATH/mqft/samples/fteant` 中, 提供了使用 Managed File Transfer 作業的 Ant Script 的範例。

在通訊協定橋接器代理程式上, Ant Script 在通訊協定橋接器代理程式系統上執行。這些 Ant Script 無法直接存取 FTP 或 SFTP 伺服器上的檔案。

名稱空間

名稱空間用於區分檔案傳送 Ant 作業與可能共用相同名稱的其他 Ant 作業。您可以在 Ant Script 的專案標籤中定義名稱空間。

```
<?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>
```

屬性 `xmlns:fte="antlib:com.ibm.wmqfte.ant.taskdefs"` 會告知 Ant 在程式庫 `com.ibm.wmqfte.ant.taskdefs` 中尋找字首為 `fte` 的作業定義。

您不需要使用 `fte` 作為名稱空間字首；您可以使用任何值。所有 Ant Script 範例使用名稱空間字首 `fte`。

執行 Ant Script

若要執行包含檔案傳送 Ant 作業的 Ant Script，請使用 **fteAnt** 指令。例如：

```
fteAnt -file ant_script_location/ant_script_name
```

如需相關資訊，請參閱 [第 318 頁的『fteAnt: 在 MFT 中執行 Ant 作業』](#)。

回覆碼

檔案傳送 Ant 作業傳回與 Managed File Transfer 指令相同的回覆碼。如需相關資訊，請參閱 [第 284 頁的『MFT 的回覆碼』](#)。

相關參考

[第 731 頁的『MFT 提供的 Ant 作業』](#)

Managed File Transfer 提供了許多 Ant 作業，您可用於存取檔案傳送功能。

[第 318 頁的『fteAnt: 在 MFT 中執行 Ant 作業』](#)

fteAnt 指令可在提供 Managed File Transfer Ant 作業的環境中執行 Ant Script。

[第 212 頁的『MFT 的 Ant 作業範例』](#)

Managed File Transfer 安裝隨附多個 Ant Script 範例。這些範例位於 `MQ_INSTALLATION_PATH/mqft/samples/fteant` 目錄中。每一個範例 Script 包含一個 `init` 目標，請編輯 `init` 目標中的內容集，以搭配您的配置來執行這些 Script。

MFT 的 Ant 作業範例

Managed File Transfer 安裝隨附多個 Ant Script 範例。這些範例位於 `MQ_INSTALLATION_PATH/mqft/samples/fteant` 目錄中。每一個範例 Script 包含一個 `init` 目標，請編輯 `init` 目標中的內容集，以搭配您的配置來執行這些 Script。

電子郵件

email 範例示範如何使用 Ant 作業來傳送檔案，以及傳送失敗時如何傳送電子郵件至指定的電子郵件位址。Script 會使用 Managed File Transfer ping 作業來確認來源及目的地代理程式為作用中，且能夠處理傳送。如果兩個代理程式都在作用中，則 Script 會使用 Managed File Transfer filecopy 作業，在來源與目的地代理程式之間傳送檔案，但不會刪除原始檔案。如果傳送失敗，Script 會使用標準 Ant email 作業傳送電子郵件，包含失敗的相關資訊。

hub

hub 範例由兩個 Script 組成: `hubcopy.xml` 和 `hubprocess.xml`。 `hubcopy.xml` Script 顯示如何使用 Ant Script 來建置「hub 和輪輻」樣式拓撲。在此範例中，兩個檔案從輻射機器上執行的代理程式傳送至主軸機器上執行的代理程式。兩個檔案同時傳送，當傳送完成時，主軸機器上會執行 `hubprocess.xml` Ant Script 來處理檔案。如果兩個檔案的傳送都正確，Ant Script 會連結檔案的內容。如果檔案未正確傳送，Ant Script 會刪除任何已傳送的檔案資料，以完成清除工作。您必須將 `hubprocess.xml` Script 放置在主軸代理程式的指令路徑，此範例才能正確運作。如需設定代理程式指令路徑的相關資訊，請參閱 [commandPath](#)。

librarytransfer (僅適用於 IBM i 平台)



IBM i librarytransfer 範例示範如何使用 Ant 作業將某個 IBM i 系統上的 IBM i 程式庫傳送至第二個 IBM i 系統。

IBM i IBM i 上的 IBM WebSphere MQ File Transfer Edition 7.0.2 不包括對原生 IBM i 程式庫物件傳送是直接支援。librarytransfer 範例使用 IBM i 支援的原生儲存檔與 Managed File Transfer 中可用的預先定義 Ant 作業，在兩個 IBM i 系統之間傳送原生程式庫物件。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 執行此範例之前，您需要完成 librarytransfer.xml 檔中說明的一些配置。您在兩部 IBM i 機器上也必須有實際運作的 Managed File Transfer 環境。設定必須包含第一部 IBM i 機器上執行的來源代理程式及第二部 IBM i 機器上執行的目的地代理程式。兩個代理程式必須能夠彼此通訊。

IBM i librarytransfer 範例由下列三個檔案組成：

- librarytransfer.xml
- librarysave.sh (<presrc> executable script)
- libraryrestore.sh (<postdst> executable script)

範例檔案位於下列目錄：`/QIBM/ProdData/WMQFTE/V7/samples/fteant/ibmi/librarytransfer`

IBM i 若要執行此範例，使用者必須完成下列步驟：

1. 啟動 QShell 階段作業。在 IBM i 指令視窗，鍵入：STRQSH
2. 切換至 bin 目錄，如下所示：

```
cd /QIBM/ProdData/WMQFTE/V7/bin
```

3. 完成必要配置之後，使用下列指令執行範例：

```
fteant -f /QIBM/ProdData/WMQFTE/V7/samples/fteant/ibmi/librarytransfer/librarytransfer.xml
```

physicalfiletransfer (僅適用於 IBM i 平台)

IBM i physicalfiletransfer 範例示範如何使用 Ant 作業，將來源實體或資料庫檔案從某個 IBM i 系統上的程式庫傳送至第二個 IBM i 系統上的程式庫。

IBM i IBM i 上的 IBM WebSphere MQ File Transfer Edition 7.0.2 不包括直接支援 IBM i 上原生「來源實體」或「資料庫」檔案的傳送。physicalfiletransfer 範例使用 IBM i 支援的原生儲存檔與 Managed File Transfer 中可用的預先定義 Ant 作業，在兩個 IBM i 系統之間傳送完整的來源實體及資料庫檔案。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 執行此範例之前，您必須完成 physicalfiletransfer.xml 檔中說明的一些配置。您在兩個 IBM i 系統上也必須有實際運作的 Managed File Transfer 環境。設定必須包含第一個 IBM i 系統上執行的來源代理程式及第二個 IBM i 系統上執行的目的地代理程式。兩個代理程式必須能夠彼此通訊。

IBM i physicalfiletransfer 範例由下列三個檔案組成：

- physicalfiletransfer.xml

- `physicalfilesave.sh` (<presrc> executable script)
- `physicalfilerestore.sh` (<postdst> executable script)

範例檔案位於下列目錄：`/QIBM/ProdData/WMQFTE/V7/samples/fteant/ibmi/physicalfiletransfer`

IBM i 若要執行此範例，使用者必須完成下列步驟：

1. 啟動 QShell 階段作業。在 IBM i 指令視窗，鍵入：`STRQSH`
2. 切換至 `bin` 目錄，如下所示：

```
cd /QIBM/ProdData/WMQFTE/V7/bin
```

3. 完成必要配置之後，使用下列指令執行範例：

```
fteant -f /QIBM/ProdData/WMQFTE/V7/samples/fteant/ibmi/physicalfiletransfer/physicalfiletransfer.xml
```

逾時

`timeout` 範例示範如何使用 Ant 作業來嘗試檔案傳送，以及在所費時間超過指定的逾時值時取消傳送。Script 會使用 Managed File Transfer `filecopy` 作業來起始檔案傳送。此傳送的結果會延遲。Script 使用 Managed File Transfer 第 732 頁的『`fte:awaitoutcome` Ant 作業』作業，依指定的秒數等待傳送完成。如果傳送未在指定時間內完成，則使用 Managed File Transfer 第 735 頁的『`fte:cancel` Ant 作業』作業來取消檔案傳送。

vsamtransfer

z/OS

z/OS `vsamtransfer` 範例示範如何使用 Ant 作業，透過 Managed File Transfer 從 VSAM 資料集傳送至另一個 VSAM 資料集。Managed File Transfer 目前不支援傳送 VSAM 資料集。範例 Script 會使用 `presrc` 巢狀元素來呼叫執行檔 `datasetcopy.sh`，將 VSAM 資料記錄卸載至循序資料集。Script 會使用 Managed File Transfer 第 739 頁的『`fte:filemove` Ant 作業』作業，將循序資料集從來源代理程式傳送至目的地代理程式。然後，Script 使用 `postdst` 巢狀元素來呼叫 `loadvsam.jcl` Script。此 JCL Script 會將傳送的資料集記錄載入目的地 VSAM 資料集。此範例使用 JCL 進行目的地呼叫，以示範此語言選項。使用第二個 Shell Script 也能獲得相同結果。

z/OS 此範例不要求來源及目的地資料集必須為 VSAM。如果來源及目的地資料集是相同類型，則此範例適用於任何資料集。

z/OS 您必須將 `datasetcopy.sh` Script 放置在來源代理程式的指令路徑，將 `loadvsam.jcl` Script 放置在目的地代理程式的指令路徑，此範例才能正確運作。如需設定代理程式指令路徑的相關資訊，請參閱 [commandPath](#)。

zip

`zip` 範例由兩個 Script 組成：`zip.xml` 及 `zipfiles.xml`。此範例示範如何在執行檔案傳送移動作業之前，使用 Managed File Transfer 第 739 頁的『`fte:filemove` Ant 作業』作業內的 `presrc` 巢狀元素來執行 Ant Script。`zip.xml` Script 中的 `presrc` 巢狀元素所呼叫的 `zipfiles.xml` Script 會壓縮目錄的內容。`zip.xml` Script 會傳送壓縮檔。此範例要求 `zipfiles.xml` Ant Script 必須位於來源代理程式的指令路徑。這是因為 `zipfiles.xml` Ant Script 包含目標，可用來壓縮來源代理程式上的目錄內容。如需設定代理程式指令路徑的相關資訊，請參閱 [commandPath](#)。

利用使用者結束程式自訂 MFT

您可以使用專屬程式（即所謂的使用者結束常式）來自訂 Managed File Transfer 的特性。

Managed File Transfer 在程式碼提供位置點，讓 Managed File Transfer 能夠將控制權傳遞給您所撰寫的程式（使用者結束程式）。這些點即所謂的使用者結束點。當您的程式完成其工作時，Managed File Transfer 可以回復控制權。您未必要使用任何使用者結束程式，但如果您想要延伸及自訂 Managed File Transfer 系統的功能來符合特定需求，則使用者結束程式會很有幫助。

在檔案傳送處理期間，有兩個位置點可供您在來源系統呼叫使用者結束程式，在檔案傳送處理期間，還有兩個位置點可供您在目的地系統呼叫使用者結束程式。下表彙總這些使用者結束點，以及您要使用結束點時必須實作的 Java 介面。

表 23: 來源端和目的地端結束點及 Java 介面的摘要	
結束點	要實作的 Java 介面
來源端結束點:	
在整個檔案傳送開始之前	SourceTransferStartExit.java
在整個檔案傳送完成後	SourceTransferEndExit.java
目的地端結束點:	
在整個檔案傳送開始之前	DestinationTransferStartExit.java
在整個檔案傳送完成後	DestinationTransferEndExit.java

依下列順序呼叫使用者結束程式：

1. SourceTransferStartExit
2. DestinationTransferStartExit
3. DestinationTransferEndExit
4. SourceTransferEndExit

SourceTransferStartExit 及 DestinationTransferStartExit 結束程式所做的變更會延伸到後續的結束程式成為輸入。比方說，例如 SourceTransferStartExit 結束程式修改傳送 meta 資料，此變更會反映在其他結束程式的輸入傳送 meta 資料中。

使用者結束程式和程式呼叫的呼叫順序如下：

```

- 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.

```

附註：

1. 只有在傳送順利完成或局部順利完成時，才會執行 **DestinationTransferEndExits**。
2. 只有在傳送順利完成或局部順利完成時，才會執行 **postDestinationCall**。
3. **SourceTransferEndExits** 是針對成功、局部成功或失敗的傳送而執行。
4. 只有在下列情況下，才會呼叫 **postSourceCall**：
 - 傳送未取消。
 - 有成功或部分成功的結果。
 - 已順利執行任何後置目的地傳送程式。

建置使用者結束程式

建置使用者結束程式的介面包含在 `MQ_INSTALL_DIRECTORY/mqft/lib/com.ibm.wmqfte.exitroutines.api.jar` 中。在建置結束程式時，您必須將這個 .jar 檔併入類別路徑中。若要執行結束程式，請將結束程式解壓縮成 .jar 檔，然後將這個 .jar 檔放到下一節所說明的目錄中。

使用者結束程式位置

您可以在兩個可能的位置儲存使用者結束常式：

- `exits` 目錄。每一個代理程式目錄下都有一個 `exits` 目錄。例如：
`var\mqm\mqft\config\QM_JUPITER\agents\AGENT1\exits`
- 您可以設定 `exitClassPath` 內容來指定替代位置。如果 `exits` 目錄及 `exitClassPath` 所設定的類別路徑中都有結束程式類別，則以 `exits` 目錄中的類別為優先，這表示如果這兩個位置有相同名稱的類別，以 `exits` 目錄中的類別為優先。

配置代理程式使用使用者結束程式

可設定四項代理程式內容來指定代理程式呼叫的使用者結束程式。這些代理程式內容是 `sourceTransferStartExitClasses`、`sourceTransferEndExitClasses`、`destinationTransferStartExitClasses` 及 `destinationTransferEndExitClasses`。如需如何使用這些內容的相關資訊，請參閱第 759 頁的『使用者結束程式的 MFT 代理程式內容』。

在通訊協定橋接器代理程式上執行使用者結束程式

當來源代理程式呼叫結束程式時，它會將一份要傳送的來源項目清單傳遞給結束程式。對於一般代理程式，這是一份完整檔名清單。因為檔案應該是本端檔案（或可透過裝載來存取），所以結束程式能夠存取它並對它進行加密。

但是，對於通訊協定橋接器代理程式，清單中的項目為下列格式：

```
"<file server identifier>:<fully-qualified file name of the file on the remote file server>"
```

對於清單中的每一個項目，結束程式都必須先連接至檔案伺服器（使用 FTP、FTPS 或 SFTP 通訊協定），下載檔案，在本端加密檔案，然後將已加密的檔案上傳回檔案伺服器。

在 Connect:Direct 橋接器代理程式上執行使用者結束程式

您無法在 Connect:Direct 橋接器代理程式上執行使用者結束程式。

MFT 來源及目的地使用者結束程式

目錄分隔字元

無論您在 `fteCreateTransfer` 指令或「IBM MQ Explorer」中指定的目錄分隔字元為何，原始檔規格中的目錄分隔字元一律使用正斜線 (/) 字元來表示。在撰寫結束程式時，您必須考慮到這一點。比方說，如果想要檢查下列原始檔是否存在：`c:\a\b.txt`，並且已使用 `fteCreateTransfer` 指令或 IBM MQ Explorer 指定此原始檔，請注意，該檔名實際上是儲存為 `c:/a/b.txt`。因此，如果以 `c:\a\b.txt` 的原始字串進行搜尋，將找不到相符項。

來源端結束點

在整個檔案傳送開始之前

當傳送要求為擱置傳送清單中的下一個項目，並且傳送即將開始時，來源代理程式會呼叫此結束程式。

此結束點的使用範例包括：使用外部指令分階段將檔案傳送至代理程式具有讀/寫存取權限的目錄，或者重新命名目的地系統上的檔案。

將下列引數傳遞至此結束程式：

- 來源代理程式名稱
- 目的地代理程式名稱
- 環境 meta 資料
- 傳送 meta 資料
- 檔案規格（包括檔案 meta 資料）

從此結束程式傳回的資料如下所示：

- 更新的傳送 meta 資料。可以新增、修改及刪除項目。
- 更新的檔案規格清單，由來源檔案名稱及目的地檔名稱配對組成。可以新增、修改及刪除項目
- 指定是否繼續進行傳送的指示器
- 要插入到「傳送日誌」的字串。

實作 [SourceTransferStartExit.java](#) 介面來呼叫此結束點的使用者結束程式碼。

在整個檔案傳送完成後

在整個檔案傳送完成後，來源代理程式會呼叫此結束程式。

此結束點的使用範例包括執行一些完成作業，例如傳送電子郵件或 IBM MQ 訊息，以標示傳送已完成。

將下列引數傳遞至此結束程式：

- 傳送結束程式結果
- 來源代理程式名稱
- 目的地代理程式名稱
- 環境 meta 資料
- 傳送 meta 資料
- 檔案結果

從此結束程式傳回的資料如下所示：

- 要插入到「傳送日誌」的已更新字串。

實作 [SourceTransferEndExit.java](#) 介面來呼叫此結束點的使用者結束程式碼。

目的地端結束點

在整個檔案傳送開始之前

此結束點的一個使用範例是驗證目的地上的權限。

將下列引數傳遞至此結束程式：

- 來源代理程式名稱
- 目的地代理程式名稱
- 環境 meta 資料
- 傳送 meta 資料
- 檔案規格

從此結束程式傳回的資料如下所示：

- 更新的目的地檔案名稱集。可以修改，但不能新增或刪除項目。
- 指定是否繼續進行傳送的指示器
- 要插入到「傳送日誌」的字串。

實作 [DestinationTransferStartExit.java](#) 介面來呼叫此結束點的使用者結束程式碼。

在整個檔案傳送完成後

此使用者結束程式的使用範例是啟動使用已傳送檔案的批次程序，或者在傳送失敗時傳送電子郵件。

將下列引數傳遞至此結束程式：

- 傳送結束程式結果
- 來源代理程式名稱
- 目的地代理程式名稱
- 環境 meta 資料
- 傳送 meta 資料

- 檔案結果

從此結束程式傳回的資料如下所示：

- 要插入到「傳送日誌」的已更新字串。

實作 `DestinationTransferEndExit.java` 介面來呼叫此結束點的使用者結束程式碼。

相關概念

第 760 頁的『MFT 使用者結束程式的 Java 介面』

本節的主題為您提供使用者結束常式的 Java 介面的相關參照資訊。

相關參考

第 220 頁的『對 MFT 使用者結束程式啟用遠端除錯』

在開發使用者結束程式時，您想要使用除錯器來協助找出程式碼的問題。

第 221 頁的『MFT 來源傳送使用者結束程式範例』



第 755 頁的『MFT 資源監視器使用者結束程式』

資源監視器使用者結束程式可讓您配置在滿足監視器的觸發條件時，可在關聯作業啟動之前執行的自訂程式碼。

使用 MFT 傳送 I/O 使用者結束程式

您可以使用 Managed File Transfer 傳送 I/O 使用者結束程式，配置自訂程式碼來執行 Managed File Transfer 傳送的基礎檔案系統 I/O 工作。

通常針對 MFT 傳送，代理程式會選取其中一個內建 I/O 提供者，來與適當的檔案系統互動以進行傳送。內建 I/O 提供者支援下列類型的檔案系統：

- 一般 UNIX 型及 Windows 型檔案系統
-  z/OS 循序及分割的資料集（僅限在 z/OS 上）
-  IBM i 原生儲存檔（僅限在 IBM i 上）
- IBM MQ 佇列
- 遠端 FTP 及 SFTP 通訊協定伺服器（僅限通訊協定橋接器代理程式）
- 遠端 Connect:Direct 節點（僅適用於 Connect:Direct 橋接器代理程式）

對於不支援的檔案系統，或是需要自訂 I/O 行為的地方，您可以撰寫傳送 I/O 使用者結束程式。

傳送 I/O 使用者結束程式使用使用者結束程式的現有基礎架構。不過，這些傳送 I/O 使用者結束程式與其他使用者結束程式不同，因為在傳送每一個檔案時會多次存取其功能。

使用代理程式內容 `IOExitClasses`（在 `agent.properties` 檔中）來指定要載入的 I/O 結束程式類別。請以逗點區隔每一個結束程式類別，例如：

```
IOExitClasses=testExits.TestExit1,testExits.testExit2
```

用於傳送 I/O 使用者結束程式的 Java 介面如下所示：

IOExit

用來決定是否使用 I/O 結束程式的主要進入點。這個實例負責建立 `IOExitPath` 實例。

您只需要對代理程式內容 `IOExitClasses` 指定 `IOExit` I/O 結束程式介面。

IOExitPath

代表抽象介面；例如，資料儲存器或代表一組資料儲存器的萬用字元。您不能建立實作此介面的類別實例。此介面可以檢查路徑及列出衍生的路徑。`IOExitResourcePath` 及 `IOExitWildcardPath` 介面延伸 `IOExitPath`。

IOExitChannel

能夠在 `IOExitPath` 資源中讀取或寫入資料。

IOExitRecordChannel

對於記錄導向的 IOExitPath 資源延伸 IOExitChannel 介面，能夠以多筆記錄的形式在 IOExitPath 資源中讀取或寫入資料。

IOExitLock

代表對 IOExitPath 資源的共用或互斥存取的鎖定。

z/OS IOExitRecordResourcePath

延伸 IOExitResourcePath 介面，以代表記錄導向檔案的資料儲存器；例如，z/OS 資料集。您可以使用此介面來尋找資料，以及建立 IOExitRecordChannel 實例來進行讀取或寫入作業。

IOExitResourcePath

延伸 IOExitPath 介面來代表資料儲存器；例如，檔案或目錄。您可以使用此介面來尋找資料。如果此介面代表目錄，您可以使用 listPaths 方法來傳回路徑清單。

IOExitWildcardPath

延伸 IOExitPath 介面來代表一個表示萬用字元的路徑。您可以使用此介面來比對多個 IOExitResourcePaths。

IOExitProperties

指定內容，以決定 Managed File Transfer 如何針對 I/O 的特定層面處理 IOExitPath。例如，如果重新啟動傳送，則是使用中間檔還是從頭開始重新讀取資源。

IBM i 上的 MFT 範例使用者結束程式

Managed File Transfer 為您的安裝提供了 IBM i 特定的範例使用者結束程式。這些範例位在 `MQMFT_install_dir/samples/ioexit-IBMi` 和 `MQMFT_install_dir/samples/userexit-IBMi` 目錄中。

com.ibm.wmqfte.exit.io.ibm.i.qdls.FTEQDLSExit

`com.ibm.wmqfte.exit.io.ibm.i.qdls.FTEQDLSExit` 範例使用者結束程式會傳送 IBM i 上 QDLS 檔案系統中的檔案。安裝結束程式之後，對以 /QDLS 開頭的檔案的任何傳送都會自動使用結束程式。

若要安裝此結束程式，請完成下列步驟：

1. 將 `com.ibm.wmqfte.samples.ibm.i.ioexits.jar` 檔案從 `WMQFTE_install_dir/samples/ioexit-IBMi` 目錄複製到代理程式的 `exits` 目錄。
2. 將 `com.ibm.wmqfte.exit.io.ibm.i.qdls.FTEQDLSExit` 新增至 `IOExitClasses` 內容。
3. 請重新啟動代理程式。

com.ibm.wmqfte.exit.user.ibm.i.FileMemberMonitorExit

`com.ibm.wmqfte.exit.user.ibm.i.FileMemberMonitorExit` 範例使用者結束程式行為如同 MFT 檔案監視器，且會自動從 IBM i 程式庫傳送實體檔案成員。

若要執行此結束程式，請為 "library.qsys.monitor" meta 資料欄位指定值（例如，使用 `-md` 參數）。此參數會擷取檔案成員的 IFS 樣式的路徑，且可以包含檔案及成員萬用字元。例如，`/QSYS.LIB/FOO.LIB/BAR.FILE/*.MBR`、`/QSYS.LIB/FOO.LIB/*.FILE/BAR.MBR` 及 `/QSYS.LIB/FOO.LIB/*.FILE/*.MBR`。

此範例結束程式還具有選用的 meta 資料欄位 "naming.scheme.qsys.monitor"，您可以使用它來判斷傳送期間使用的命名方法。依預設，此欄位設定為 "unix"，這會導致目的地檔案稱為 `FOO.MBR`。您也可以指定值 "ibmi" 以使用 IBM i FTP FILE.MEMBER 方法，例如，將 `/QSYS.LIB/FOO.LIB/BAR.FILE/BAZ.MBR` 作為 `BAR.BAZ` 傳送。

若要安裝此結束程式，請完成下列步驟：

1. 將 `com.ibm.wmqfte.samples.ibm.i.userexits.jar` 檔案從 `WMQFTE_install_dir/samples/userexit-IBMi` 目錄複製到代理程式的 `exits` 目錄。

2. 將 `com.ibm.wmqfte.exit.user.ibm.FileMemberMonitorExit` 新增至 `agent.properties` 檔案中的 `sourceTransferStartExitClasses` 內容。
3. 請重新啟動代理程式。

com.ibm.wmqfte.exit.user.ibm.EmptyFileDeleteExit

如果在傳送時刪除來源檔案成員，`com.ibm.wmqfte.exit.user.ibm.EmptyFileDeleteExit` 範例使用者結束程式會刪除空檔案物件。因為 IBM i 檔案物件可能會保留許多成員，所以 MFT 會將檔案物件視為目錄。因此，您無法使用 MFT 對檔案物件執行移動作業；僅在成員層次上支援移動作業。從而，在您對成員執行移動作業時，目前的空檔案會保留。在傳送要求時，如果您要刪除這些空檔案，請使用此範例結束程式。

如果您為 "empty.file.delete" meta 資料指定 "true"，且傳送 FTEFileMember，則範例結束程式會刪除空的上層檔案。

若要安裝此結束程式，請完成下列步驟：

1. 將 `com.ibm.wmqfte.samples.ibm.userexits.jar` 檔案從 `WMQFTE_install_dir/samples/userexit-IBMi` 複製到代理程式的 `exits` 目錄。
2. 將 `com.ibm.wmqfte.exit.user.ibm.EmptyFileDeleteExit` 新增至 `agent.properties` 檔案中的 `sourceTransferStartExitClasses` 內容。
3. 請重新啟動代理程式。

相關參考

第 218 頁的『使用 MFT 傳送 I/O 使用者結束程式』

您可以使用 Managed File Transfer 傳送 I/O 使用者結束程式，配置自訂程式碼來執行 Managed File Transfer 傳送的基礎檔案系統 I/O 工作。

第 759 頁的『使用者結束程式的 MFT 代理程式內容』

除了 `agent.properties` 檔案中的標準內容外，還具有數個專用於使用者結束程式常式的進階內容。依預設，並不包含這些內容，因此，如果要使用其中任何內容，您必須手動編輯 `agent.properties` 檔案。如果在該代理程式正在執行時對 `agent.properties` 檔案進行變更，請停止並重新啟動代理程式以取得變更。

對 MFT 使用者結束程式啟用遠端除錯

在開發使用者結束程式時，您想要使用除錯器來協助找出程式碼的問題。

因為結束程式是在執行代理程式的 Java 虛擬機器內執行，所以您無法使用整合開發環境通常會包括的直接除錯支援。不過，您可以啟用 JVM 的遠端除錯，然後連接適當的遠端除錯器。

若要啟用遠端除錯，請使用標準 JVM 參數 `-Xdebug` 及 `-Xrunjdwp`。這些內容會由

`BFG_JVM_PROPERTIES` 環境變數傳遞至執行代理程式的 JVM。例如，在 UNIX 上，下列指令會啟動代理程式，並使 JVM 在 TCP 埠 8765 接聽除錯器連線。

```
export BFG_JVM_PROPERTIES="-Xdebug -Xrunjdwp:transport=dt_socket,server=y,address=8765"
fteStartAgent -F TEST_AGENT
```

代理程式會等到除錯器連接之後才啟動。在 Windows 上，請使用 `set` 指令而不是 `export` 指令。

您也可以除錯器與 JVM 之間使用其他通訊方法。例如，JVM 可以開啟除錯器的連線，反之則不然，或者，您可以使用共用記憶體代替 TCP。如需進一步詳細資料，請參閱 [Java 平台除錯器架構文件](#)。

當您在遠端除錯模式中啟動代理程式時，必須使用 `-F` (前景) 參數。

使用 Eclipse 除錯器

下列步驟適用於 Eclipse 開發環境中的遠端除錯功能。您也可以使用 JPDA 相容的其他遠端除錯器。

1. 按一下執行 > 開啟除錯對話框 (或執行 > 除錯配置或執行 > 除錯對話框，視您的 Eclipse 版本而定)。
2. 在配置類型清單中按兩下遠端 Java 應用程式，以建立除錯配置。
3. 完成配置欄位，並儲存除錯配置。如果您已在除錯模式中啟動代理程式 JVM，可立即連接至 JVM。

MFT 來源傳送使用者結束程式範例

```
/*
 * 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";
    }
}
```

通訊協定橋接器認證使用者結束程式範例

如需如何使用此使用者結束程式範例的相關資訊，請參閱第 188 頁的『使用結束類別對映檔案伺服器的認證』

```
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
}
}

```

通訊協定橋接器內容使用者結束程式範例

如需如何使用此使用者結束程式範例的相關資訊，請參閱第 183 頁的『[ProtocolBridgePropertiesExit2: 查閱通訊協定檔案伺服器內容](#)』

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;  
}  
  
}
```

將訊息放置在代理程式指令佇列上以控制 MFT

您可以撰寫應用程式，以透過將訊息放置在代理程式指令佇列上來對 Managed File Transfer 進行控制。

您可以將訊息放置在代理程式的指令佇列上，以要求代理程式執行下列其中一個動作：

- 建立檔案傳送
- 建立排定的檔案傳送
- 取消檔案傳送
- 取消排定的檔案傳送
- 呼叫指令
- 建立監視器
- 刪除監視器
- 傳回連線測試，以指出代理程式處於作用中狀態

若要要求代理程式執行上述其中一個動作，訊息必須使用符合下列其中一個綱目的 XML 格式：

FileTransfer.xsd

使用此格式的訊息可用來建立檔案傳送或排定的檔案傳送、呼叫指令，或取消檔案傳送或排定的檔案傳送。如需相關資訊，請參閱 [第 668 頁的『檔案傳送要求訊息格式』](#)。

Monitor.xsd

使用此格式的訊息可用來建立或刪除資源監視器。如需相關資訊，請參閱 [第 684 頁的『MFT 監視器要求訊息格式』](#)。

PingAgent.xsd

使用此格式的訊息可用來連線測試代理程式，以確認它處於作用中狀態。如需相關資訊，請參閱 [第 693 頁的『連線測試 MFT 代理程式要求訊息格式』](#)。

代理程式會對要求訊息傳回回覆。回覆訊息會放置在要求訊息中所定義的回覆佇列上。回覆訊息會使用下列綱目所定義的 XML 格式：

Reply.xsd

如需相關資訊，請參閱 [第 694 頁的『MFT 代理程式回覆訊息格式』](#)。

疑難排解 MFT

使用下列參照資訊來協助您診斷 Managed File Transfer 中的錯誤：

MFT 一般疑難排解

使用下列參照資訊來協助您診斷 Managed File Transfer 中的錯誤：

相關概念

[第 229 頁的『使用 MFT 的提示及要訣』](#)

這裡提供了一些建議，可協助您充分使用 Managed File Transfer。

[第 273 頁的『以 Windows 服務方式執行 MFT 代理程式或日誌程式的指引』](#)

您可以用 Windows 服務方式來執行 Managed File Transfer 代理程式、獨立式資料庫日誌程式及獨立式檔案日誌程式。如果您使用這些 Windows 服務時發生問題，可利用服務日誌檔及本主題的資訊來診斷問題。

相關工作

[第 229 頁的『對 MFT 執行追蹤』](#)

您可以使用本節中所述的方法來追蹤 Managed File Transfer。

[第 278 頁的『在 Oracle 資料庫上更新 MFT 資料庫綱目時收到錯誤時應該如何處理』](#)

使用 `ftelog_tables_oracle_702_703.sql` 檔案將資料庫綱目更新至最新層次時，您可能會收到下列錯誤訊息：`ERROR at line 1: ORA-02289: sequence does not exist`。發生此錯誤的原因是表格所使用的順序及觸發程式不在與表格相同的綱目中。

相關參考

[第 251 頁的『一般 MFT 問題』](#)

在 Managed File Transfer 網路中可能會發生的一般問題。

[第 253 頁的『fteListAgents 指令未列出您的 MFT 代理程式時應該如何處理』](#)

如果您的代理程式未由 **fteListAgents** 指令列出或未顯示在 IBM MQ Explorer 中，或您的檔案傳送未顯示在 IBM MQ Explorer 的 **傳送日誌** 中，您可以執行許多問題判斷步驟來調查原因。

[第 255 頁的『若 MFT 代理程式程序消失，但未記載任何診斷資訊時應該如何處理』](#)

在 UNIX 平台上，如果代理程式程序消失，但代理程式日誌檔卻沒有任何說明，這可能是因為代理程式的啟動方式所致。

[第 264 頁的『認為檔案傳送停滯時應該如何處理』](#)

在負載沉重的系統上，或來源與目的地代理程式之間發生網路問題時，傳送有時會呈現停滯在佇列或回復狀態。有許多因素可能會造成此狀況。

[第 264 頁的『通訊協定橋接器代理程式報告找不到檔案時應該如何處理』](#)

當通訊協定橋接器代理程式報告通訊協定橋接器所連接的 SFTP 或 FTP 伺服器傳回 `File not found` 錯誤訊息時，此訊息可能表示發生許多不同的錯誤案例之一。

[第 266 頁的『由佇列資源監視器啟動的傳送所建立的目的地檔案包含錯誤資料時應該如何處理』](#)

您可以建立資源監視器來監視佇列，並將佇列上的訊息或訊息群組傳送至檔案。使用該訊息或該訊息群組的第一個訊息上的 MQMD 訊息描述子來指定檔案名稱。如果「訊息轉為檔案」的傳送失敗，且訊息或訊息群組留在佇列上，則下次觸發監視器時，它可能導致所建立的檔案包含錯誤資料。

[第 268 頁的『訊息若是建置在 SYSTEM.MANAGED.DURABLE 佇列上或填入檔案系統時應該如何處理』](#)

如果 IBM MQ Explorer 外掛程式在協調佇列管理程式上使用可延續訂閱，則訊息可以在 SYSTEM.MANAGED.DURABLE 佇列。如果您有流量高的 Managed File Transfer 網路，而且（或者）又不常使用「IBM MQ Explorer」外掛程式，則此訊息資料可能會填入本端檔案系統中。

[第 268 頁的『在發佈之前檢查訊息』](#)

因為代理程式可連接到 IBM WebSphere MQ 6.0 佇列管理程式，所以代理程式不會使用 IBM WebSphere MQ 7.0 所引進的直接發佈方式，而是將一般訊息傳送至包含 MQRFH 標頭的協調佇列管理程式。MQRFH 標頭需要發佈訊息的內容。這些訊息會傳送至協調佇列管理程式上的 SYSTEM.FTE 佇列，且訊息通常會直接從該佇列發佈。如果發生錯誤狀況導致此發佈停止，您可以在嘗試發佈前先檢查該佇列上的訊息，以協助進行診斷。您可以透過完成下列步驟來執行此作業：

[第 269 頁的『傳送 IBM i 儲存檔時發生可能的錯誤』](#)

如果您使用 Managed File Transfer 來傳送相同的 IBM i 儲存檔數次，則傳送可能失敗。

[第 270 頁的『設定與訊息大小相關聯的 MQ 屬性及 MFT 內容的指引』](#)

您可以變更 IBM MQ 屬性及 Managed File Transfer 內容，以影響 Managed File Transfer 在讀取或寫入不同大小訊息時的行為。

[第 278 頁的『MFT 日誌程式錯誤處理及拒絕』](#)

Managed File Transfer 日誌程式可識別兩種類型的錯誤：個別訊息錯誤及一般錯誤。

[第 279 頁的『已啟動 MFT 日誌程式，但資料庫中未記載任何傳送資訊時應該如何處理』](#)

Managed File Transfer 日誌程式所使用的資料庫表格，需要資料庫具有 8KB 以上的分頁大小。如果資料庫的分頁大小不夠大，則無法適當地建立表格，而且您會看到錯誤 `SQLSTATE=42704`。

[第 397 頁的『fteDisplayVersion』](#)

使用 **fteDisplayVersion** 指令可顯示已安裝 Managed File Transfer 的版本。

[第 282 頁的『BFGSS0023E 錯誤及如何避免錯誤』](#)

如果您從安裝中解除安裝修正套件以回到舊版的產品，但在解除安裝時與此安裝相關聯的代理程式涉及受管理傳送，則該代理程式將無法啟動，而且會報告 BFGSS0023E 錯誤。您可以完成一些步驟來避免此錯誤，這些步驟應可在代理程式重新啟動時防止 BFGSS0023E 訊息出現。


[第 284 頁的『MFT 的回覆碼』](#)

Managed File Transfer 指令、Ant 作業及日誌訊息提供回覆碼，指出功能是否順利完成。

使用 MFT 的提示及要訣

這裡提供了一些建議，可協助您充分使用 Managed File Transfer。

- 如果變更 `agent.properties` 檔案，請停止並重新啟動代理程式以取得變更。
- 如果您已啟動檔案傳送，但未顯示傳送進度且未報告任何錯誤，則請檢查來源代理程式是否在執行中。如果顯示傳送但沒有任何進度，則請檢查目的地代理程式是否也在執行中。您可以在代理程式日誌中檢查代理程式的現行狀態，或使用 `ftePingAgent` 指令檢查代理程式是否處於作用中狀態。
- 當您要使用 `fteCancelTransfer` 指令取消個別傳送時，在 `-agentName` 參數中使用來源或目的地代理程式均可。但是，使用 `fteDeleteScheduledTransfer` 指令刪除傳送排程時，您必須在 `-agentName` 參數中使用來源代理程式名稱。
- 當您建立檔案傳送時，來源及目的地檔案路徑（絕對或相對）僅在來源及目的地代理程式上有效。從中發出 `fteCreateAgent` 指令的系統及目錄與正在傳送的檔案無關。
- 您的預設環境設定可能無法完整支援 Managed File Transfer，特別是在執行多個並行傳送時。如果有代理程式發生錯誤指示記憶體不足，請依需要檢查並更新下列參數：

-  對於 UNIX 平台：執行以下指令：`ulimit -m 1048576`（或大約 1 GB）。此常駐集大小上限足以容許最多 25 個並行傳送（25 個並行傳送為代理程式傳送數上限的預設值）。
- 對於所有平台：設定 `BFG_JVM_PROPERTIES` 環境變數，如下所示：`BFG_JVM_PROPERTIES="-Xmx1024M"`

如果您想要容許並行傳送數大於上限預設值 25，請對 `ulimit` and `BFG_JVM_PROPERTIES` 使用比建議值更大的大小。

註：對於 Connect:Direct 橋接器代理程式，並行傳送數上限的預設值為 5。

- 當您使用 Managed File Transfer 在不同平台之間以文字模式傳送檔案時，目的地平台可能會不支援來源平台的預設檔案編碼。這會導致傳送失敗，且會顯示以下錯誤：

```
BFGI00058E: The transfer source encoding xxx is illegal or for an unsupported character set.
```

您可以透過以下方式解決此錯誤：使用環境變數將來源編碼設定為目的地平台所支援的編碼。在來源系統上設定 `BFG_JVM_PROPERTIES` 系統環境變數如下：`BFG_JVM_PROPERTIES="-Dfile.encoding=xxx"`，其中 `xxx` 是目的地平台支援的編碼。例如，如果您要以文字模式將檔案從 Solaris 平台傳送至不同平台，且來源語言環境設為 "ja"，請設定 `BFG_JVM_PROPERTIES` 如下：

`BFG_JVM_PROPERTIES="-Dfile.encoding=EUC-JP"`。如果來源語言環境設定為 "ja_JP.PCK"，請設定 `BFG_JVM_PROPERTIES` 如下：`BFG_JVM_PROPERTIES="-Dfile.encoding=Shift_JIS"`。

當您啟動新建傳送時，您也可以使用 `-sce` 參數解決個別傳送的此錯誤。如需相關資訊，請參閱 [第 367 頁的『fteCreateTransfer: 啟動新的檔案傳送』](#)。

相關參考

[第 508 頁的『MFT 的 Java 系統內容』](#)

有一些 Managed File Transfer 指令及代理程式內容必須定義為 Java 系統內容，因為它們為無法使用指令或代理程式內容機制的早期功能定義配置。

對 MFT 執行追蹤

您可以使用本節中所述的方法來追蹤 Managed File Transfer。

關於這項作業

如需追蹤各種 Managed File Transfer 資源的詳細資訊，請參閱 [第 232 頁的『在 Multiplatforms 上追蹤 Managed File Transfer 資源』](#) 和 [第 238 頁的『追蹤 Managed File Transfer for z/OS 資源』](#)。

fteSetAgentTraceLevel

使用 `fteSetAgentTraceLevel` 指令可動態地修改代理程式的現行追蹤層次。

用途

使用此指令來開啟及關閉代理程式追蹤，或者變更已設定的代理程式追蹤的層次。使用 **fteSetAgentTraceLevel** 指令時，您無需關閉並重新啟動代理程式來修改追蹤層次。產生的追蹤檔案位於 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%/trace%PID%.txt` 中，其中 `%PID%` 是代理程式實例的程序 ID。



小心：

Multi 在 Multiplatforms 上使用 IBM WebSphere MQ 7.5 或更新版本時，只有執行代理程式處理程序的使用者可以執行 **fteSetAgentTraceLevel** 指令。

V 9.0.2 **z/OS** 若為 z/OS，則從 Continuous Delivery 中的 IBM MQ 9.0.2 開始，下列使用者 ID 或成員可以執行 **fteSetAgentTraceLevel** 指令：

- 以其身分執行代理程式程序的使用者 ID。
- 代理程式內容 **adminGroup** 所指定之群組的成員。

V 9.0.0.1 這也適用於 IBM MQ 9.0.0 Fix Pack 1 中的 Long Term Support 版本。

如需相關資訊，請參閱第 463 頁的『MFT agent.properties 檔案』中的 **adminGroup** 內容。

在 IBM WebSphere MQ 7.5 及更新版本中，**fteSetAgentTraceLevel** 指令也會寫入代理程式程序控制器的追蹤。產生的追蹤檔案位於 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/pctrace%PID%/pctrace%PID%.txt` 中，其中 `%PID%` 是代理程式實例的程序 ID。

您也可以使用該指令，讓代理程式程序產生 Javacore。代理程式會在下列目錄中產生 Javacore 檔：`MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name`。

因為執行追蹤會大幅影響效能，而且會產生大量追蹤資料，因此，請謹慎執行追蹤，並且僅在必要時才執行。一般而言，只需要在 IBM 客戶服務代表要求時，再啟用追蹤即可。



小心：

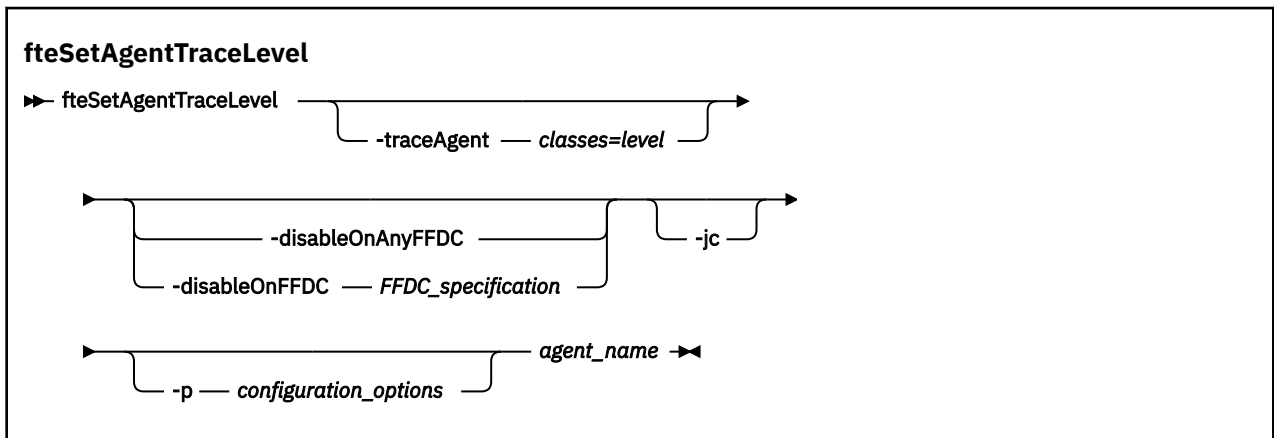
1. 您必須在執行代理程式的系統上執行此指令。
2. 在代理程式重新啟動時，追蹤及記載不會持續保存。

如果代理程式終止並由「處理程序控制器」處理程序重新啟動，則在更新 `agent.properties` 檔案以包括必要的追蹤及日誌內容之前，動態追蹤及日誌不會生效。

您可以在 `agent.properties` 檔中設定進一步追蹤內容，例如，追蹤檔大小及要保留的追蹤檔數目。[進階代理程式內容](#)中說明了這些內容。

僅在要使用一組不同於您的預設集的配置選項時，才為此指令指定選用的 `-p` 參數。如需相關資訊，請參閱第 463 頁的『MFT agent.properties 檔案』。

語法



參數

-traceAgent classes=level

必要項目。為代理程式追蹤設定的層次及追蹤適用的類別。

V 9.0.5 從 IBM MQ 9.0.5 開始，您可以指定類別規格清單（以冒號區隔）。此選項可讓您在不同層次設定不同的類別追蹤。例如：

```
fteSetAgentTraceLevel -traceAgent com.ibm.wmqfte.agent=all:com.ibm.wmqfte.cmdhandler=moderate
AGENT1
```

您仍可指定要套用追蹤層次的類別規格清單（以冒號區隔）。如果未指定此參數，則追蹤層次會套用到所有代理程式類別。請使用下列格式：

```
classes=level
```

例如：

```
com.ibm.wmqfte=all
```

您可以將 *classes* 替換為 Managed File Transfer 套件名稱，只追蹤特定的套件。不過，因為此選項只會擷取代理程式行為的子集，所以一般不建議您使用套件過濾。

如果 (*classes*) 的開頭有加號 (+)，則加號之後的追蹤類別清單，都會加入目前正在追蹤的任何現有追蹤類別中。

有效的追蹤層次選項如下，並且會依照追蹤檔案大小及其詳細資料遞增列出：

off

關閉代理程式追蹤，但繼續將資訊寫入日誌檔。這是預設選項。

flow

擷取與代理程式中的程序流程相關聯的追蹤點資料。

moderate

擷取追蹤中的適量診斷資訊。

verbose

擷取追蹤中的大量診斷資訊。

全部

設定對所有代理程式類別執行代理程式追蹤。

若要啟動代理程式的完整追蹤，請執行下列指令：

```
fteSetAgentTraceLevel -traceAgent =all AGENT_NAME
```

若要停止代理程式的完整追蹤，請執行下列指令：

```
fteSetAgentTraceLevel -traceAgent =off AGENT_NAME
```

-disableOnAnyFFDC

選用項目。如果已指定此參數，則代理程式產生「首次失敗資料擷取 (FFDC)」檔案時，會停用追蹤。

您只能指定其中一個 **-disableOnAnyFFDC** 及 **-disableOnFFDC** 參數。

-disableOnFFDC FFDC_specification

選用項目。如果已指定此參數，則代理程式產生與 *FFDC_specification* 相符的「首次失敗資料擷取 (FFDC)」檔案時，會停用追蹤。*FFDC_specification* 是值的清單（以逗點區隔）。值的格式可以為：

class_name

FFDC 所源自的類別名稱。例如，`com.ibm.wmqfte.classA`。

class_name:probe_ID

FFDC 所源自的類別中的類別名稱及位置的探測 ID。例如, `com.ibm.wmqfte.classB:1`。

您只能指定其中一個 **-disableOnAnyFFDC** 及 **-disableOnFFDC** 參數。

-jc

選用項目。要求代理程式產生 Javacore 檔。IBM 服務團隊可能要求您使用此參數來執行該指令, 以協助診斷問題。此參數不能與 **-p** 以外的任何其他參數一起使用。

-p configuration_options

選用項目。此參數會決定用於設定代理程式追蹤層次的配置選項集。請使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後, 指令即會使用與此非預設協調佇列管理程式相關聯的一組內容檔。

如果未指定此參數, 則會使用一組以預設協調佇列管理程式為基礎的配置選項。

agent_name

必要項目。您要為其設定追蹤層次的 Managed File Transfer Agent 的名稱。

/? 或 -h

選用項目。顯示指令語法。

範例

在此範例中, AGENT1 的所有類別的追蹤層次設定為 all:

```
fteSetAgentTraceLevel -traceAgent com.ibm.wmqfte=all AGENT1
```

在此範例中, AGENT1 的類別 `com.ibm.wmqfte.agent.Agent` 及 `com.ibm.wmqfte.cmdhandler` 的追蹤層次設定為 all:

```
fteSetAgentTraceLevel -traceAgent com.ibm.wmqfte.agent.Agent,com.ibm.wmqfte.cmdhandler=moderate AGENT1
```

在此範例中, 從追蹤中排除子類別, 因為 **-traceLevel** 參數設定為關閉。以詳細層次追蹤以 `com.ibm.outer` 開頭的所有類別 (以 `com.ibm.outer.inner` 開頭的類別除外):

```
fteSetAgentTraceLevel -traceAgent com.ibm.outer=verbose AGENT1  
fteSetAgentTraceLevel -traceAgent +com.ibm.outer.inner=off AGENT1
```

回覆碼**0**

指令已順利完成。

1

指令未順利結束。

Multi 在 Multiplatforms 上追蹤 Managed File Transfer 資源

提供 Managed File Transfer 中的追蹤機能, 以協助「IBM 支援中心」診斷您的問題。您可以追蹤各種不同的資源。

關於這項作業

請參閱

- 第 233 頁的『[在 Multiplatforms 上追蹤 Managed File Transfer 代理程式](#)』, 以取得如何追蹤代理程式的相關資訊。
- 第 235 頁的『[在 Multiplatforms 上追蹤 Managed File Transfer 指令](#)』, 以取得如何追蹤指令的相關資訊。

Multi

在 *Multiplatforms* 上追蹤 *Managed File Transfer* 代理程式

提供 *Managed File Transfer* 中的追蹤機能，以協助「IBM 支援中心」診斷您的問題。各種指令及內容會控制此機能的行為。

關於這項作業

如果要求您提供追蹤輸出以調查代理程式的問題，請使用下列其中一個選項，視您是否可以在短時間內停止代理程式而定。

如果您不確定要使用哪個選項，請聯絡「IBM 支援中心」代表，他們會建議您針對您所看到的問題收集追蹤資料的最佳方式。

程序

- 如果您可以在短時間內停止代理程式，請從啟動開始收集代理程式的追蹤資料。
如需相關資訊，請參閱第 233 頁的『從啟動開始收集 *Managed File Transfer* 代理程式追蹤』。
- 如果無法停止代理程式，請使用 `fteSetAgentTraceLevel` 指令動態收集追蹤。
如需相關資訊，請參閱第 234 頁的『動態收集 *Managed File Transfer* 代理程式追蹤』。

Multi

從啟動開始收集 *Managed File Transfer* 代理程式追蹤

如果您可以在短時間內停止代理程式，則應該從啟動開始收集 *Managed File Transfer* 代理程式追蹤。

開始之前

您需要在 `agent.properties` 檔案中為需要追蹤的代理程式設定各種內容。

關於這項作業

若要從啟動收集追蹤，請完成下列步驟。

程序

1. 找出需要追蹤之代理程式的 `agent.properties` 檔案。
`agent.properties` 檔案可在 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` 目錄中找到。
2. 編輯檔案並新增下列內容的項目：
 - **trace**=追蹤規格
trace 內容會決定要追蹤的內部類別及套件。除非 IBM 支援中心代表另有指定，否則請將此內容設為值 `com.ibm.wmqfte=all`。
 - **traceFiles**=要使用的追蹤檔數目
 - **traceSize**=每一個追蹤檔的大小 (MB)
traceFiles 和 **traceSize** 內容用來控制所收集的追蹤資料量。您應該將這些內容設為較大的值，以儘可能收集更多追蹤資料。

例如，若要使用追蹤規格 `com.ibm.wmqfte=all` 收集 1GB 折返追蹤，請將下列行新增至 `agent.properties` 檔案：

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

這會導致代理程式將追蹤資料寫入最多 5 個檔案，其中每一個檔案的大小都是 200MB。

如需這些代理程式內容的相關資訊，請參閱 `MFT agent.properties` 檔案。

3. 使用 `fteStopAgent` 指令，停止需要追蹤的代理程式。
4. 執行 `fteStartAgent` 指令，以啟動代理程式。

5. 重新產生問題。
6. 停止代理程式。
7. 編輯代理程式的 `agent.properties` 檔案，並移除您在步驟 [第 233 頁的『2』](#) 中新增之 **trace**、**traceFiles** 及 **traceSize** 內容的項目。
這可確保下次重新啟動代理程式時不會啟用追蹤。

結果

產生的追蹤檔會寫入 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` 目錄，其中 `%PID%` 是代理程式的處理程序 ID。

Multi 動態收集 Managed File Transfer 代理程式追蹤

fteSetAgentTraceLevel 指令容許從執行中代理程式收集追蹤。如果「IBM 支援中心」需要查看無法停止之代理程式的追蹤，這會非常有用。

關於這項作業

若要使用 **fteSetAgentTraceLevel** 指令從代理程式收集追蹤，請完成下列步驟。

程序

1. 執行下列指令，以開啟代理程式的追蹤：

```
fteSetAgentTraceLevel -traceAgent classes=level agent_name
```

`-traceAgent` 參數決定要追蹤的內部類別及套件。除非 IBM 支援中心代表另有指定，否則請將此內容設為值 `com.ibm.wmqfte=all`。

2. 重新產生問題。
3. 透過執行下列指令，關閉代理程式的追蹤：

```
fteSetAgentTraceLevel -traceAgent =off agent_name
```

如果代理程式忙碌，追蹤檔可能會快速折返，並改寫調查問題所需的資訊。如果是這樣，請排定一些時間來停止代理程式，然後依照下列步驟中的詳細說明來繼續進行。如果您無法在短時間內停止代理程式，請聯絡「IBM 支援中心」代表，並討論要使用的替代追蹤規格，以減少所產生的追蹤資料量。

4. 找出需要追蹤之代理程式的 `agent.properties` 檔案。

`agent.properties` 檔案可在 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` 目錄中找到。

5. 編輯檔案並新增下列內容的項目：

```
traceFiles=number_of_trace_files_to_use  
traceSize=size_of_each_trace_file_in_MB
```

traceFiles 和 **traceSize** 內容用來控制所收集的追蹤資料量。

traceFiles 內容的預設值為 5，而 **traceSize** 內容的預設值為 20MB。這表示如果您動態開啟追蹤，且尚未設定內容，代理程式會將追蹤資訊寫入 5 個覆蓋追蹤檔，每一個追蹤檔的大小上限為 20MB。

您應該將這些內容設為較大的值，以儘可能收集更多追蹤資料。

例如，若要收集 1GB 折返追蹤，請將下列行新增至 `agent.properties` 檔案：

```
traceFiles=5  
traceSize=200
```

這會導致代理程式將追蹤資料寫入最多 5 個檔案，其中每一個檔案的大小都是 200MB。

如需這些代理程式內容的相關資訊，請參閱 [MFT agent.properties](#) 檔案。

6. 透過執行 **fteStopAgent** 指令，停止代理程式。
7. 執行 **fteStartAgent** 指令，以啟動代理程式。
8. 透過執行下列指令，啟用代理程式的追蹤：

```
fteSetAgentTraceLevel -traceAgent classes=level agent_name
```

除非 IBM 支援中心代表另有指定，否則請將 **-traceAgent** 內容設為值 `com.ibm.wmqfte=all`。

9. 重新產生問題。
10. 透過執行下列指令，關閉代理程式上的追蹤：

```
fteSetAgentTraceLevel -traceAgent =off agent_name
```

結果

產生的追蹤檔會寫入 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` 目錄，其中 `%PID%` 是代理程式的處理程序 ID。

Multi 在 *Multiplatforms* 上追蹤 *Managed File Transfer* 指令提供 *Managed File Transfer* 中的追蹤機能，以協助「IBM 支援中心」診斷您的問題。您可以使用此機能來追蹤指令。

關於這項作業



小心： 追蹤指令只會收集指令所執行之處理程序的相關資訊。它不會追蹤代理程式在處理該指令時可能執行的任何活動。

程序

1. 啟動命令提示字元，並導覽至 `MQ_INSTALLATION_PATH\bin` 目錄。
2. 執行指令：

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` 是要追蹤的指令名稱。
- `classes=level` 是要使用的追蹤層次，以及要啟用追蹤的類別。除非 IBM 支援代表另有指定，否則請將此設為 `com.ibm.wmqfte=all`。
- `directory_path` 是將寫入追蹤檔的目錄。
- `command_arguments` 是需要傳遞給指令的引數，例如 **ftePingAgent** 指令的代理程式名稱。

結果

產生的追蹤檔會寫入 **-tracePath** 參數指定的目錄中。

追蹤檔案稱為 `trace%PID%.txt.number`，其中：

- `%PID%` 是指令的處理程序 ID。
- `number` 是追蹤檔的序號。通常，指令所產生的追蹤資訊包含在序號為 0 的單一追蹤檔中。

不過，指令可能會產生許多追蹤資訊。在此狀況下，追蹤會寫入多個檔案。現行追蹤檔的序號為 0，下一個最舊追蹤檔的序號為 1，依此類推。

指令的追蹤輸出會寫入最多五個折返追蹤檔。每一個追蹤檔的大小上限是 20MB。

註: 如果執行指令的使用者無權寫入 **-tracePath** 參數指定的目錄, 則追蹤輸出會寫入標準錯誤。

範例

在此範例中, 會追蹤 **fteListAgents** 指令, 並將追蹤寫入 C:\trace 目錄:

```
fteListAgents -trace com.ibm.wmqfte=all -tracePath C:\trace
```

在此範例中, 會追蹤 **fteCreateTransfer** 指令, 並將追蹤寫入 /tmp 目錄:

```
fteCreateTransfer -trace com.ibm.wmqfte=all -tracePath /tmp -t text -sa AGENT1  
-da AGENT2 -df /import/transferredfile.txt /export/originalfile.txt
```

寫入 /tmp 的追蹤檔只包含 **fteCreateTransfer** 指令所執行之處理程序的相關資訊, 例如指令如何建置傳送至代理程式的傳送要求訊息, 以及它等待代理程式傳回確認通知的時間長度, 指出它已收到要求。追蹤檔不包含任何關於傳送本身的資訊。

Multi

在 *Multiplatforms* 上追蹤 *Managed File Transfer* 獨立式日誌程式提供 *Managed File Transfer* 中的追蹤機能, 以協助「IBM 支援中心」診斷您的問題。各種指令及內容會控制此機能的行為。

關於這項作業

如果要求您提供追蹤輸出以調查日誌程式的問題, 請使用下列其中一個選項, 視您是否可以在短時間內停止日誌程式而定。

如果您不確定要使用哪個選項, 請聯絡「IBM 支援中心」代表, 他們會建議您針對您所看到的問題收集追蹤資料的最佳方式。

程序

- 如果您可以在短時間內停止日誌程式, 請從啟動開始收集日誌程式的追蹤資料。
請參閱 第 236 頁的『從啟動開始收集 *Managed File Transfer* 獨立式日誌程式追蹤』。
- 如果無法停止日誌程式, 請使用 **fteSetLoggerTraceLevel** 指令動態收集追蹤。
請參閱 第 237 頁的『動態收集 *Managed File Transfer* 獨立式日誌程式追蹤』。

Multi

從啟動開始收集 *Managed File Transfer* 獨立式日誌程式追蹤如果您可以在短時間內停止日誌程式, 則應該從啟動開始收集 *Managed File Transfer* 日誌程式追蹤。

開始之前

您需要在 `logger.properties` 檔案中為需要追蹤的日誌程式設定各種內容。

關於這項作業

若要從啟動收集追蹤, 請完成下列步驟。

程序

1. 找出需要追蹤之日誌程式的 `logger.properties` 檔。
`logger.properties` 檔案可在 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 目錄中找到。
2. 編輯檔案並新增下列內容的項目:
 - **trace**=追蹤規格

trace 內容會決定要追蹤的內部類別及套件。除非 IBM 支援中心代表另有指定，否則請將此內容設為值 `com.ibm.wmqfte=all`。

- **traceFiles**=要使用的追蹤檔數目
- **traceSize**=每一個追蹤檔的大小 (MB)

traceFiles 和 **traceSize** 內容用來控制所收集的追蹤資料量。您應該將這些內容設為較大的值，以儘可能收集更多追蹤資料。

例如，若要使用追蹤規格 `com.ibm.wmqfte=all` 收集 1GB 折返追蹤，請將下列行新增至 `logger.properties` 檔案：

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

這會導致日誌程式將追蹤資料寫入最多 5 個檔案，其中每一個檔案的大小都是 200MB。

如需這些日誌程式內容的相關資訊，請參閱 [MFT logger.properties](#) 檔案。

3. 使用 **fteStopLogger** 指令，停止需要追蹤的日誌程式。
4. 執行 **fteStartLogger** 指令，以啟動日誌程式。
5. 重新產生問題。
6. 停止日誌程式。
7. 編輯日誌程式的 `logger.properties` 檔案，並移除您在步驟 [第 236 頁的『2』](#) 中新增之 **trace**、**traceFiles** 及 **traceSize** 內容的項目。
這可確保下次重新啟動日誌程式時不會啟用追蹤。

結果

產生的追蹤檔會寫入 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/logger/logger_name/logs/trace%PID%` 目錄，其中 `%PID%` 是日誌程式的程序 ID。

Multi

動態收集 *Managed File Transfer* 獨立式日誌程式追蹤

fteSetLoggerTraceLevel 指令容許從執行中的日誌程式收集追蹤。如果「IBM 支援中心」需要查看無法停止之日誌程式的追蹤，這會非常有用。

關於這項作業

若要使用 **fteSetLoggerTraceLevel** 指令從日誌程式收集追蹤，請完成下列步驟。

程序

1. 執行下列指令，以開啟日誌程式的追蹤：

```
fteSetLoggerTraceLevel -traceLogger classes=level logger_name
```

`-traceLogger` 參數決定要追蹤的內部類別及套件。除非 IBM 支援中心代表另有指定，否則請將此內容設為值 `com.ibm.wmqfte=all`。

2. 重新產生問題。
3. 使用下列指令關閉追蹤：

```
fteSetLoggerTraceLevel -traceLogger =off logger_name
```

4. 如果日誌程式忙碌，追蹤檔可能會快速折返，並改寫調查問題所需的資訊。
如果您可以短暫停止日誌程式，請完成下列步驟，以減少所收集的追蹤資料量。否則，請聯絡「IBM 支援中心」並討論替代追蹤規格，以減少所收集的追蹤資料量。
 - a) 請排定一些時間來停止日誌程式。
 - b) 找出需要追蹤之日誌程式的 `logger.properties` 檔。

logger.properties 檔案可在 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 目錄中找到。

- c) 編輯檔案並新增下列內容的項目：

```
traceFiles=number_of_trace_files_to_use
traceSize=size_of_each_trace_file_in_MB
```

traceFiles 和 **traceSize** 內容用來控制所收集的追蹤資料量。

traceFiles 內容的預設值為 5，而 **traceSize** 內容的預設值為 20MB。這表示如果您動態開啟追蹤，且尚未設定內容，代理程式會將追蹤資訊寫入 5 個覆蓋追蹤檔，每一個追蹤檔的大小上限為 20MB。

您應該將這些內容設為較大的值，以儘可能收集更多追蹤資料。

例如，若要收集 1GB 折返追蹤，請將下列行新增至 logger.properties 檔案：

```
traceFiles=5
traceSize=200
```

這會導致日誌程式將追蹤資料寫入最多 5 個檔案，其中每一個檔案的大小都是 200MB。

如需這些日誌程式內容的相關資訊，請參閱 [MFT logger.properties](#) 檔案。

- d) 執行 **fteStopLogger** 指令來停止日誌程式。
e) 執行 **fteStartLogger** 指令，以啟動日誌程式。
f) 執行下列指令，以開啟日誌程式的追蹤。除非 IBM 支援中心代表另有指定，否則請將 **-traceLogger** 內容設為值 `com.ibm.wmqfte=all`。

```
fteSetLoggerTraceLevel -traceLogger classes=level logger_name
```

- g) 重新產生問題。
h) 執行下列指令，以關閉日誌程式的追蹤：

```
fteSetLoggerTraceLevel -traceLogger =off logger_name
```

結果

產生的追蹤檔會寫入 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%` 目錄，其中 %PID% 是日誌程式的程序 ID。

追蹤 Managed File Transfer for z/OS 資源

提供 Managed File Transfer for z/OS 中的追蹤機能，以協助「IBM 支援中心」診斷您的問題。您可以追蹤各種不同的資源。

關於這項作業

請參閱

- 第 238 頁的『[追蹤 Managed File Transfer for z/OS 代理程式](#)』，以取得如何追蹤代理程式的相關資訊。
- 第 244 頁的『[追蹤 Managed File Transfer for z/OS 指令](#)』，以取得如何追蹤指令的相關資訊。

追蹤 Managed File Transfer for z/OS 代理程式

提供 Managed File Transfer for z/OS 中的追蹤機能，以協助「IBM 支援中心」診斷您的問題。各種指令及內容會控制此機能的行為。

關於這項作業

如果系統要求您提供追蹤輸出以調查代理程式的問題，請使用下列其中一個選項。

如果您不確定要使用哪個選項，請聯絡「IBM 支援中心」代表，他們會建議您針對您所看到的問題收集追蹤資料的最佳方式。

程序

- 如果您可以在短時間內停止代理程式，請從啟動開始收集代理程式的追蹤資料。
如需相關資訊，請參閱第 239 頁的『從啟動開始收集 Managed File Transfer for z/OS 代理程式追蹤』。
- 如果無法停止代理程式，請使用 **fteSetAgentTraceLevel** 指令動態收集追蹤。
如需相關資訊，請參閱第 241 頁的『動態收集 Managed File Transfer for z/OS 代理程式追蹤』。

z/OS 從啟動開始收集 Managed File Transfer for z/OS 代理程式追蹤

如果您可以在短時間內停止代理程式，則應該從啟動開始收集 IBM MQ Managed File Transfer 代理程式追蹤。

關於這項作業

收集追蹤的方式取決於使用 Unix 系統服務 (USS) 或 JCL 來管理代理程式。

如果您不確定要使用下列哪一個選項，請聯絡「IBM 支援中心」代表，他們會建議您針對您所看到的問題收集追蹤資料的最佳方式。

程序

- 如果您使用 z/OS UNIX，請參閱第 239 頁的『使用 USS 從啟動收集代理程式追蹤』。
- 如果您使用 JCL，請參閱第 240 頁的『使用 JCL 從啟動收集代理程式追蹤』。

z/OS 使用 USS 從啟動收集代理程式追蹤

若要收集從啟動開始使用「Unix 系統服務 (USS)」來管理之 Managed File Transfer for z/OS 代理程式的追蹤資料，您需要在該代理程式啟動之前，在 `agent.properties` 檔案中設定該代理程式的各種內容需求。

關於這項作業

若要從啟動收集追蹤，請完成下列步驟。

程序

1. 找出需要追蹤之代理程式的 `agent.properties` 檔案。
`agent.properties` 檔案可在 `BFG_DATA/mqft/config/coordination_qmgr_name/agents/agent_name` 目錄中找到。
2. 編輯檔案並新增下列內容的項目：
 - **trace**=追蹤規格
trace 內容會決定要追蹤的內部類別及套件。除非 IBM 支援中心代表另有指定，否則請將此內容設為值 `com.ibm.wmqfte=all`。
 - **traceFiles**=要使用的追蹤檔數目
 - **traceSize**=每一個追蹤檔的大小 (MB)
traceFiles 和 **traceSize** 內容用來控制所收集的追蹤資料量。您應該將這些內容設為較大的值，以儘可能收集更多追蹤資料。

例如，若要使用追蹤規格 `com.ibm.wmqfte=all` 收集 1GB 折返追蹤，請將下列行新增至 `agent.properties` 檔案：

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

這會導致代理程式將追蹤資料寫入最多 5 個檔案，其中每一個檔案的大小都是 200MB。

如需這些代理程式內容的相關資訊，請參閱 [MFT agent.properties](#) 檔案。

3. 使用 **fteStopAgent** 指令，停止需要追蹤的代理程式。
4. 執行 **fteStartAgent** 指令，以啟動代理程式。
5. 重新產生問題。
6. 停止代理程式。
7. 編輯代理程式的 `agent.properties` 檔案，並移除您在步驟 第 239 頁的『2』中新增之 **trace**、**traceFiles** 及 **traceSize** 內容的項目。
這可確保下次重新啟動代理程式時不會啟用追蹤。

結果

產生的追蹤檔會寫入 `BFG_DATA/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` 目錄，其中 `%PID%` 是代理程式的處理程序 ID。

使用 JCL 從啟動收集代理程式追蹤

若要收集從啟動開始使用 JCL 來管理之 Managed File Transfer for z/OS 代理程式的追蹤資料，您需要在該代理程式啟動之前，在 `agent.properties` 檔案中設定該代理程式的各種內容需求。

關於這項作業

若要從啟動收集追蹤，請完成下列步驟。

程序

1. 找出需要追蹤之代理程式的 `agent.properties` 檔案。
`agent.properties` 檔案可在 `BFG_DATA/mqft/config/coordination_qmgr_name/agents/agent_name` 目錄中找到。
2. 編輯檔案並新增下列內容的項目：
 - **trace**=追蹤規格
trace 內容會決定要追蹤的內部類別及套件。除非 IBM 支援中心代表另有指定，否則請將此內容設為值 `com.ibm.wmqfte=all`。
 - **traceFiles**=要使用的追蹤檔數目
 - **traceSize**=每一個追蹤檔的大小 (MB)
traceFiles 和 **traceSize** 內容用來控制所收集的追蹤資料量。您應該將這些內容設為較大的值，以儘可能收集更多追蹤資料。

例如，若要使用追蹤規格 `com.ibm.wmqfte=all` 收集 1GB 折返追蹤，請將下列行新增至 `agent.properties` 檔案：

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

這會導致代理程式將追蹤資料寫入最多 5 個檔案，其中每一個檔案的大小都是 200MB。

如需這些代理程式內容的相關資訊，請參閱 [MFT agent.properties](#) 檔案。

3. 尋找包含需要追蹤之代理程式 JCL 的資料集。
4. 提交資料集內的 BFGAGSP 成員以停止代理程式。
5. 透過提交資料集中的 BFGAGST 成員，重新啟動代理程式。
6. 重新產生問題。
7. 請在資料集中提交 BFGAGSP 成員，以再次停止代理程式。
8. 編輯代理程式的 `agent.properties` 檔案，並移除您在步驟 第 240 頁的『2』中新增之 **trace**、**traceFiles** 及 **traceSize** 內容的項目。

這可確保下次重新啟動代理程式時不會啟用追蹤。

結果

產生的追蹤檔會寫入 `BFG_DATA/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` 目錄，其中 `%PID%` 是代理程式的處理程序 ID。

z/OS 動態收集 *Managed File Transfer for z/OS* 代理程式追蹤
如果您無法在短時間內停止代理程式，則應該動態收集 *Managed File Transfer for z/OS* 代理程式追蹤。

關於這項作業

收集追蹤的方式取決於使用 Unix 系統服務 (USS) 或 JCL 來管理代理程式。

如果您不確定要使用下列哪一個選項，請聯絡「IBM 支援中心」代表，他們會建議您針對您所看到的問題收集追蹤資料的最佳方式。

程序

- 如果您使用：
 - Unix 系統服務 (USS)，請參閱第 241 頁的『使用 USS 動態收集代理程式追蹤』。
 - JCL，請參閱第 242 頁的『使用 JCL 動態收集代理程式追蹤』。

z/OS 使用 USS 動態收集代理程式追蹤
fteSetAgentTraceLevel 指令容許從執行中代理程式收集追蹤。如果「IBM 支援中心」需要查看無法停止之代理程式的追蹤，這會非常有用。

關於這項作業

若要使用 **fteSetAgentTraceLevel** 指令從 *Managed File Transfer for z/OS* 代理程式收集追蹤，請完成下列步驟。

程序

1. 執行下列指令，以開啟代理程式的追蹤：

```
fteSetAgentTraceLevel -traceAgent classes=level agent_name
```

`-traceAgent` 參數決定要追蹤的內部類別及套件。除非 IBM 支援中心代表另有指定，否則請將此內容設為值 `com.ibm.wmqfte=all`。

2. 重新產生問題。
3. 透過執行下列指令，關閉代理程式的追蹤：

```
fteSetAgentTraceLevel -traceAgent =off agent_name
```

4. 如果代理程式忙碌，則追蹤檔可能會快速覆蓋並改寫調查問題所需的資訊。
如果您可以在短時間內停止代理程式，請完成下列步驟，以減少所收集的追蹤資料量。否則，請聯絡「IBM 支援中心」並討論替代追蹤規格，以減少所收集的追蹤資料量。
 - a) 排定一些時間來停止代理程式。
 - b) 找出需要追蹤之代理程式的 `agent.properties` 檔案。
`agent.properties` 檔案可在 `BFG_DATA/mqft/config/coordination_qmgr_name/agents/agent_name` 目錄中找到。
 - c) 編輯檔案並新增下列內容的項目：

```
traceFiles=number_of_trace_files_to_use  
traceSize=size_of_each_trace_file_in_MB
```

traceFiles 和 **traceSize** 內容用來控制所收集的追蹤資料量。

traceFiles 內容的預設值為 5，而 **traceSize** 內容的預設值為 20MB。這表示如果您動態開啟追蹤，且尚未設定內容，代理程式會將追蹤資訊寫入 5 個覆蓋追蹤檔，每一個追蹤檔的大小上限為 20MB。

您應該將這些內容設為較大的值，以儘可能收集更多追蹤資料。

例如，若要收集 1GB 折返追蹤，請將下列行新增至 `agent.properties` 檔案：

```
traceFiles=5
traceSize=200
```

這會導致代理程式將追蹤資料寫入最多 5 個檔案，其中每一個檔案的大小都是 200MB。

如需這些代理程式內容的相關資訊，請參閱 [MFT agent.properties](#) 檔案。

- d) 透過執行 **fteStopAgent** 指令，停止代理程式。
- e) 執行 **fteStartAgent** 指令，以啟動代理程式。
- f) 執行下列指令，以開啟代理程式的追蹤：

```
fteSetAgentTraceLevel -traceAgent trace_specification agent_name
```

- g) 重新產生問題。
- h) 透過執行下列指令，關閉代理程式的追蹤：

```
fteSetAgentTraceLevel -traceAgent =off agent_name
```

結果

產生的追蹤檔會寫入 `BFG_DATA/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` 目錄，其中 `%PID%` 是代理程式的處理程序 ID。

z/OS 使用 JCL 動態收集代理程式追蹤

對於需要追蹤的代理程式，您可以在包含 JCL 的資料集內使用 BFGAGTC 成員，以從執行中的 Managed File Transfer for z/OS 代理程式收集追蹤。如果「IBM 支援中心」需要從無法停止的代理程式查看追蹤，這會非常有用。

關於這項作業

若要使用 BFGAGTC 成員從代理程式收集追蹤，請完成下列步驟。

程序

1. 尋找包含需要追蹤之代理程式 JCL 的資料集。
2. 編輯資料集內的 BFGAGTC 成員，並尋找包含下列文字的行：

```
-traceAgent
```

此之後的文字包含要追蹤的內部類別及套件清單。依預設，此清單設為：

```
com.ibm.wmqfte=all
```

除非 IBM 支援中心代表另有指定，否則請依現狀保留此值。

3. 提交 BFGAGTC 成員。
4. 重新產生問題。
5. 再次編輯 BFGAGTC 成員，並將 **-traceAgent** 參數設為 `= off`，如下所示：

```
-traceAgent =off +
```

6. 重新提交 BFGAGTC 成員，以關閉追蹤。

7. 如果代理程式忙碌，追蹤檔可能會快速折返並改寫調查問題所需的資訊。

在此狀況下，有兩個選項：

a) 第一個選項是：

i) 排定一些時間來停止代理程式。

ii) 找出需要追蹤之代理程式的 `agent.properties` 檔案。`agent.properties` 檔案可在 `BFG_DATA/mqft/config/coordination_qmgr_name/agents/agent_name` 目錄中找到。

iii) 編輯檔案並新增下列內容的項目：

- **traceFiles**=要使用的追蹤檔數目
- **traceSize**=每一個追蹤檔的大小 (MB)

traceFiles 和 **traceSize** 內容用來控制所收集的追蹤資料量。

traceFiles 內容的預設值為 5，而 **traceSize** 內容的預設值為 20MB。這表示如果您動態開啟追蹤，且尚未設定內容，代理程式會將追蹤資訊寫入 5 個覆蓋追蹤檔，每一個追蹤檔的大小上限為 20MB。

您應該將這些內容設為較大的值，以儘可能收集更多追蹤資料。

例如，若要收集 1GB 折返追蹤，請將下列行新增至 `agent.properties` 檔案：

```
traceFiles=5
traceSize=200
```

這會導致代理程式將追蹤資料寫入最多 5 個檔案，其中每一個檔案的大小都是 200MB。

如需這些代理程式內容的相關資訊，請參閱 [MFT agent.properties](#) 檔案。

iv) 尋找包含需要追蹤之代理程式 JCL 的資料集。

v) 提交資料集內的 BFGAGSP 成員以停止代理程式。

vi) 透過提交資料集中的 BFGAGST 成員，重新啟動代理程式。

vii) 編輯資料集內的 BFGAGTC 成員，並尋找包含下列文字的行：

```
-traceAgent
```

此之後的文字包含要追蹤的內部類別及套件清單。依預設，此清單設為：

```
com.ibm.wmqfte=all
```

除非 IBM 支援中心代表另有指定，否則請依現狀保留此值。

viii) 當要啟用追蹤時，請提交 BFGAGTC 成員。

ix) 重新產生問題。

x) 再次編輯 BFGAGTC 成員，並將 **-traceAgent** 參數設為 `= off`，如下所示：

```
-traceAgent =off +
```

xi) 重新提交 BFGAGTC 成員，以關閉追蹤。

b) 第二個選項是聯絡「IBM 支援中心」代表 (如果無法在短時間內停止代理程式)。

然後，您可以討論要使用的替代追蹤規格，以減少所產生的追蹤資料量。

結果

產生的追蹤檔會寫入 `BFG_DATA/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` 目錄，其中 `%PID%` 是代理程式的處理程序 ID。

z/OS 追蹤 Managed File Transfer for z/OS 指令

提供 Managed File Transfer for z/OS 中的追蹤功能，以協助「IBM 支援中心」診斷您的問題。您可以使用此功能來追蹤指令。

關於這項作業



小心: 追蹤指令只會收集指令所執行之處理程序的相關資訊。它不會追蹤代理程式在處理該指令時可能執行的任何活動。

收集追蹤的方式取決於指令是使用 Unix 系統服務 (USS) 還是 JCL 來執行。

如果您不確定要使用哪個選項，請聯絡「IBM 支援中心」代表，他們會建議您針對您所看到的問題收集追蹤資料的最佳方式。

程序

- 如果您使用 z/OS UNIX，請參閱第 239 頁的『[使用 USS 從啟動收集代理程式追蹤](#)』。
- 如果您使用 JCL，請參閱第 240 頁的『[使用 JCL 從啟動收集代理程式追蹤](#)』。

z/OS 使用 USS 收集指令的追蹤

若要使用 Unix 系統服務 (USS) 來收集 Managed File Transfer for z/OS 指令的追蹤，請執行下列程序。

程序

1. 啟動命令提示字元，並導覽至 `BFG_PROD/bin` 目錄。
2. 執行指令：

```
./command_name -trace classes=level -tracePath directory_path command_arguments
```

其中

- `command_name` 是要追蹤的指令名稱。
- `classes=level` 是要使用的追蹤層次，以及要啟用追蹤的類別。除非 IBM 支援代表另有指定，否則請將此設為 `com.ibm.wmqfte=all`。
- `directory_path` 是將寫入追蹤檔的目錄。
- `command_arguments` 是需要傳遞給指令的引數，例如 `ftePingAgent` 指令的代理程式名稱。

結果

產生的追蹤檔會寫入 `-tracePath` 參數指定的目錄中。

追蹤檔案稱為 `trace%PID%.txt.number`，其中：

- `%PID%` 是指令的處理程序 ID。
- `number` 是追蹤檔的序號。通常，指令所產生的追蹤資訊包含在序號為 0 的單一追蹤檔中。

不過，指令可能會產生許多追蹤資訊。在此狀況下，追蹤會寫入多個檔案。現行追蹤檔的序號為 0，下一個最舊追蹤檔的序號為 1，依此類推。

指令的追蹤輸出會寫入最多五個折返追蹤檔。每一個追蹤檔的大小上限是 20MB。

註: 如果執行指令的使用者無權寫入 `-tracePath` 參數指定的目錄，則追蹤輸出會寫入標準錯誤。

範例

在此範例中，會追蹤 `fteListAgents` 指令，並將追蹤寫入 `/u/fteuser` 目錄：

```
./fteListAgents -trace com.ibm.wmqfte=all -tracePath /u/fteuser
```

在此範例中，會追蹤 **fteCreateTransfer** 指令，並將追蹤寫入 /tmp 目錄：

```
./fteCreateTransfer -trace com.ibm.wmqfte=all -tracePath /tmp -t text -sa AGENT1  
-da AGENT2 -df /tmp/IEEUJV.txt "'SYS1.SAMPLIB(IEEUJV)'"
```

寫入 /tmp 的追蹤檔只包含 **fteCreateTransfer** 指令所執行之處理程序的相關資訊，例如指令如何建置傳送至代理程式的傳送要求訊息，以及它等待代理程式傳回確認通知的時間長度，指出它已收到要求。追蹤檔不包含任何關於傳送本身的資訊。

使用 JCL 收集指令的追蹤

若要收集使用 JCL 提交之 Managed File Transfer for z/OS 指令的追蹤資料，您需要完成下列步驟。

程序

1. 尋找包含需要追蹤之指令的 JCL 的資料集。
2. 在資料集內，尋找該指令的成員。
3. 編輯成員，並尋找包含需要追蹤之指令名稱的行。修改此行，使其在指令名稱之後及 + 符號之前包含文字：

```
-trace classes=level -tracePath directory_path
```

其中：

- *classes=level* 是要使用的追蹤層次，以及要啟用追蹤的類別。除非 IBM 支援代表另有指定，否則請將此設為 *com.ibm.wmqfte=all*。
 - *directory_path* 是將寫入追蹤檔的 USS 目錄。
4. 提交成員。
 5. 重新產生問題之後，重新編輯成員並移除文字：

```
-trace classes=level -tracePath directory_path
```

您在步驟 [第 245 頁的『3』](#) 中新增加的。

結果

產生的追蹤檔會寫入 **-tracePath** 參數指定的目錄中。

追蹤檔案稱為 *trace%PID%.txt.number*，其中：

- *%PID%* 是指令的處理程序 ID。
- *number* 是追蹤檔的序號。通常，指令所產生的追蹤資訊包含在序號為 0 的單一追蹤檔中。

不過，指令可能會產生許多追蹤資訊。在此狀況下，追蹤會寫入多個檔案。現行追蹤檔的序號為 0，下一個最舊追蹤檔的序號為 1，依此類推。

指令的追蹤輸出會寫入最多五個折返追蹤檔。每一個追蹤檔的大小上限是 20MB。

註：如果執行指令的使用者無權寫入 **-tracePath** 參數指定的目錄，則追蹤輸出會寫入標準錯誤。

範例

在此範例中，成員 BFGMNL1 已修改為追蹤 **fteListMonitors** 指令：

```
/******  
/* <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
/*
//

```

提交成員時，**fteListMonitors** 指令會將追蹤寫入 USS 目錄 /u/trace。

追蹤 *Managed File Transfer for z/OS* 獨立式資料庫日誌程式

提供 *Managed File Transfer for z/OS* 中的追蹤功能，以協助「IBM 支援中心」診斷您的問題。各種指令及內容會控制此機能的行為。

關於這項作業

如果您要求您提供追蹤輸出來調查獨立式資料庫日誌程式的問題，請使用下列其中一個選項。

如果您不確定要使用哪個選項，請聯絡「IBM 支援中心」代表，他們會建議您針對您所看到的問題收集追蹤資料的最佳方式。

程序

- 如果您可以在短時間內停止日誌程式，請從啟動開始收集日誌程式的追蹤資料。
如需相關資訊，請參閱第 246 頁的『從啟動開始收集 *Managed File Transfer for z/OS* 獨立式資料庫日誌程式追蹤』。
- 如果無法停止日誌程式，請使用 **fteSetLoggerTraceLevel** 指令動態收集追蹤。
如需相關資訊，請參閱第 248 頁的『動態收集 *Managed File Transfer for z/OS* 獨立式資料庫日誌程式追蹤』。

從啟動開始收集 *Managed File Transfer for z/OS* 獨立式資料庫日誌程式追蹤

如果您可以在短時間內停止日誌程式，則應該從啟動開始收集 IBM MQ *Managed File Transfer* 日誌程式追蹤。

關於這項作業

收集追蹤的方式取決於日誌程式是使用「UNIX 系統服務 (USS)」還是 JCL 來管理。

如果您不確定要使用下列哪一個選項，請聯絡「IBM 支援中心」代表，他們會建議您針對您所看到的問題收集追蹤資料的最佳方式。

程序

- 如果您使用：
 - UNIX 系統服務 (USS)，請參閱第 246 頁的『使用 USS 從啟動收集獨立式資料庫日誌程式追蹤』。
 - JCL，請參閱第 247 頁的『使用 JCL 從啟動收集獨立式資料庫日誌程式追蹤』。

使用 USS 從啟動收集獨立式資料庫日誌程式追蹤

若要從啟動開始收集使用「Unix 系統服務 (USS)」來管理之 *Managed File Transfer for z/OS* 日誌程式的追蹤，您需要在該日誌程式啟動之前，在 `logger.properties` 檔案中設定該日誌程式的各種內容。

關於這項作業

若要從啟動收集追蹤，請完成下列步驟。

程序

1. 找出需要追蹤之日誌程式的 `logger.properties` 檔。
`logger.properties` 檔案可在 `BFG_DATA/mqft/config/coordination_qmgr_name/loggers/logger_name` 目錄中找到。
2. 編輯檔案並新增下列內容的項目：
 - **trace**=追蹤規格
trace 內容會決定要追蹤的內部類別及套件。除非 IBM 支援中心代表另有指定，否則請將此內容設為值 `com.ibm.wmqfte=all`。
 - **traceFiles**=要使用的追蹤檔數目
 - **traceSize**=每一個追蹤檔的大小 (MB)
traceFiles 和 **traceSize** 內容用來控制所收集的追蹤資料量。您應該將這些內容設為較大的值，以儘可能收集更多追蹤資料。

例如，若要使用追蹤規格 `com.ibm.wmqfte=all` 收集 1GB 折返追蹤，請將下列行新增至 `logger.properties` 檔案：

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```


這會導致日誌程式將追蹤資料寫入最多 5 個檔案，其中每一個檔案的大小都是 200MB。

如需這些日誌程式內容的相關資訊，請參閱 `MFT logger.properties` 檔案。

3. 使用 **`fteStopLogger`** 指令，停止需要追蹤之日誌程式。
4. 執行 **`fteStartLogger`** 指令，以啟動日誌程式。
5. 重新產生問題。
6. 停止日誌程式。
7. 編輯日誌程式的 `logger.properties` 檔案，並移除您在步驟 [第 247 頁的『2』](#) 中新增之 **`trace`**、**`traceFiles`** 及 **`traceSize`** 內容的項目。
這可確保下次重新啟動日誌程式時不會啟用追蹤。

結果

產生的追蹤檔會寫入 `BFG_DATA/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%` 目錄，其中 `%PID%` 是日誌程式的程序 ID。

 使用 JCL 從啟動收集獨立式資料庫日誌程式追蹤

若要收集從啟動開始使用 JCL 來管理之 Managed File Transfer for z/OS 日誌程式的追蹤，您需要在該日誌程式啟動之前，在 `logger.properties` 檔案中設定該日誌程式的各種內容。

關於這項作業

若要從啟動收集追蹤，請完成下列步驟。

程序

1. 找出需要追蹤之日誌程式的 `logger.properties` 檔。
`logger.properties` 檔案可在 `BFG_DATA/mqft/config/coordination_qmgr_name/loggers/logger_name` 目錄中找到。
2. 編輯檔案並新增下列內容的項目：

- **trace**=追蹤規格

trace 內容會決定要追蹤的內部類別及套件。除非 IBM 支援中心代表另有指定，否則請將此內容設為值 `com.ibm.wmqfte=all`。

- **traceFiles**=要使用的追蹤檔數目
- **traceSize**=每一個追蹤檔的大小 (MB)

traceFiles 和 **traceSize** 內容用來控制所收集的追蹤資料量。您應該將這些內容設為較大的值，以儘可能收集更多追蹤資料。

例如，若要使用追蹤規格 `com.ibm.wmqfte=all` 收集 1GB 折返追蹤，請將下列行新增至 `logger.properties` 檔案：

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

這會導致日誌程式將追蹤資料寫入最多 5 個檔案，其中每一個檔案的大小都是 200MB。

如需這些日誌程式內容的相關資訊，請參閱 [MFT logger.properties](#) 檔案。

3. 尋找包含需要追蹤之日誌程式 JCL 的資料集。
4. 提交資料集內的 BFGLGSP 成員以停止日誌程式。
5. 透過提交資料集中的 BFGLGST 成員來重新啟動日誌程式。
6. 重新產生問題。
7. 請提交資料集中的 BFGLGSP 成員，以再次停止日誌程式。
8. 編輯日誌程式的 `logger.properties` 檔案，並移除您在步驟 [第 247 頁](#) 的『2』中新增之 **trace**、**traceFiles** 及 **traceSize** 內容的項目。

這可確保下次重新啟動日誌程式時不會啟用追蹤。

結果

產生的追蹤檔會寫入 `BFG_DATA/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%` 目錄，其中 `%PID%` 是日誌程式的程序 ID。

動態收集 Managed File Transfer for z/OS 獨立式資料庫日誌程式追蹤

如果您無法在短時間內停止日誌程式，則應該動態收集 Managed File Transfer for z/OS 日誌程式追蹤。

關於這項作業

收集追蹤的方式取決於日誌程式是使用 Unix 系統服務 (USS) 還是 JCL 來管理。

如果您不確定要使用下列哪一個選項，請聯絡「IBM 支援中心」代表，他們會建議您針對您所看到的問題收集追蹤資料的最佳方式。

程序

- 如果您使用：
 - Unix 系統服務 (USS)，請參閱 [第 248 頁](#) 的『使用 USS 動態收集獨立式資料庫日誌程式追蹤』。
 - JCL，請參閱 [第 250 頁](#) 的『使用 JCL 動態收集獨立式資料庫日誌程式追蹤』。

使用 USS 動態收集獨立式資料庫日誌程式追蹤

在 (USS) 之下，您可以使用 `fteSetLoggerTraceLevel` 指令來收集執行中日誌程式的追蹤。如果「IBM 支援中心」需要查看無法停止之日誌程式的追蹤，這會非常有用。

關於這項作業

若要使用 `fteSetLoggerTraceLevel` 指令從 Managed File Transfer for z/OS 日誌程式收集追蹤，請完成下列步驟。

程序

1. 執行下列指令，以開啟日誌程式的追蹤：

```
fteSetLoggerTraceLevel -traceLogger classes=level logger_name
```

`-traceLogger` 參數決定要追蹤的內部類別及套件。除非 IBM 支援中心代表另有指定，否則請將此內容設為值 `com.ibm.wmqfte=all`。

2. 重新產生問題。
3. 執行下列指令，以關閉日誌程式的追蹤：

```
fteSetLoggerTraceLevel -traceLogger =off logger_name
```

4. 如果日誌程式忙碌，追蹤檔可能會快速折返，並改寫調查問題所需的資訊。
如果您可以短暫停止日誌程式，請完成下列步驟，以減少所收集的追蹤資料量。否則，請聯絡「IBM 支援中心」並討論替代追蹤規格，以減少所收集的追蹤資料量。
 - a) 請排定一些時間來停止日誌程式。
 - b) 找出需要追蹤之日誌程式的 `logger.properties` 檔。
`logger.properties` 檔案可在 `BFG_DATA/mqft/config/coordination_qmgr_name/loggers/logger_name` 目錄中找到。
 - c) 編輯檔案並新增下列內容的項目：

```
traceFiles=number_of_trace_files_to_use  
traceSize=size_of_each_trace_file_in_MB
```

traceFiles 和 **traceSize** 內容用來控制所收集的追蹤資料量。

traceFiles 內容的預設值為 5，而 **traceSize** 內容的預設值為 20MB。這表示如果您動態開啟追蹤，且尚未設定內容，代理程式會將追蹤資訊寫入 5 個覆蓋追蹤檔，每一個追蹤檔的大小上限為 20MB。

您應該將這些內容設為較大的值，以儘可能收集更多追蹤資料。

例如，若要收集 1GB 折返追蹤，請將下列行新增至 `logger.properties` 檔案：

```
traceFiles=5  
traceSize=200
```

這會導致日誌程式將追蹤資料寫入最多 5 個檔案，其中每一個檔案的大小都是 200MB。

如需這些日誌程式內容的相關資訊，請參閱 `MFT logger.properties` 檔案。

- d) 執行 **fteStopLogger** 指令來停止日誌程式。
- e) 執行 **fteStartLogger** 指令，以啟動日誌程式。
- f) 執行下列指令，以開啟日誌程式的追蹤：

```
fteSetLoggerTraceLevel -traceLogger trace_specification logger_name
```

- g) 重新產生問題。
- h) 執行下列指令，以關閉日誌程式的追蹤：

```
fteSetLoggerTraceLevel -traceLogger =off logger_name
```

結果

追蹤檔會寫入 `BFG_DATA/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%` 目錄，其中 `%PID%` 是日誌程式的處理程序 ID。

對於需要追蹤的日誌程式，您可以在包含 JCL 的資料集內使用 BFGLGTC 成員，以從執行中的 Managed File Transfer for z/OS 日誌程式收集追蹤。如果「IBM 支援中心」需要查看無法停止之日誌程式的追蹤，這會非常有用。

關於這項作業

若要使用 BFGLGTC 成員從日誌程式收集追蹤，請完成下列步驟。

程序

1. 尋找包含需要追蹤之日誌程式 JCL 的資料集。
2. 編輯資料集內的 BFGLGTC 成員，並尋找包含下列文字的行：

```
-traceLogger
```

此之後的文字包含要追蹤的內部類別及套件清單。依預設，此清單設為：

```
com.ibm.wmqfte=all
```

除非 IBM 支援中心代表另有指定，否則請依現狀保留此值。

3. 提交 BFGLGTC 成員。
4. 重新產生問題。
5. 重新編輯 BFGLGTC 成員，並將 **-traceLogger** 參數設為 `= off`，如下所示：

```
-traceLogger =off +
```

6. 重新提交 BFGLGTC 成員，以關閉追蹤。
7. 如果日誌程式忙碌，追蹤檔可能會快速折返，並改寫調查問題所需的資訊。
在此狀況下，有兩個選項：

a) 第一個選項是：

- i) 請排定一些時間來停止日誌程式。
- ii) 找出需要追蹤之日誌程式的 `logger.properties` 檔。`logger.properties` 檔案可在 `BFG_DATA/mqft/config/coordination_qmgr_name/loggers/logger_name` 目錄中找到。
- iii) 編輯檔案並新增下列內容的項目：

- **traceFiles**=要使用的追蹤檔數目
- **traceSize**=每一個追蹤檔的大小 (MB)

traceFiles 和 **traceSize** 內容用來控制所收集的追蹤資料量。

traceFiles 內容的預設值為 5，而 **traceSize** 內容的預設值為 20MB。這表示如果您動態開啟追蹤，且尚未設定內容，日誌程式會將追蹤資訊寫入 5 個覆蓋追蹤檔，每一個追蹤檔的大小上限為 20MB。

您應該將這些內容設為較大的值，以儘可能收集更多追蹤資料。

例如，若要收集 1GB 折返追蹤，請將下列行新增至 `logger.properties` 檔案：

```
traceFiles=5  
traceSize=200
```

這會導致日誌程式將追蹤資料寫入最多 5 個檔案，其中每一個檔案的大小都是 200MB。

如需這些日誌程式內容的相關資訊，請參閱 [MFT logger.properties](#) 檔案。

- iv) 尋找包含需要追蹤之日誌程式 JCL 的資料集。
- v) 提交資料集內的 BFGLGSP 成員以停止日誌程式。

- vi) 透過提交資料集中的 BFGLGST 成員來重新啟動日誌程式。
- vii) 編輯資料集內的 BFGLGTC 成員，並尋找包含下列文字的行：

```
-traceLogger
```

此之後的文字包含要追蹤的內部類別及套件清單。依預設，此清單設為：

```
com.ibm.wmqfte=all
```

除非 IBM 支援中心代表另有指定，否則請依現狀保留此值。

- viii) 當要啟用追蹤時，請提交 BFGLGTC 成員。
- ix) 重新產生問題。
- x) 重新編輯 BFGLGTC 成員，並將 **-traceLogger** 參數設為 *= off*，如下所示：

```
-traceLogger =off +
```

- xi) 重新提交 BFGLGTC 成員，以關閉追蹤。
- b) 第二個選項是如果無法在短時間內停止日誌程式，請聯絡「IBM 支援中心」代表。然後，您可以討論要使用的替代追蹤規格，以減少所產生的追蹤資料量。

結果

產生的追蹤檔會寫入 `BFG_DATA/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%` 目錄，其中 %PID% 是日誌程式的程序 ID。

一般 MFT 問題

在 Managed File Transfer 網路中可能會發生的一般問題。

- 如果文字傳送失敗，且出現下列錯誤：

```
BFGI00060E: Text data conversion has failed
```

可能是因為下列兩個原因之一造成：

1. 來源檔案中有一個以上字元無法從來源檔案字碼頁轉換成目的地檔案字碼頁。當字碼頁有不同字集，而且彼此之間無法進行特定字元轉換時，即會發生此問題。

如果部分字元轉換無法進行是可以接受的，您可以在目的地代理程式上定義取代字元序列，讓傳送不會失敗。請指定代理程式內容 **textReplacementCharacterSequence**，來定義取代字元序列。如需相關資訊，請參閱第 464 頁的表 45。

2. 來源檔案編碼不符合來源代理程式的預設編碼。在此情況下，使用預設值執行文字傳送會毀損字元資料。若要傳送的來源檔案的編碼與來源代理程式的編碼不同，請執行下列其中一個步驟：
 - a. 在傳送定義檔中指定檔案編碼。如需相關資訊，請參閱第 133 頁的『使用傳送定義檔』。
 - b. 搭配使用 **-sce** 參數與 **fteCreateTransfer** 指令，來指定檔案編碼。如需相關資訊，請參閱第 367 頁的『fteCreateTransfer: 啟動新的檔案傳送』主題。
 - c. 指定檔案編碼作為 Ant 移動或複製作業的一部分。如需相關資訊，請參閱第 211 頁的『搭配使用 Apache Ant 與 MFT』。

若要確認已選取正確的來源檔案編碼進行傳送，請執行下列步驟：

1. 將目的地檔案編碼設為 UTF-8。
2. 以文字模式傳送檔案。
3. 使用 UTF-8 檔案檢視器來檢視檔案內容。如果檔案中的所有字元都正確顯示，則來源檔案編碼正確無誤。

- 如果您看到 **fteCreateAgent** 指令有下列輸出：

```
BFGMQ1007I: 無法聯絡協調佇列管理程式，或已拒絕  
connection attempt.  
The IBM MQ reason code was 2058. 不會發佈代理程式的顯示狀態。
```

它指示無法聯絡協調佇列管理程式，並為原因提供 IBM MQ 原因碼。此參考訊息可能指示目前無法使用協調佇列管理程式，或您未正確定義配置。

- 如果您使用的是使用者結束常式，當呼叫使用者結束程式時或就在呼叫結束程式之後發生失敗（例如，產品失敗或斷電），則可能會不止一次呼叫使用者結束程式。
- 如果您在系統上有一個含有佇列管理程式的代理程式，其 IP 位址由 DHCP 指派（而非靜態 IP 位址），且該代理程式是使用用戶端 TCP/IP 連線來連接至該系統，則啟動代理程式時必須設定下列系統環境變數：

–  在 Windows 上：

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

–  在 UNIX 上：

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

其中 *value* 是每一次清除 JVM 的快取 DNS 值的時間間隔，以秒為單位。如果佇列管理程式系統的 IP 位址基於任何理由而重新指派（例如，因為網路中斷、IP 租約到期或系統重新開機），代理程式會向佇列管理程式報告其失去連線。在清除 JVM DNS 快取之後，代理程式可順利重新連接。如果未設定此環境變數，則在此狀況下，若 JVM 未重新啟動，代理程式將無法重新連接。此行為是因為 JVM 在內部快取主機名稱的 IP 位址，但依預設未重新整理它們。

- 如果您執行 **fteStartAgent** 指令時，看到下列錯誤訊息，表示您的環境中可能有其他的程式庫路徑與 Managed File Transfer 衝突：

```
BFGCL0001E: An internal error has occurred. 異常狀況為: 'CC=2;RC=2495;AMQ8568:  
找不到原生 JNI 程式庫 'mqjbnd'. [3=mqjbnd]
```

當代理程式是以 Java 32 位元版本執行時（目前大部分平台都是如此），如果 LD_LIBRARY_PATH 或 LIBPATH 環境變數設為在 32 位元版本之前先參照 64 位元版本程式庫，則會發生此錯誤。

若要解決此問題，請設定 Managed File Transfer 代理程式內容 `javaLibraryPath`，以參照正確的程式庫位置。例如，若為 AIX 上的 `mqjbnd`，請設為：`/usr/mqm/java/lib`。若為 Linux 上的 `mqjbnd`，請設為：`/opt/mqm/java/lib`

- 如果您已在代理程式內容檔中指定 `authorityChecking=true` 來啟用使用者權限檢查，但所有權限檢查都失敗（即使該使用者對相關的權限佇列具有必要權限也一樣）：
 - 請確定執行代理程式的使用者對代理程式佇列管理程式具有 ALT_USER 存取控制。
- 如果您已透過在代理程式內容檔中指定 `authorityChecking=true` 來啟用使用者權限檢查，並將 IBM MQ 錯誤訊息寫入代理程式 `output0.log` 檔案中，請執行下列其中一個動作：
 - 忽略訊息，代理程式不受影響。
 - 授與執行代理程式的使用者對屬於該代理程式的 SYSTEM.FTE.AUTH* 佇列的 GET 權限。
- 如果您已編輯代理程式內容檔，但代理程式未採用它們：
 - 請重新啟動代理程式，以確保代理程式讀取新內容。

z/OS

- 如果您在 z/OS 上使用代理程式來傳送至 PDS 或 PDSE 資料集，但發生異常終止，則系統磁碟空間可能有限。此異常終止可能會有系統完成碼 B14 及回覆碼 0C，指出沒有可用空間。

如果您是傳送至循序資料集，則傳送失敗，並指出空間不足狀況，但代理程式仍然可以運作。
- 如果您在 z/OS 上使用代理程式，而且 WMQFTEP 作業在產生了一些 Java 核心傾出之後變成無回應，請套用 OMVS 系統服務 APAR OA43472。
- 如果在 z/OS 上執行配置或管理 Script 時看到下列輸出：

```
FSUM7332 syntax error: got (, expecting Newline
```

此輸出指出在執行配置或管理 Script 的環境中未設定環境變數 `_BPXK_AUTOCVT=ON`。如需此環境變數以及如何進行設定的相關資訊，請參閱第 73 頁的『z/OS 上 MFT 的環境變數』。

JZOS 的一般 MFT 問題

如果您遇到 JZOS 的問題，以下是一些建議。

- 如果 JZOS 無法順利處理，請執行下列動作：
 - 將 `PARM='+T'` 新增至 JCL。例如：

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

- 將 `set -x` 新增至環境檔案。
- 如果結果為：
 - JVMJZBL1038E 子 Shell 程序結束，並出現結束碼：1
 - JVMJZBL1042E JZOS 批次啟動程式失敗，回覆碼=102

這表示您的環境檔案和 Managed File Transfer 指令有問題。這可能是因為指定的路徑無效。

- 從您的環境檔案中找到 **BFG_PROD** 的值。

1. 進入 OMVS 並使用 `ls -ltr` 指令。

比方說，如果 **BFG_PROD** 為 `/HMF8800/`，請鍵入下列指令：

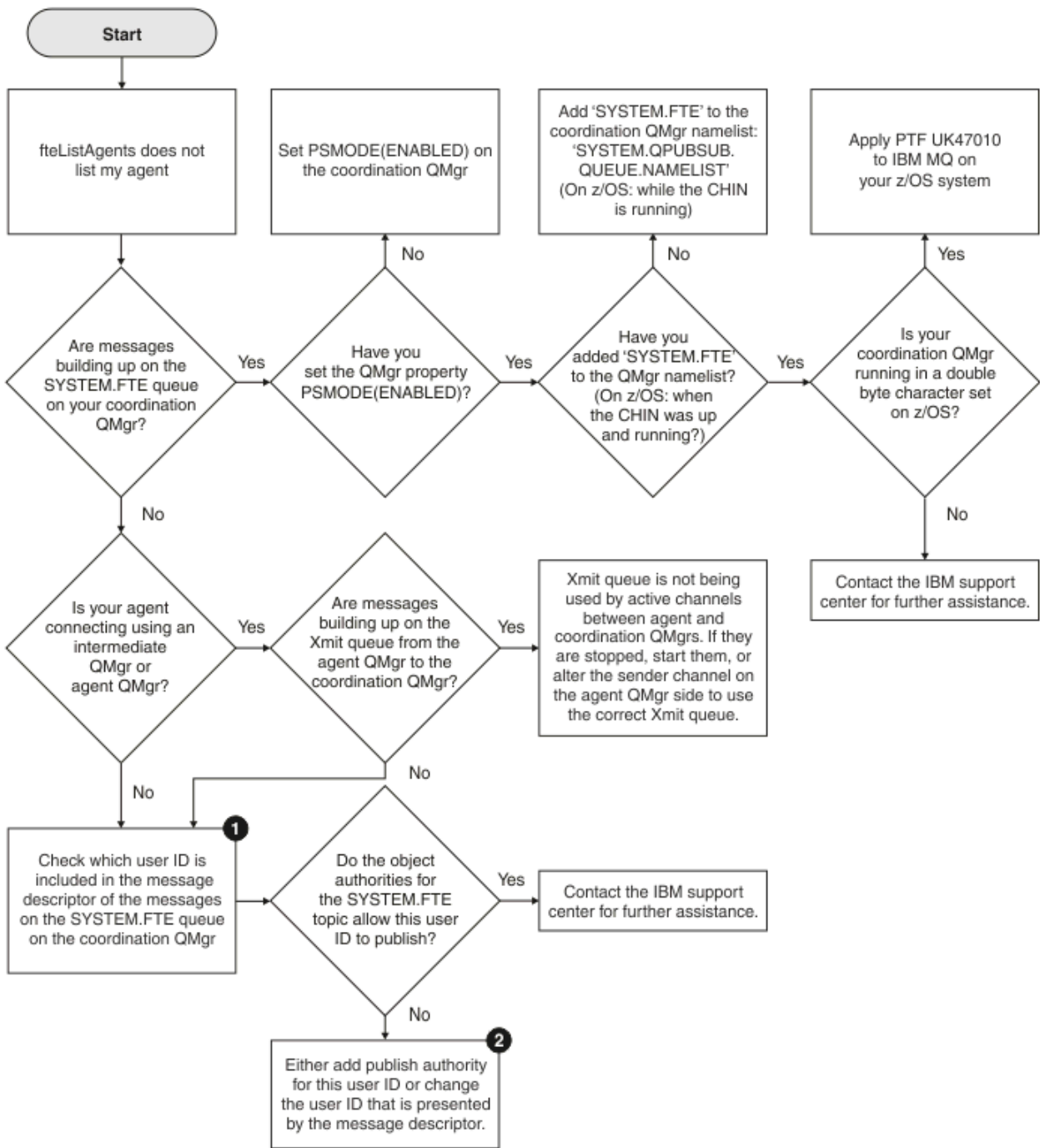
```
ls -ltr HMF8800/bin/fteBatch
```

2. 檢查此檔案是否存在，以及批次工作對該檔案是否具有讀取權。
 3. 解決所有問題。
- 如果 JCL 仍然無法正確處理，請執行下列動作：
 1. 在 USS 中建立檔案（例如，`myenv`），然後使用編輯器將環境檔案中的資訊複製到這個 `myenv` 檔案。
 2. 儲存此檔案。
 3. 從指令行中使用 `chmod +x myenv` 指令，以容許執行此檔案。
 4. 發出 `. myenv` 指令。請注意，這是（點空格檔名）。
 - 執行此指令會報告 `myenv` 檔案中的任何錯誤。
 5. 更正 `myenv` 和環境檔案中的任何錯誤。

fteListAgents 指令未列出您的 MFT 代理程式時應該如何處理

如果您的代理程式未由 **fteListAgents** 指令列出或未顯示在 IBM MQ Explorer 中，或您的檔案傳送未顯示在 IBM MQ Explorer 的 **傳送日誌** 中，您可以執行許多問題判斷步驟來調查原因。

使用下列流程圖有助於您診斷問題並決定應採取的後續動作：



流程圖註解：

- 如需如何檢查呈現的使用者 ID 的相關資訊，請參閱第 268 頁的『在發佈之前檢查訊息』。使用者 ID 必須符合 MQ 使用者名稱 12 個字元的限制。如果使用者名稱長度超過 12 個字元（例如，Administrator），則會截斷使用者名稱，然後再檢查其授權。在使用 Administrator 的範例中，會將下列錯誤訊息新增至佇列管理程式錯誤日誌：

AMQ8075: Authorization failed because the SID for entity 'administrato' cannot be obtained.

- 如需 SYSTEM.FTE 佇列所需權限的相關資訊，請參閱第 305 頁的『發佈 MFT 代理程式日誌及狀態訊息的權限』。

若 MFT 代理程式程序消失，但未記載任何診斷資訊時應該如何處理

在 UNIX 平台上，如果代理程式程序消失，但代理程式日誌檔卻沒有任何說明，這可能是因為代理程式的啟動方式所致。

您可以透過下列方式檢查代理程式診斷資訊：

- 檢查代理程式的日誌檔是否指出代理程式已停止。
- 檢查代理程式鎖定檔 `agent.lck` 是否仍存在。

比方說，例如您是從 Shell Script 啟動代理程式，則在 Script 完成時，所有與該 Script 相關聯的子程序都會被移除（包括代理程式程序）。若要讓代理程式在呼叫它的 Script 終止後仍持續執行，請執行下列步驟：

1. 在 **fteStartAgent** 指令前面加上 **nohup** 指令，以解除 **fteStartAgent** 程序（及任何子程序）與 Script 之間的關聯。

日後當 Script 終止時，代理程式仍可繼續執行。

代理程式顯示為處於 UNKNOWN 狀態時要執行的動作

您的代理程式正在執行中並順利回應 **ftePingAgent** 指令，且項目正在正常傳送。不過，**fteListAgents** 和 **fteShowAgentDetails** 指令以及 IBM MQ Explorer Managed File Transfer 外掛程式會將代理程式報告為處於 UNKNOWN 狀態。

發生此問題的原因

每一個代理程式會定期將其狀態發佈至 SYSTEM.FTE 主題。代理程式發佈其狀態的頻率由下列代理程式內容控制：

agentStatusPublishRateLimit

代理程式由於檔案傳送狀態變更而重新發佈其狀態的速率上限 (秒)。此內容的預設值為 30 秒。

agentStatusPublishRateMin

代理程式發佈其狀態的速率下限 (秒)。此值必須大於或等於 **agentStatusPublishRateLimit** 內容的值。**agentStatusPublishRateMin** 內容的預設值為 300 秒 (或 5 分鐘)。

fteListAgents 和 **fteShowAgentDetails** 指令以及 IBM MQ Explorer Managed File Transfer (MFT) 外掛程式使用這些出版品來判定代理程式的狀態。為了執行此動作，指令和外掛程式會執行下列步驟：

1. 連接至協調佇列管理程式。
2. 訂閱 SYSTEM.FTE 主題。
3. 接收代理程式狀態發佈。
4. 在協調佇列管理程式上建立暫時佇列。
5. 將訊息放入暫時佇列，並儲存放置時間，以取得協調佇列管理程式系統上的現行時間。
6. 關閉暫時佇列。
7. 使用出版品中包含的資訊及現行時間，以判定代理程式的狀態。
8. 中斷與協調佇列管理程式的連線。

如果代理程式的發佈時間與現行時間之間的差異大於下列值，則會將代理程式的狀態訊息視為過時：代理程式內容的值 **agentStatusPublishRateMin** (包括在狀態訊息中) 加上進階協調佇列管理程式內容的值 **agentStatusJitterTolerance**。

依預設，**agentStatusJitterTolerance** 內容具有值 3000 毫秒 (3 秒)。

如果 **agentStatusPublishRateMin** 及 **agentStatusJitterTolerance** 內容設為其預設值，則當代理程式發佈的時間與現行時間之間的差異大於 303 秒 (或 5 分 3 秒) 時，會將代理程式的狀態視為過時。

任何具有陳舊狀態訊息的代理程式，都會由 **fteListAgents** 和 **fteShowAgentDetails** 指令以及 IBM MQ Explorer MFT 外掛程式報告為處於 UNKNOWN 狀態。

代理程式的狀態發佈可能由於下列其中一個原因而過時：

1. 代理程式佇列管理程式執行所在系統與協調佇列管理程式所在系統之間的系統時間差異很大。
2. 代理程式佇列管理程式與協調佇列管理程式之間的通道已停止 (這會防止新的狀態訊息到達協調佇列管理程式)。
3. 授權問題導致代理程式無法將其狀態發佈至 SYSTEM.FTE 主題。
4. 發生代理程式失敗。

對問題進行疑難排解

需要採取一些步驟來判斷代理程式狀態報告為 UNKNOWN 的原因:

1. 透過登入代理程式系統，檢查代理程式是否在執行中。如果代理程式已停止，請調查代理程式不再執行的原因。再次執行之後，請檢查其狀態現在是否已正確報告。
2. 請檢查協調佇列管理程式是否在執行中。如果不是，請重新啟動它，然後使用 **fteListAgents** 或 **fteShowAgentDetails** 指令或 IBM MQ Explorer MFT 外掛程式，以查看現在是否正確地報告代理程式狀態。
3. 如果代理程式及協調佇列管理程式正在執行中，請檢查其錯誤日誌，以查看是否有任何授權問題導致代理程式無法發佈其狀態訊息。如果日誌顯示發生授權問題，請確保執行代理程式處理程序的使用者具有將訊息發佈至 SYSTEM.FTE 主題。

如果佇列管理程式的錯誤日誌未報告任何授權問題，請檢查狀態訊息是否未停留在 IBM MQ 網路中。請驗證用來將訊息從代理程式佇列管理程式遞送至協調佇列管理程式的所有傳送端及接收端通道都在執行中。

如果通道在執行中，請檢查與通道相關聯的傳輸佇列，以確定狀態訊息不會停留在通道上。此外，您也應該檢查佇列管理程式的任何無法傳送郵件的佇列，以確定由於某些原因未將狀態訊息放置在該處。

4. 如果通道正在執行中，且狀態訊息正在流經 IBM MQ 網路，則接下來要檢查的是佇列管理程式的佇列發佈/訂閱引擎是否正在挑選訊息。

fteSetupCoordination 指令 (用來定義協調佇列管理程式) 提供您一些 MQSC 指令，這些指令必須在協調佇列管理程式上執行，以配置已排入佇列的發佈/訂閱引擎來接收發佈。這些指令會執行下列步驟:

- 建立 SYSTEM.FTE 主題及其相關聯的主題字串。
- 定義稱為 SYSTEM.FTE 。
- 透過將佇列管理程式上的 **PSMODE** 屬性設為 **ENABLED**，啟用已排入佇列的發佈/訂閱引擎。
- 修改 SYSTEM.QPUBSUB.QUEUE.NAMELIST 名單，由排入佇列的發佈/訂閱引擎使用，因此它包括新 SYSTEM.FTE 佇列。

如需此作業的相關資訊，包括需要執行的 MQSC 指令，請參閱第 433 頁的『[fteSetupCoordination](#)』。

如果 SYSTEM.FTE 佇列，則您應該檢查 SYSTEM.QPUBSUB.QUEUE.NAMELIST 名稱清單已正確設定且包含該佇列的項目。如果遺漏項目，則排入佇列的發佈/訂閱引擎將不會偵測來自代理程式的任何送入狀態訊息，且不會處理它們。

您也應該確定佇列管理程式上的 **PSMODE** 屬性設為 **ENABLED**，這會開啟已排入佇列的發佈/訂閱引擎。

5. 如果通道在執行中，且狀態訊息流經 IBM MQ 網路 並由佇列管理程式的佇列發佈/訂閱引擎從 SYSTEM.FTE 佇列中挑選，請收集下列追蹤資料:
 - 來自代理程式的 IBM MQ MFT 追蹤，涵蓋的時段等於代理程式內容 **agentStatusPublishRateMin** 值的三倍。這可確保追蹤涵蓋代理程式發佈至少三則包含其狀態的訊息的時間。應該使用追蹤規格來動態收集追蹤:

```
com.ibm.wmqfte.statestore.impl.FTEAgentStatusPublisher,  
com.ibm.wmqfte.utils.AgentStatusDetails,  
com.ibm.wmqfte.wmqiface.AgentPublicationUtils,  
com.ibm.wmqfte.wmqiface.RFHMessageFactory=all
```

註: 使用這些字串的輸出會減少追蹤數量。

如需如何對在 IBM MQ for Multiplatforms 上執行的代理程式啟用追蹤的相關資訊，請參閱第 234 頁的『[動態收集 Managed File Transfer 代理程式追蹤](#)』。

如需如何對在 IBM MQ for z/OS 上執行的代理程式啟用追蹤的相關資訊，請參閱 [第 241 頁的『動態收集 Managed File Transfer for z/OS 代理程式追蹤』](#)。

- 用來將狀態訊息從代理程式佇列管理程式遞送至協調佇列管理程式的佇列管理程式並行追蹤。
- **fteListAgents** 指令的追蹤，涵蓋代理程式顯示為處於 UNKNOWN 狀態的時間。應該使用追蹤規格來收集追蹤：

```
com.ibm.wmqfte=all
```

如需如何對在 IBM MQ for Multiplatforms 上執行的指令啟用追蹤的相關資訊，請參閱 [第 235 頁的『在 Multiplatforms 上追蹤 Managed File Transfer 指令』](#)。

如需如何對在 IBM MQ for z/OS 上執行的指令啟用追蹤的相關資訊，請參閱 [第 244 頁的『追蹤 Managed File Transfer for z/OS 指令』](#)。

收集追蹤資料之後，應該會提供給「IBM 支援中心」進行分析。

相關資訊

[fteListAgents](#)

[fteShowAgentDetails](#)

[MFT 代理程式狀態值](#)

[MFT agent.properties 檔案](#)

[MFT coordination.properties 檔](#)

如果 ftePingAgent 逾時並報告 BFGCL0214I 訊息，怎麼辦？

ftePingAgent 是隨 IBM MQ Managed File Transfer 提供的有用指令行公用程式，可讓您檢查代理程式是否可聯繫，以及它是否能夠回應要求。

指令如何運作

您可以使用 **ftePingAgent** 指令來檢查代理程式是否可聯繫，以及它是否能夠處理要求。當指令執行時，它會執行下列步驟：

- 連接至 Managed File Transfer (MFT) 拓撲的指令佇列管理程式。
- 在指令佇列管理程式上建立暫時回覆佇列。

依預設，暫時佇列的名稱以字首 WMQFTE 開頭。不過，您可以透過在 [MFT command.properties 檔](#) 中設定安裝架構的 **dynamicQueuePrefix** 內容來變更此設定。

- 將連線測試 MFT 代理程式要求訊息傳送至佇列 SYSTEM.FTE.COMMAND.agent_name (透過指令佇列管理程式)。要求訊息包含暫時回覆佇列的名稱。
- 等待回覆訊息到達暫時回覆佇列。

代理程式內的其中一個執行緒是 CommandHandler。此執行緒會從 SYSTEM.FTE.COMMAND.agent_name 佇列，並處理它們。

如果此執行緒收到包含連線測試 MFT 代理程式要求的訊息，則會建置 MFT 代理程式回覆訊息，並將它傳送至指令佇列管理程式上的暫時佇列。此訊息會通過代理程式的佇列管理程式。

一旦訊息到達暫時佇列，**ftePingAgent** 指令即會挑選該訊息。然後，在結束之前，指令會將如下所示的訊息寫入主控台：

```
BFGCL0213I: agent <agent_name> responded to ping in 0.088 seconds.
```

下列兩個圖表顯示流程：

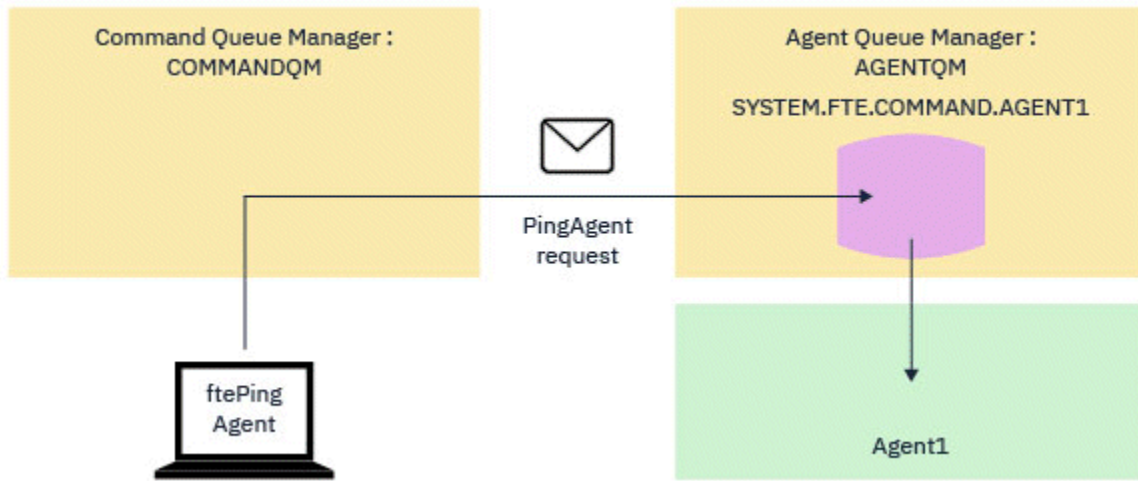


圖 7: pingAgent 要求會移至 SYSTEM.FTE.COMMAND.agent_name 佇列 (透過指令佇列管理程式)

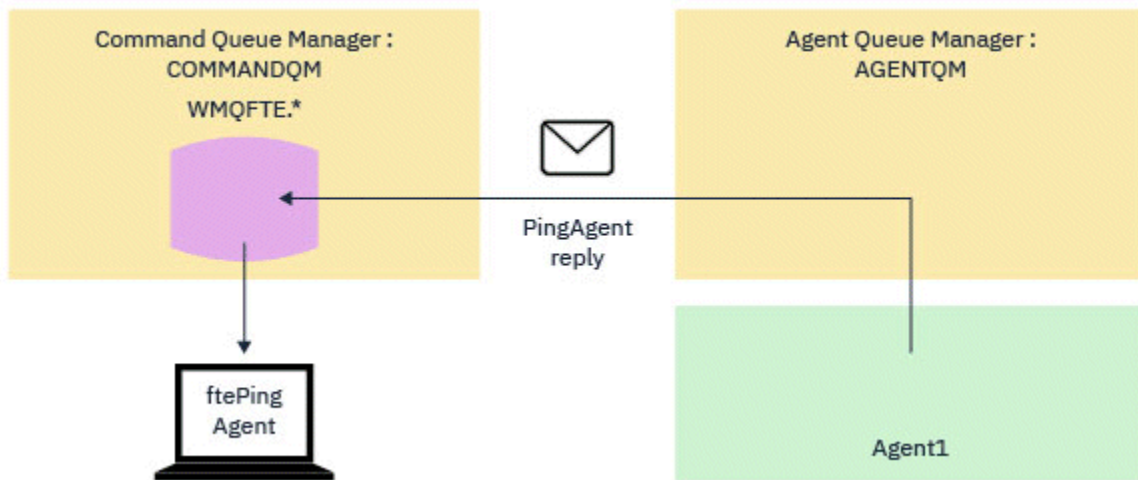


圖 8: pingAgent 回覆會透過代理程式佇列管理程式回到指令佇列管理程式。

指令逾時時要執行的動作

依預設，**ftePingAgent** 指令會等待五秒，讓 MFT 代理程式回覆訊息抵達暫時佇列。如果回覆訊息未在 5 秒內送達，則指令會將 BFGCL0214I 訊息寫入主控台。下列訊息是範例：

BFGCL0214I: agent AGENT1 在 5 秒後未回應連線測試。

您可以使用下列步驟來調查回覆訊息未到達的原因：

- 首先要執行的是檢查代理程式是否在執行中。如果不是，則無法回應指令所傳送的 Ping MFT 代理程式要求。
- 如果代理程式正在執行中且忙於處理要求，則取得 Ping MFT 代理程式要求並傳回回覆可能需要超過 5 秒的時間。

若要查看您是否應該重新執行 **ftePingAgent** 指令，請使用 **-w** 參數來指定較長的等待間隔。例如，若要指定 60 秒等待間隔，請發出下列指令：

```
ftePingAgent -w 60 AGENT1
```

- 如果指令仍逾時，請檢查指令佇列管理程式與代理程式佇列管理程式之間透過 IBM MQ 網路的路徑。如果路徑中有一個以上通道失敗，則連線測試 MFT 代理程式要求訊息及/或 MFT 代理程式回覆訊息將停留在傳輸佇列某處。在此狀況下，您應該重新啟動通道，然後重新執行 **ftePingAgent** 指令。

在您執行之前的步驟之後，如果指令仍報告 BFGCL0214I 訊息，則需要追蹤連線測試 MFT 代理程式要求及 MFT 代理程式回覆訊息，因為它們會流經 IBM MQ 網路來查看：

- 連線測試 MFT 代理程式訊息是否曾經到達 SYSTEM.FTE.COMMAND.agent_name 佇列。
- 如果代理程式從佇列中挑選訊息，並傳回 MFT 代理程式回覆訊息。

若要執行此動作，您應該執行下列步驟：

- 在指令及代理程式佇列管理程式上啟用佇列管理程式追蹤。
- 使用追蹤規格 com.ibm.wmqfte=all 在代理程式上動態啟用追蹤。

執行此動作的方式視代理程式執行所在的平台而定。對於在上執行的代理程式：

- IBM MQ for Multiplatforms，請參閱第 234 頁的『動態收集 Managed File Transfer 代理程式追蹤』。
- IBM MQ for z/OS，請參閱第 241 頁的『動態收集 Managed File Transfer for z/OS 代理程式追蹤』。
- 接下來，使用追蹤規格 com.ibm.wmqfte=all 來執行 **ftePingAgent** 指令並啟用追蹤。如需在上追蹤指令的相關資訊：
 - IBM MQ for Multiplatforms，請參閱第 235 頁的『在 Multiplatforms 上追蹤 Managed File Transfer 指令』。
 - IBM MQ for z/OS，請參閱第 244 頁的『追蹤 Managed File Transfer for z/OS 指令』。

當指令逾時，停止代理程式追蹤及佇列管理程式追蹤。然後，代理程式及佇列管理程式追蹤資料，以及來自指令的追蹤資料，應該可供 IBM 支援中心進行分析。

MFT 代理程式或日誌程式配置不安全時應該如何處理

如果 Managed File Transfer 程序在啟動時偵測到配置檔包含機密性資訊、屬於金鑰儲存庫檔或信任儲存庫檔案，且具有全系統的讀取、寫入或刪除權限的狀況，此程序將無法啟動。如果在啟動時未偵測到此狀況，但是在執行時期偵測到此狀況，則 Managed File Transfer 會產生警告訊息，並忽略配置檔的內容。這關係到通訊協定橋接器及 Connect:Direct 橋接器功能，這會在配置於代理程式執行期間有所變更時予以重新載入。

完成下列檢查，判斷問題的原因：

1. 從提供的錯誤訊息中找出報告為不安全的配置檔。
2. 確定檔案存取權限符合必要的需求。如需相關資訊，請參閱第 304 頁的『存取機密配置資訊的 MFT 權限』。
3. 重新啟動代理程式或日誌程式。如果是通訊協定橋接器或 Connect:Direct 認證檔案的問題，請等待下一次重新載入。

範例

在下列錯誤訊息範例中，資料庫日誌程式將無法啟動：

```
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'.
```

在下列錯誤訊息範例中，通訊協定橋接器代理程式將無法啟動：

```
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'.
```

相關參考

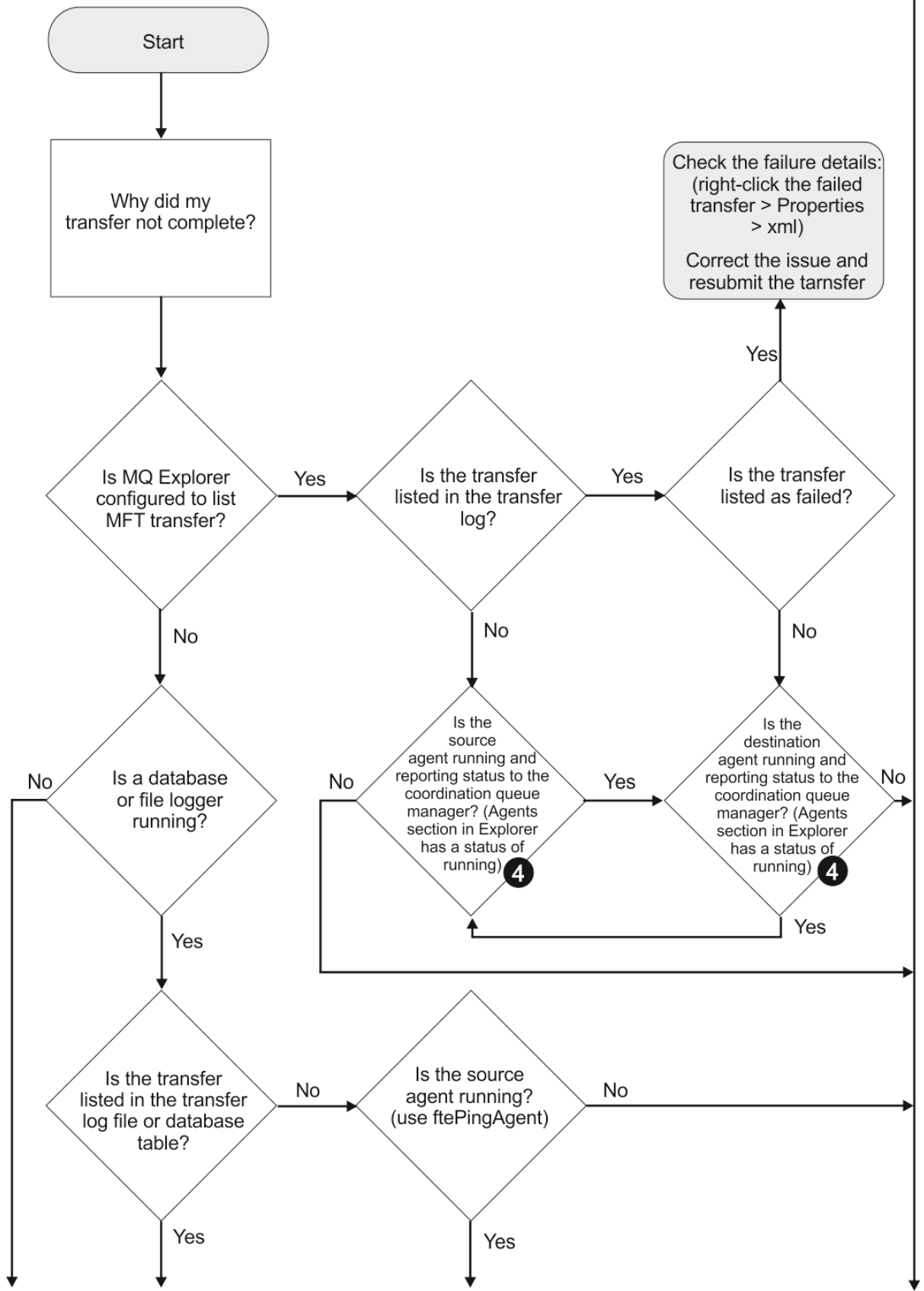
第 304 頁的『存取機密配置資訊的 MFT 權限』

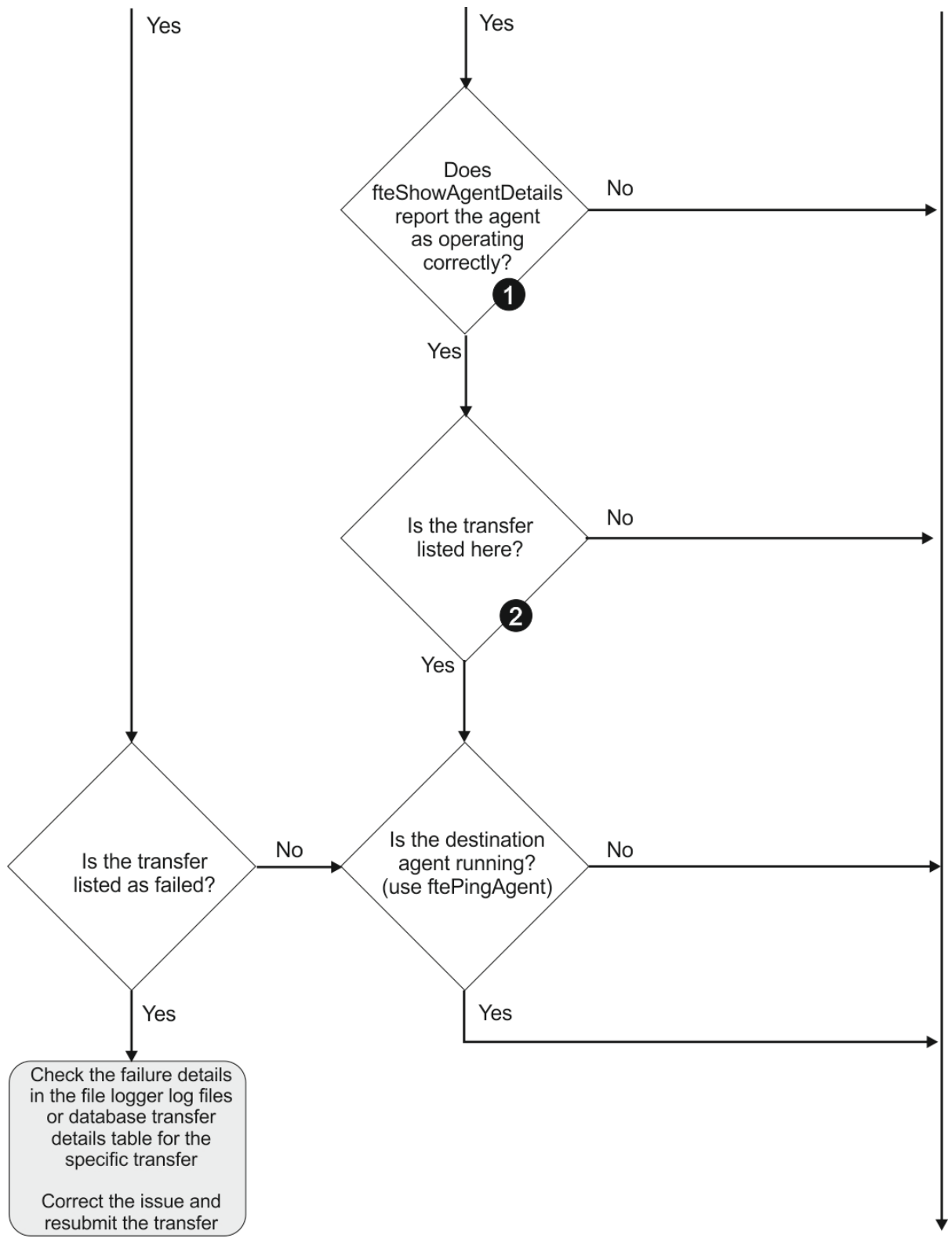
任何用來儲存機密性配置資訊的檔案（即任何利用 IBM MQ 配置樹狀結構參照的檔案）皆不得具有全系統的讀取、寫入或刪除（如果適用）權限。這些限制也適用於信任儲存庫檔案及金鑰儲存庫檔。

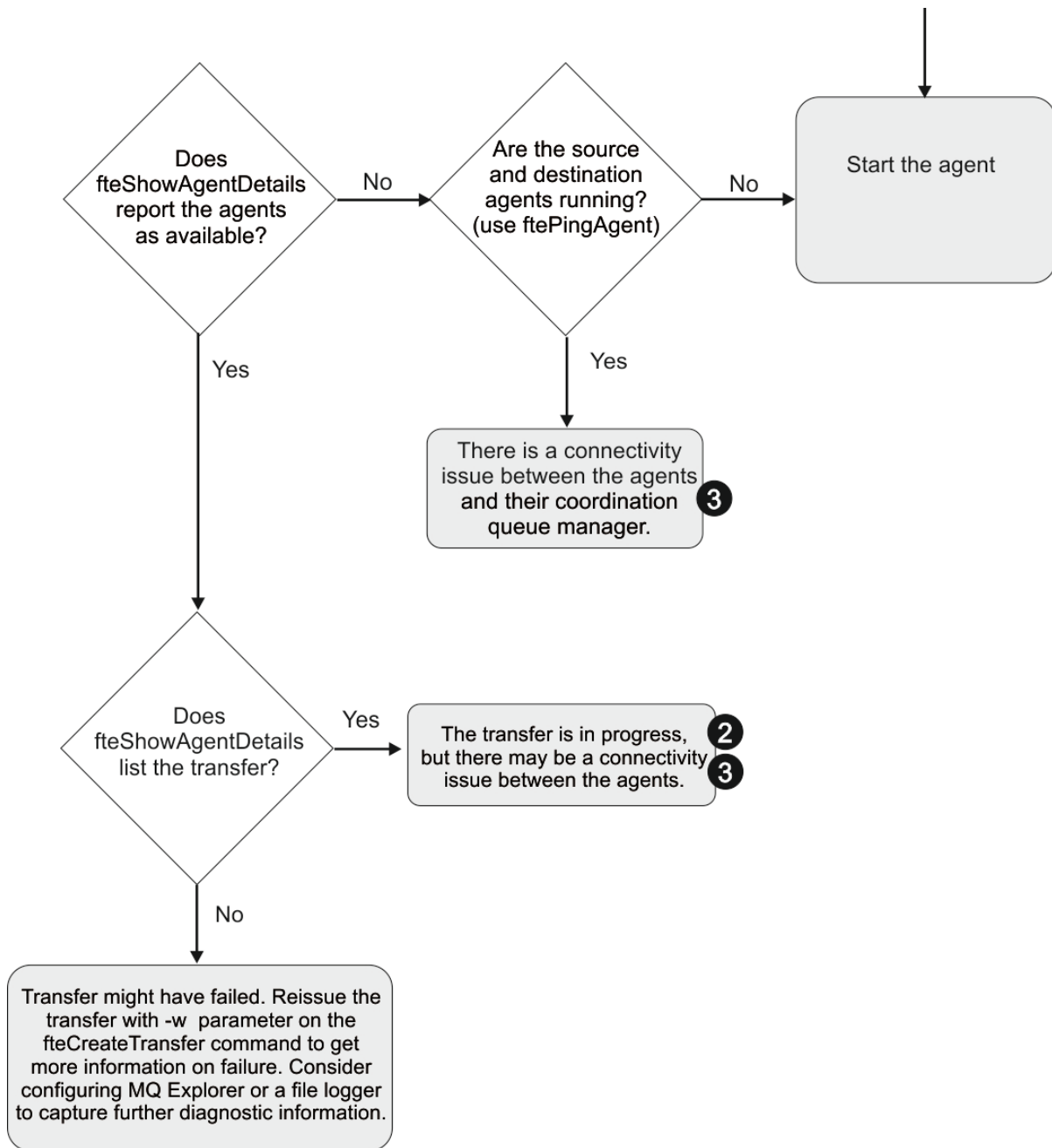
傳送未完成時應該如何處理

如果您的傳送未完成，您可以執行一些問題判斷步驟來調查原因。

使用下列流程圖有助於您診斷問題並決定應採取的後續動作：







流程圖註解：

1. 檢查代理程式 `output0.log` 是否發生錯誤。如果代理程式回報已順利啟動，但「IBM MQ Explorer」或 **fteShowAgentDetails** 皆未回報代理程式在執行中，請檢查代理程式佇列管理程式與協調佇列管理程式之間的連線功能。這可能是因為佇列管理程式通道的佇列管理程式無法使用。
2. 如果來源代理程式將傳送 ID 列為 `In progress` 傳送，但目的地代理程式未列出，則來源與目的地佇列管理程式之間可能存在連線問題。請從目的地代理程式機器，對使用目的地代理程式佇列管理程式作為指令佇列管理程式的來源代理程式，使用 `command.properties` 檔案中的 **ftePingAgent** 指令。您也可以反向從來源對目的地執行此指令。
3. 如果來源及目的地代理程式都將傳送 ID 列為 `In progress`，則表示自起始傳送以來，來源與目的地佇列管理程式之間發生連線問題。請從目的地代理程式機器，對使用目的地代理程式佇列管理程式作為指令佇列管理程式的來源代理程式，使用 `command.properties` 檔案中的 **ftePingAgent** 指令。您也可以反向從來源對目的地執行此指令。
4. 如果您已完成此迴圈，請檢查下列任一陳述是否與您的狀況有關：

- 來源及目的地代理程式都報告為 **Running**，但未列出任何傳送。傳送要求未到達代理程式指令佇列，或代理程式雖然報告為 **Running**，但不再監視指令佇列。請查看來源代理程式 `output0.log` 中的錯誤。請在先前作為傳送來源的機器中，對來源代理程式使用 **ftePingAgent** 指令，以驗證指令佇列管理程式與代理程式佇列管理程式之間的連線功能，以及代理程式是否正為指令佇列提供服務。
- 來源及目的地代理程式都會報告為 **Running**，且傳送會以 **In progress**、**recovering** 列出。請從目的地代理程式機器，對使用目的地代理程式佇列管理程式作為指令佇列管理程式的來源代理程式，使用 `command.properties` 檔案中的 **ftePingAgent** 指令。您也可以反向從來源對目的地執行此指令。

認為檔案傳送停滯時應該如何處理

在負載沉重的系統上，或來源與目的地代理程式之間發生網路問題時，傳送有時會呈現停滯在佇列或回復狀態。有許多因素可能會造成此狀況。

完成下列檢查，判斷問題的原因：

1. 使用 **ftePingAgent** 指令，或者在「IBM MQ Explorer」的代理程式畫面中，用滑鼠右鍵按一下代理程式名稱，然後選取**連線測試**，以檢查來源及目的地代理程式是否在作用中且可回應新的要求。查看代理程式日誌，以檢查目前是否發生網路連線問題。
2. 檢查目的地代理程式是否在能力範圍下執行。可能有許多來源代理程式全部都在要求將傳送檔案至相同的目的地代理程式。搭配使用 **fteShowAgentDetails** 指令與 **-v**（詳細）參數，或者在「IBM MQ Explorer」的代理程式畫面中，用滑鼠右鍵按一下代理程式名稱，然後選取**內容**，以查看代理程式的現行傳送活動。如果執行中目的地傳送數目已達到或接近代理程式的目的地傳送數目上限，則可解釋來源代理程式的部分傳送為何呈現停滯。
3. 如果無法連接通訊協定檔案伺服器，則與通訊協定橋接器代理程式的來回傳送會進入回復狀態。查看代理程式日誌，以檢查目前是否發生連線問題。
4. 代理程式會依優先順序處理傳送。因此，在負載沉重的系統中，當代理程式忙於處理較高優先順序的傳送時，低優先順序的傳送可能會停留在佇列狀態一陣子。即使有較新的高優先順序傳送，已在佇列中停留一陣子的低優先順序傳送最終一定會啟動。

排定的檔案傳送未執行或延遲時應該如何處理

如果排定的傳送已到期但未執行或延遲，可能是因為代理程式正在處理指令佇列上的指令。因為代理程式忙碌中，不會檢查排定的傳送，因此未執行。

若要暫行解決此問題，請使用下列其中一個步驟：

- 在 `agent.properties` 檔中配置 `maxSchedulerRunDelay` 內容，以設定代理程式等待檢查排定傳送的間隔上限（以分鐘為單位）。設定此內容可確保即使代理程式忙碌中，代理程式仍持續檢查排定的傳送。如需此內容的相關資訊，請參閱第 463 頁的『MFT `agent.properties` 檔案』。
- 或者，您也可以使用資源監視器來代替排定傳送。資源監視器的運作不同於排定傳送，不會因為代理程式忙碌而受到影響。比方說，例如您想要取得目的地系統上的最新檔案，資源監視器可減少網路資料流量。這是因為只有在新版本可用時才會傳送檔案，而不會自動傳送檔案。不過，在通訊協定橋接器代理程式或 `Connect:Direct` 橋接器代理程式上，並不支援資源監視。

如需相關資訊，請參閱第 141 頁的『監視 MFT 資源』。

通訊協定橋接器代理程式報告找不到檔案時應該如何處理

當通訊協定橋接器代理程式報告通訊協定橋接器所連接的 SFTP 或 FTP 伺服器傳回 `File not found` 錯誤訊息時，此訊息可能表示發生許多不同的錯誤案例之一。

下列可能的實務範例可能導致 SFTP 或 FTP 伺服器傳回 `File not found` 錯誤。

- 檔案不存在。請確認您嘗試傳送的檔案存在於管理 SFTP 或 FTP 伺服器的系統上。
- 檔案路徑不存在。請確認檔案路徑存在於管理 SFTP 或 FTP 伺服器的系統上。請確認您在傳送要求中輸入正確的檔案路徑。必要的話，請更正檔案路徑，然後重新提交傳送要求。

- 檔案已由另一個應用程式鎖定。請檢查檔案是否被另一個應用程式鎖定。等到檔案不再鎖定之後，再重新提交傳送要求。
- 檔案權限不允許讀取檔案。請檢查檔案是否有正確的檔案權限。必要的話，請變更檔案權限，然後重新提交傳送要求。
- SFTP 或 FTP 伺服器使用虛擬化根路徑。如果在傳送要求中指定了相對檔案路徑，則通訊協定橋接器代理程式將會嘗試根據用來登入通訊協定伺服器的起始目錄，將相對路徑轉換成絕對檔案路徑。Managed File Transfer 通訊協定橋接器代理程式僅支援允許以絕對檔案路徑存取檔案的 SFTP 或 FTP 伺服器。通訊協定橋接器代理程式不支援只能根據現行目錄來存取檔案的通訊協定伺服器。

相關概念

第 178 頁的『通訊協定橋接器』

通訊協定橋接器可供 Managed File Transfer (MFT) 網路存取儲存在 MFT 網路之外的檔案伺服器上的檔案（在本端網域或遠端位置）。此檔案伺服器可使用 FTP、FTPS 或 SFTP 網路通訊協定。每一個檔案伺服器需要至少一個專用的代理程式。專用代理程式即所謂的通訊協定橋接器代理程式。橋接器代理程式可以與多個檔案伺服器互動。

MFT 目錄資源監視器未觸發檔案時應該如何處理

目錄資源監視器會輪詢目錄中是否有符合觸發規格的檔案。系統會為每一個符合觸發規格的檔案，對代理程式產生傳送要求。提交要求之後，將會忽略觸發的檔案，直至變更檔案為止。

檔案未觸發的可能原因

1. 目錄資源監視器找到符合觸發規格的檔案，但產生的傳送要求無效，因此代理程式無法處理要求。原因可能包括：
 - 目的地代理程式無效
 - 遺漏目的地代理程式
 - 程式呼叫已取消傳送

在上述所有範例中，即使傳送失敗，目錄資源監視器還是會將觸發的檔案標示為已處理，並忽略該檔案。

2. 檔案不在資源監視器觸發規格的範圍內。原因可能包括：
 - 觸發型樣不正確
 - 監視不正確的目錄
 - 檔案權限不足
 - 無法連接至遠端檔案系統

檔案可以觸發第二次傳送的原因

觸發檔案可以產生 Managed File Transfer 傳送要求的原因如下：

- 如果偵測到觸發檔案存在（之前並不在該處）。
- 如果觸發檔案已更新，導致前次修改日期變更。

第二次觸發的可能狀況如下：

- 檔案遭移除，然後進行取代。
- 檔案由某個應用程式鎖定，然後由其他應用程式解除鎖定。
- 監視器檔案系統失敗。比方說，例如網路連線失敗，則會出現檔案遭移除，然後進行取代的狀況。
- 檔案目錄已由其他應用程式更新，導致前次修改日期變更。

用來為代理程式的所有資源監視器設定參考層次輸出的指令

V 9.0.3

在此範例中，將監視所有資源監視器，因為您沒有指定特定資源監視器的名稱。代理程式的名稱是 AGENT1。

```
fteSetAgentLogLevel -logMonitor=info AGENT1
```

請參閱 [fteSetAgentLogLevel](#) 以取得 **logMonitor** 參數的詳細資料，及不同選項之用法的範例。

代理程式的所有資源監視器的參考層次輸出範例

V 9.0.3

```
=====
[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
=====
```

相關參考

第 421 頁的『[fteSetAgentLogLevel: 設定 MFT 通訊協定橋接器代理程式記載層次](#)』

使用 **fteSetAgentLogLevel** 指令，針對 Managed File Transfer 通訊協定橋接器代理程式與 FTP/SFTP/FTPS 檔案伺服器之間的檔案傳送，啟用或停用診斷資訊記載。

由佇列資源監視器啟動的傳送所建立的目的地檔案包含錯誤資料時應該如何處理

您可以建立資源監視器來監視佇列，並將佇列上的訊息或訊息群組傳送至檔案。使用該訊息或該訊息群組的第一個訊息上的 MQMD 訊息描述子來指定檔案名稱。如果「訊息轉為檔案」的傳送失敗，且訊息或訊息群組留在佇列上，則下次觸發監視器時，它可能導致所建立的檔案包含錯誤資料。

發生此問題的原因

1. 「訊息轉為檔案」的傳送失敗，且該訊息或該訊息群組留在佇列上。
2. 新訊息或新訊息群組抵達佇列。
3. 新訊息或新訊息群組觸發資源監視器。
4. 資源監視器建立新的傳送，其使用來自新訊息或新訊息群組的 MQMD 訊息描述子或來自佇列上第一個訊息或訊息群組的資料。
5. 建立包含錯誤資料的檔案。

避免此問題

若要避免遇到此問題，您必須使用 **fteCreateTransfer** 指令手動建立一個傳送定義檔，然後編輯該檔案的 <queue> 元素，以包含屬性 `groupId="${GROUPID}"`。然後，使用 **fteCreateMonitor** 指令提交傳送定義檔。

範例

在此範例中: 來源代理程式 (也是監視代理程式) 稱為 AGENT_MON; 目的地代理程式稱為 AGENT_DEST; 目的地檔案名稱稱為 /out/files/\${WMQFTEFileName}。此範例要求訊息已設定 MQMD 訊息描述子 WMQFTEFileName。所監視的佇列是 LIVE_QUEUE。

1. 執行下列指令來建立傳送定義檔:

```
fteCreateTransfer -sa AGENT_MON -da AGENT_DEST -df "/out/files/${WMQFTEFileName}"  
-de error -gt /tmp/TransferDefinition1.xml -sqgi -sq LIVE_QUEUE
```

即會產生傳送定義檔 /tmp/TransferDefinition1.xml。

2. 編輯 <queue> 元素以包含屬性 groupId="\${GROUPID}"。將此行:

```
<queue useGroups="true">LIVE_QUEUE</queue>
```

至

```
<queue useGroups="true" groupId="${GROUPID}">LIVE_QUEUE</queue>
```

這是必要屬性, 如此在傳送時會從佇列中讀取觸發傳送的訊息或訊息群組, 而不是佇列的第一個訊息或訊息群組。

3. 執行下列指令來建立監視器:

```
fteCreateMonitor -ma AGENT_MON -mq LIVE_QUEUE -mn QueueMon1 -mt /tmp/TransferDefinition1.xml  
-tr completeGroups -dv WMQFTEFileName=UNKNOWN
```

此監視器每 60 秒會輪詢佇列, 以查看是否有新訊息或新訊息群組抵達佇列。

目的地佇列是叢集佇列或叢集佇列的別名時應該如何處理

在使用 Managed File Transfer 將檔案傳入佇列時, 如果您使用的目的地是叢集佇列或叢集佇列的別名, 則會出現原因碼 2085 或 2082。從 IBM WebSphere MQ 7.5.0 Fix Pack 4 開始, 如果您將 enableClusterQueueInputOutput 內容設為 true, 便可解決此問題。

發生此問題的原因

當 -dq 上沒有明確的佇列管理程式名稱時, 目的地代理程式的佇列管理程式名稱會附加至 -dq 參數的佇列名稱。因為在連接至不具該本端叢集佇列的叢集 MQ queueManager 時, 無法在 MQOPEN 呼叫上指定 queueManager 物件, 所以會出現原因碼 2085 或 2082。

避免此問題

1. 在佇列管理程式上建立叢集佇列。
2. 設定指向叢集佇列的遠端佇列定義。

範例

此範例使用遠端佇列定義。

配置:

- 來源代理程式: SAGENT
- 來源代理程式佇列管理程式: SQM
- 目的地代理程式: DAGENT
- 目的地代理程式佇列管理程式: DQM
- 傳送的目的地佇列是佇列管理程式 SQM 上的 CQ6

若要將 DQM 上的「遠端佇列定義」Q6_SQM 定義至 SQM 中的叢集佇列 CQ6（假設叢集佇列 CQ6 已在 SQM 中定義），請在 DQM 佇列管理程式上發出 MQSC 指令：

```
define qremote(Q6_SQM) rname(CQ6) rqnname(SQM) xmitq(SQM)
```

註：rname 會指向叢集佇列。

現在您可以傳送至佇列。例如：

```
fteCreateTransfer -sa SAGENT -sm SQM -da DAGENT -dm DQM -dq Q6_SQM /tmp/single_record.txt
```

訊息若是建置在 SYSTEM.MANAGED.DURABLE 佇列上或填入檔案系統時應該如何處理

如果 IBM MQ Explorer 外掛程式在協調佇列管理程式上使用可延續訂閱，則訊息可以在 SYSTEM.MANAGED.DURABLE 佇列。如果您有流量高的 Managed File Transfer 網路，而且（或者）又不常使用「IBM MQ Explorer」外掛程式，則此訊息資料可能會填入本端檔案系統中。

若要移除建置在 SYSTEM.MANAGED.DURABLE 佇列上的訊息，您可以執行下列其中一個動作：

- 啟動使用可延續訂閱的「IBM MQ Explorer」。IBM MQ Explorer 的 Managed File Transfer 外掛程式會耗用佇列中的訊息。
- 手動刪除佇列中的訊息。

V 9.0.2

您可以透過下列其中一種方式來避免在可延續佇列上累積訊息：

- 指定「IBM MQ Explorer」外掛程式對協調佇列管理程式使用不可延續訂閱。請在 IBM MQ Explorer 中執行下列步驟：
 1. 選取視窗 > 喜好設定 > **IBM MQ Explorer** > **Managed File Transfer**
 2. 從傳送日誌訂閱類型清單中，選擇 NON_DURABLE。
- 清除協調佇列管理程式中由 IBM MQ Explorer MFT 外掛程式所建立的可延續訂閱。

可延續訂閱的名稱會加上字首，以顯示該訂閱是由「IBM MQ Explorer」MFT 外掛程式、主機名稱和使用者的名稱所建立，例如 MQExplorer_MFT_Plugin_HOST_TJWatson。

相關參考

第 88 頁的『保留 MFT 日誌訊息』

Managed File Transfer 會將檔案傳送進度及日誌資訊，傳送至協調佇列管理程式。協調佇列管理程式會將此資訊發佈至 SYSTEM.FTE 主題的任何相符訂閱。如果沒有訂閱，則不保留此資訊。

在發佈之前檢查訊息

因為代理程式可連接到 IBM WebSphere MQ 6.0 佇列管理程式，所以代理程式不會使用 IBM WebSphere MQ 7.0 所引進的直接發佈方式，而是將一般訊息傳送至包含 MQRFH 標頭的協調佇列管理程式。MQRFH 標頭需要發佈訊息的內容。這些訊息會傳送至協調佇列管理程式上的 SYSTEM.FTE 佇列，且訊息通常會直接從該佇列發佈。如果發生錯誤狀況導致此發佈停止，您可以在嘗試發佈前先檢查該佇列上的訊息，以協助進行診斷。您可以透過完成下列步驟來執行此作業：

1. 停用協調佇列管理程式中的發佈/訂閱引擎。

您可以使用 IBM MQ Explorer 或使用 MQSC 指令來完成此步驟。請注意，這會暫時停止佇列管理程式上的所有發佈/訂閱活動，包括與 Managed File Transfer 無關的活動（如果您的協調佇列管理程式還有其他用途）。

IBM MQ Explorer:

- a. 在「瀏覽器」視圖中，用滑鼠右鍵按一下協調佇列管理程式，然後選取內容。

- b. 從內容窗格中，選取**發佈/訂閱**。
- c. 從**發佈/訂閱模式**清單中，選取**相容性**。

MQSC:

```
ALTER QMGR PSMODE (COMPAT)
```

2. 傳送其他訊息。

執行有發佈問題的 Managed File Transfer 動作。以代理程式登錄為例，每當代理程式啟動時，就會傳送訊息（您無須反覆刪除及建立代理程式以產生登錄訊息）。因為發佈/訂閱引擎已停用，所以不會執行任何發佈。

3. 瀏覽協調佇列管理程式上的 SYSTEM.FTE 佇列。

您應該使用 IBM MQ Explorer 來瀏覽協調佇列管理程式的 SYSTEM.FTE 佇列。

IBM MQ Explorer:

- a. 在「導覽器」視圖中，展開協調佇列管理程式，然後按一下**佇列**。在「內容」視圖中，於 SYSTEM.FTE 佇列上按一下滑鼠右鍵，然後選取**瀏覽訊息**。**訊息瀏覽器**視窗即會開啟，並顯示將會發佈的訊息。
- b. **使用者 ID** 直欄會顯示訊息描述子中包含的使用者 ID。發佈失敗的常見原因，是此使用者 ID 對 SYSTEM.FTE 主題不具有發佈權限。
- c. 用滑鼠右鍵按一下訊息並選取**內容**，可以找到其他關於每一個訊息的資訊（包括將會發佈的 XML）。

目前沒有可檢查訊息內容的 MQSC 指令。如果您沒有 IBM MQ Explorer，則必須使用其他程式來瀏覽佇列以及顯示所找到訊息的各個層面。您可以使用 **amqsbcg** 範例程式（如果已安裝），如下列主題所述：**瀏覽佇列**。UserIdentifier 一行會顯示使用者 ID。或者，您可以使用 **dmpmqmsg**；訊息的使用者 ID 位於如下行中：

```
A RTM MQ24
A USR JOHNDOE
A ACC 1A0FD4D8F2F4C3C8C9D5F1F9C6F7C1C3F3F00019F7AC300000000000000000000
```

範例中的第二行是該訊息的訊息描述子使用者 ID。

4. 重新啟用協調佇列管理程式的發佈/訂閱引擎。

您可以使用 IBM MQ Explorer 或使用 MQSC 指令來完成此步驟。在重新啟用協調佇列管理程式中的發佈/訂閱引擎後，即會立即處理 SYSTEM.FTE 佇列上的任何訊息。

IBM MQ Explorer:

- a. 在「導覽器」視圖中，用滑鼠右鍵按一下協調佇列管理程式，然後選取**內容**。
- b. 從內容窗格中，選取**發佈/訂閱**。
- c. 從**發佈/訂閱模式**清單中，選取**已啟用**。

MQSC:

```
ALTER QMGR PSMODE (ENABLED)
```

IBM i 傳送 IBM i 儲存檔時發生可能的錯誤

如果您使用 Managed File Transfer 來傳送相同的 IBM i 儲存檔數次，則傳送可能失敗。

Managed File Transfer 可能會產生下列其中一個或兩個錯誤：

- 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"

如果您對 MFT 代理程式發出數個並行要求來傳送相同的 IBM i 儲存檔，則可能發生這些錯誤。如果您想要同時傳送相同的儲存檔數次，您必須使用數個來源代理程式。請對每一次並行傳送使用不同的來源代理程式。

若要以單一來源代理程式傳送相同的儲存檔數次，則在提交每一個新的傳送要求之前，您必須等待前一個傳送要求完成。

V 9.0.1 配置可重新配送的 MFT 代理程式時可能發生的錯誤

配置 Redistributable Managed File Transfer Agent 時的錯誤訊息

無法載入 Windows 的原生程式庫

Windows 您必須在系統上安裝下列 Microsoft 程式庫，才能使用 Redistributable Managed File Transfer Agent:

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

可從 Microsoft 取得這些程式庫。請參閱[最新支援的 Visual C++ 下載項目](#)。

如果您在未安裝這些程式庫的情況下嘗試執行 MFT 指令，將報告下列錯誤:

- 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.

檢查是否已安裝 Microsoft 程式庫。如果未安裝這些程式庫，請先安裝它們，然後重新執行此指令。

不支援使用連結模式

Redistributable Managed File Transfer Agent 只能在用戶端模式下連接至 IBM MQ。如果您嘗試在連結模式下執行指令，將報告下列錯誤:

- 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

當您發出指令時，必須提供佇列管理程式主機、埠、名稱及通道名稱。

設定與訊息大小相關聯的 MQ 屬性及 MFT 內容的指引

您可以變更 IBM MQ 屬性及 Managed File Transfer 內容，以影響 Managed File Transfer 在讀取或寫入不同大小訊息時的行為。

如果從來源佇列讀取或寫入目的地佇列的訊息大小超過 1048576 個位元組 (1 MB)，您必須將 Managed File Transfer Agent 內容 **maxInputOutputMessageLength** 的值增加到大於或等於要讀取或寫入的訊息大小上限的值。

如果來源佇列上的訊息大於 1048576 個位元組，您必須設定來源代理程式的 **maxInputOutputMessageLength** 內容。如果目的地佇列上的訊息大於 1048576 個位元組，您必須設定目的地代理程式的 **maxInputOutputMessageLength** 內容。如需 **maxInputOutputMessageLength** 內容的相關資訊，請參閱 [進階代理程式內容](#)。

- 如果代理程式寫入或從中讀取的佇列位於代理程式佇列管理程式的本端位置，您必須變更 IBM MQ 佇列管理程式、佇列及通道 **MAXMSGL** 屬性。

請確定來源或目的地佇列的訊息大小上限值大於或等於 **maxInputOutputMessageLength** 代理程式內容的值。

請確定下列每個 IBM MQ 屬性的值（以位元組為單位）：

- 代理程式佇列管理程式的訊息大小上限
- `SYSTEM.FTE.STATE.agent_name` 佇列的訊息大小上限
- 用戶端通道的訊息大小上限（如果您的代理程式以用戶端模式連接至佇列管理程式）

大於或等於下列計算的結果：

若為「檔案轉為訊息」傳送（支援的檔案大小上限為 **100 MB**）：
maxInputOutputMessageLength 的值

若為「訊息轉為檔案」傳送：

$$\text{值} = 3 * (\text{maxInputOutputMessageLength}) + 1048576$$

（此計算衍生自三個檢查點可以儲存在狀態訊息中的事實，並且每一個檢查點可能必須緩衝達資料訊息量的大小上限。）

- 如果代理程式要寫入的目標佇列為遠端佇列，您必須變更 IBM MQ 佇列管理程式、佇列及通道 **MAXMSGL** 屬性。

請確定下列每個 IBM MQ 屬性的值大於或等於 **maxInputOutputMessageLength** 代理程式內容的值：

- 代理程式佇列管理程式上的遠端佇列管理程式傳輸佇列的訊息大小上限
- 代理程式佇列管理程式至遠端佇列管理程式之通道的訊息大小上限
- 遠端佇列管理程式上的目的地佇列的訊息大小上限
- 遠端佇列管理程式的訊息大小上限

請確定下列每個 IBM MQ 屬性的值（以位元組為單位）：

- 代理程式佇列管理程式的訊息大小上限
- `SYSTEM.FTE.STATE.agent_name` 佇列的訊息大小上限
- 用戶端通道的訊息大小上限（如果您的代理程式以用戶端模式連接至佇列管理程式）

大於或等於下列計算的結果：

若為「檔案轉為訊息」傳送（支援的檔案大小上限為 **100 MB**）：
maxInputOutputMessageLength 的值

若為「訊息轉為檔案」傳送：

$$\text{值} = 3 * (\text{maxInputOutputMessageLength}) + 1048576$$

（此計算衍生自三個檢查點可以儲存在狀態訊息中的事實，並且每一個檢查點可能必須緩衝達資料訊息量的大小上限。）

如果超出其中一個內容的值，代理程式即會停止，並在代理程式事件日誌中記載下列錯誤：

```
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
```

代理程式事件日誌的上述訊息中可能會包括下列 IBM MQ 原因碼：

- **rc=2010** 此原因碼對映至 `MQRC_DATA_LENGTH_ERROR`，表示已超出用戶端通道的訊息大小上限值。若要解決此問題，請確定代理程式佇列管理程式的用戶端通道訊息大小上限大於或等於下列計算的結果：

$$3 * (\text{maxInputOutputMessageLength}) + 1048576$$

- **rc=2030** 此原因碼對映至 `MQRC_MSG_TOO_BIG_FOR_Q`，表示已超出 `SYSTEM.FTE.STATE.agent_name` 佇列的訊息大小上限值。若要解決此問題，請確定 `SYSTEM.FTE.STATE.agent_name` 佇列的訊息大小上限大於或等於下列計算的結果：

$$3 * (\text{maxInputOutputMessageLength}) + 1048576$$

- **rc=2031** 此原因碼對映至 `MQRC_MSG_TOO_BIG_FOR_Q_MGR`，表示已超出代理程式佇列管理程式的訊息大小上限值。若要解決此問題，請確定代理程式佇列管理程式的訊息大小上限大於或等於下列計算的結果：

$$3 * (\text{maxInputOutputMessageLength}) + 1048576$$

如果您要傳送許多小型訊息

如果代理程式對佇列讀取或寫入的訊息平均大小小於 1310 個位元組，且代理程式所讀取或寫入的訊息數超過 10000 個，您必須增加佇列管理程式的「未確定的訊息數量上限」屬性值，或減少一個檢查點間隔內的資料量。

當代理程式對佇列讀取訊息或寫入訊息時，會將對應的 **GET** 或 **PUT** 歸類到交易中。一個交易中的 **GET** 或 **PUT** 數目，取決於處理一個檢查點間隔內所有資料時所需的數量。一個檢查點間隔內的資料量概數可透過下列計算從代理程式內容來判斷：

$$\text{Checkpoint interval data size (in bytes)} = \text{agentCheckpointInterval} * \text{agentFrameSize} * \text{agentWindowSize} * \text{agentChunkSize}.$$

預設檢查點資料大小是 $1 * 5 * 10 * 262144$ 個位元組 = 13107200 個位元組 (12.5MB)。佇列管理程式在一個交易中支援的未確定訊息數目上限，由 **MaxUncommittedMsgs** 佇列管理程式屬性控制。此屬性的預設值是 10000 個訊息。如果平均訊息大小約小於 1310 個位元組，則在寫入的訊息數超過 10000 個時，即會超出預設的未確定訊息數目上限。

如果超出 **MaxUncommittedMsgs** 上限，代理程式即會停止，並在代理程式事件日誌中記載下列錯誤：

```
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.
```

原因碼 2024 對映至：`MQRC_SYNCPOINT_LIMIT_REACHED`。

若要解決此問題，請執行下列其中一個動作：

- 對從中讀取或寫入至佇列的代理程式所連接的佇列管理程式，增加其 **MaxUncommittedMsgs** 佇列管理程式屬性的值。請參閱 [MaxUncommittedMsgs \(MQLONG\)](#)。
- 減少一個檢查點間隔中的資料量。若要這麼做，請降低下列一個以上代理程式內容的值：
 - `agentCheckpointInterval`
 - `agentFrameSize`
 - `agentWindowSize`
 - `agentChunkSize`

如需這些代理程式內容的相關資訊，請參閱 [進階代理程式內容](#)。

如果您要持續將訊息寫入佇列

如果您要傳送至佇列並持續將訊息寫入佇列，您可能必須增加佇列管理程式日誌檔空間的大小，才能記載一個檢查點間隔中的所有資料。

如果超出佇列管理程式日誌檔空間，代理程式即會停止，並在代理程式事件日誌中記載下列錯誤：

```
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
```

原因碼 '2102' 對映至：MQRC_RESOURCE_PROBLEM。

若要解決此問題，請增加目的地代理程式佇列管理程式日誌檔空間的大小。

將 UAC 及虛擬儲存庫與 MFT 搭配使用的指引

Windows Server 2008 R2 及其他類似作業系統中都有「使用者帳戶控制 (UAC)」。這是一項安全基礎架構，其特性之一是將儲存在中央 Program Files 目錄的使用者資料轉移至使用者位置，一般稱為虛擬儲存庫。

如果僅使用 Managed File Transfer 工具來管理資料結構，則 UAC 及虛擬儲存庫對於 Managed File Transfer 並沒有影響。不過，如果非 IBM MQ 管理者使用標準作業系統工具變更或重建目錄結構，則新結構有可能會轉移至虛擬儲存庫中。這有可能導致下列一個以上的狀況：

- 使用者（包括 IBM MQ 管理者）無法再於其預期位置中查看檔案。
- 代理程式可能無法啟動，並報告訊息 BFGCL0315，但未提供支援的原因碼。
- 在代理程式報告的位置上找不到日誌檔。
- 代理程式以 **-F** 參數啟動時可能會失敗，並報告訊息：

```
The current directory is invalid
```

若要更正這些狀況，請執行下列動作：

- 以 IBM MQ 管理者身分，使用 **fteDeleteAgent** 及 **fteCreateAgent** 指令重建代理程式結構。
- 以作業系統管理者身分，移除受影響使用者之虛擬儲存庫中的 IBM MQ 項目。例如，在 Windows 上，虛擬商店的位置如下：`%USERPROFILE%\AppData\Local\VirtualStore\`

相關參考

[第 389 頁的『fteDeleteAgent』](#)

fteDeleteAgent 指令會刪除 Managed File Transfer Agent 及其配置。如果代理程式為通訊協定橋接器代理程式，則使用者認證檔會留在檔案系統中。

[第 326 頁的『fteCreateAgent \(建立 MFT 代理程式\)』](#)

fteCreateAgent 指令會建立 Managed File Transfer Agent 及其相關聯的配置。

Windows 以 Windows 服務方式執行 MFT 代理程式或日誌程式的指引

您可以用 Windows 服務方式來執行 Managed File Transfer 代理程式、獨立式資料庫日誌程式及獨立式檔案日誌程式。如果您使用這些 Windows 服務時發生問題，可利用服務日誌檔及本主題的資訊來診斷問題。

如需配置代理程式、獨立式日誌程式或獨立式檔案日誌程式以 Windows 服務方式執行的相關資訊，請參閱 [第 129 頁的『以 Windows 服務方式啟動 MFT 代理程式』](#) 及 [第 415 頁的『fteModify 日誌程式 \(以 Windows 服務方式執行 MFT 日誌程式\)』](#)。

註：如果可轉散發的代理程式將以 Windows 服務形式來執行，則需要在系統環境中設定 **BFG_DATA** 環境變數，服務才能運作。

日誌檔的位置

當您使用 **fteCreateAgent**、**fteCreateCDAgent**、**fteCreateBridgeAgent**、**fteModifyAgent**、**fteCreateLogger** 或 **fteModifyLogger** 指令，以 Windows 服務方式執行代理程式或日誌程式時，可以使用 **-sl** 參數來選擇記載層次。此參數的可能值為 **error**、**info**、**warn** 及 **debug**。預設值是 **info**。

Windows 服務日誌檔的檔名為 **servicedate.log**，其中 **date** 是啟動服務的日期。代理程式的檔案會寫入 **MQ_DATA_PATH\mqft\logs\coordination_qmgr_name\agents\agent_name** 目錄中。此目錄是寫入 Managed File Transfer Agent 追蹤檔的相同目錄。日誌程式的檔案會寫入 **MQ_DATA_PATH\mqft\logs\coordination_qmgr_name\loggers\logger_name** 目錄中。

如果您以 Windows 服務方式啟動代理程式或獨立式日誌程式時發生問題，請嘗試使用 **-sl** 參數將記載層次設為 **debug**。其他資訊會寫入 **servicedate.log** 檔中。

註：當記載層次設為 **debug** 時，您用來執行 Windows 服務的使用者帳戶及密碼會以純文字格式顯示在日誌檔中。

日誌檔的數目

當您使用 **fteCreateAgent**、**fteCreateCDAgent**、**fteCreateBridgeAgent**、**fteModifyAgent**、**fteCreateLogger** 或 **fteModifyLogger** 指令，以 Windows 服務方式執行代理程式或獨立式日誌程式時，可以使用 **-sj** 參數來選擇日誌檔數目。在指令中指定下列文字以變更日誌檔數目：**-sj -Dcom.ibm.wmqfte.daemon.windows.windowsServiceLogFiles=number**，其中 **number** 是您要以正整數表示的日誌檔數目。如果您未指定日誌檔數目，則預設值是 5。

「以服務方式登入」權限

您用來執行服務的 Windows 帳戶必須具有 **Log on as a service** 權限。如果您嘗試以 **fteStartAgent**、**fteStartLogger** 指令或以 Windows **Sc.exe** 指令啟動服務，但您使用的使用者帳戶沒有此權限，則會開啟「服務」視窗。如果您想要啟動的服務是要執行代理程式，則此視窗包含下列訊息：

```
Unable to start Windows service mqmftAgentAGENT@QMGR.  
System error 1069: The service did not start due to a logon failure.
```

在此訊息中，**AGENT** 是代理程式名稱，**QMGR** 是代理程式佇列管理程式名稱。如果您嘗試以服務方式執行獨立式日誌程式，會產生類似的訊息，其參照日誌程式而非代理程式。

若要防止此錯誤，請將 **Log on as a service** 權限授與您用來執行服務的 Windows 帳戶。例如，在 Windows 7 上完成下列步驟：

1. 從開始功能表，按一下系統管理工具 > 本機安全性原則。
2. 在安全設定窗格中，展開本機原則，然後按一下使用者權限指派。
3. 在原則和安全設定窗格中，按兩下以服務方式登入。
4. 按一下新增使用者或群組，然後將您要執行服務的使用者新增至具有 **Log on as a service** 權限的使用者清單。您在執行 **fteCreateAgent**、**fteCreateCDAgent**、**fteCreateBridgeAgent**、**fteModifyAgent**、**fteCreateLogger** 或 **fteModifyLogger** 指令時提供此使用者名稱。

註：錯誤 **System error 1069: The service did not start due to a logon failure.** 也可能是由不正確的密碼所造成。

隱藏您的 Windows 帳戶密碼

當您配置代理程式或獨立式日誌程式以 Windows 服務方式執行時，請指定要使用的使用者名稱及密碼。在下列範例中，建立了代理程式 **AGENT1**，它具有代理程式佇列管理程式 **QMGR1**，並配置為以 Windows 服務方式執行：

```
fteCreateAgent -agentName AGENT1 -agentQMGr QMGR1 -s -su fteuser -sp ftepassword
```

在此範例中，Windows 服務是以使用者名稱 **fteuser** 執行，它的相關聯密碼是 **ftepassword**。當您執行 **fteCreateAgent** 指令或其中一個接受 **-s** 參數的其他指令時，您會以純文字指定 Windows 帳戶的密碼。如果您不希望顯示密碼，請執行下列步驟：

1. 執行指令 (**fteCreateAgent**、**fteCreateCDAgent**、**fteCreateBridgeAgent**、**fteModifyAgent**、**fteCreateLogger** 或 **fteModifyLogger**)，但不指定 **-sp** 參數。例如：

```
fteCreateAgent -agentName AGENT1 -agentQMGr QMGR1 -s -su fteuser
```

註：此指令會產生一則訊息來警告您，您必須先使用「Windows 服務」工具設定密碼，然後服務才能順利啟動。

2. 開啟 Windows「服務」視窗。
3. 在服務清單中，在代理程式或獨立式日誌程式服務上按一下滑鼠右鍵，然後選取內容。代理程式服務顯示名稱是 Managed File Transfer Agent AGENT @ QMGR，其中 AGENT 是代理程式名稱，而 QMGR 是代理程式佇列管理程式名稱。日誌程式服務顯示名稱是 Managed File Transfer Logger for property set coordination_qmgr_name，其中 coordination_qmgr_name 是您指定給獨立式日誌程式作為其內容集的協調佇列管理程式。如需內容集的相關資訊，請參閱第 447 頁的『fteStartLogger』及第 415 頁的『fteModify 日誌程式 (以 Windows 服務方式執行 MFT 日誌程式)』。
4. 在內容視窗中，選取登入標籤。
5. 請在密碼及確認密碼欄位中，輸入執行該服務的使用者帳戶的密碼。在您輸入時，密碼字元會隱藏起來。
6. 按一下確定。

已知問題

使用 JAVA_HOME 系統環境變數時發生問題 (僅適用於 IBM WebSphere MQ 7.5.0 Fix Pack 1 或更早版本中的 Managed File Transfer)。

請勿設定 JAVA_HOME 系統環境變數，否則可能無法啟動代理程式或日誌程式的「Windows 服務」。代理程式或日誌程式的「Windows 服務」必須與 IBM MQ Java 執行時期一起執行。

Windows 更新代理程式或日誌程式 JVM 選項的指引

如果您使用 **fteModifyAgent** 或 **fteModifyLogger** 指令的 **-sj** 參數，透過更新、新增或移除 Java 系統內容來修改代理程式或日誌程式的現有 Windows 服務定義，則會先刪除現有 Windows 服務，然後再於其位置建立新的服務，並使用新 Windows 服務的內容來更新代理程式或日誌程式內容檔。新 Windows 服務定義必須與代理程式或日誌程式內容檔中定義的已更新 Windows 服務內容一致。

從 IBM MQ 9.0.0 Fix Pack 4 開始，會在 APAR IT22423 下新增其他檢查，以驗證使用 **fteModifyAgent** 或 **fteModifyLogger** 指令的 **-sj** 參數對代理程式或日誌程式的 JVM 選項所做的任何更新，以確定已正確指定這些選項。如果發現內容無效或無法驗證這些內容，**fteModifyAgent** 或 **fteModifyLogger** 指令將會失敗，並且顯示適當的錯誤訊息。

如果 JVM 內容有效且現有「Windows 服務」的刪除成功，但在 **fteModifyAgent** 或 **fteModifyLogger** 指令建立新的「Windows 服務」時失敗，則該指令會嘗試從代理程式或日誌程式內容檔中移除定義取代 Windows 服務的內容。在此情況下，將會傳回錯誤訊息，說明代理程式或日誌程式無法修改、已刪除舊 Windows 服務，但無法建立新 Windows 服務，因此代理程式或日誌程式不會作為 Windows 服務執行。您隨後必須手動驗證 Windows 服務定義的狀態，是否與代理程式或日誌程式內容檔中定義的 Windows 服務內容一致，並採取適當的動作來更正任何不一致。

相關資訊

[fteModifyAgent](#)：以 Windows 服務方式執行 MFT 代理程式

[fteModifyLogger](#)：以 Windows 服務方式執行 MFT 日誌程式

配置 MFT 資源監視器以避免代理程式超載的指引

您可以配置 Managed File Transfer 資源監視器的內容和參數值，以減少代理程式的負載。減少代理程式的負載可以改善代理程式的效能。您有數種設定可以使用，但可能需要反覆試驗，從而找出您系統配置的最佳設定。

資源監視概觀

代理程式會在資源監視器輪詢目錄或佇列時完成下列階段：

- 尋找所有符合觸發型樣的檔案（例如目錄中所有的 *.txt 檔案）。或是尋找佇列上所有完成的訊息群組。
- 判斷哪些是新建或有所變更的檔案，或判斷佇列中哪些是新建的群組。
- 起始傳送符合前兩個階段之準則的檔案或群組。
- 新增至已傳送的檔案和群組清單，以避免檔案變更之前再次傳送。

對於目錄監視器，來源目錄中的檔案越多，觸發型樣越大，代理程式必須剖析並與已傳送檔案清單進行比較的檔案清單越大。

如果是佇列監視器，當佇列上的群組愈多時，代理程式要與已經傳送之群組清單進行比較的群組清單也就愈大。

請考量下列重要設定：

- 使用代理程式內容 **monitorMaxResourcesInPoll**，設定代理程式每次輪詢所能包含的檔案數或群組數上限。使用此參數可限制輪詢間隔中的傳送數量。亦即，代理程式在起始傳送該數量的檔案或群組前，所需進行的剖析會比較少。代理程式會在目錄監視器或佇列監視器下一次輪詢時，納入下一組檔案或群組。代理程式內容 **monitorMaxResourcesInPoll** 在 IBM WebSphere MQ File Transfer Edition 7.0.4 及更新版本中可用，對於舊版 IBM WebSphere MQ File Transfer Edition，它可作為 APAR IC78011 的臨時修正程式。
- 建立目錄監視器時，請確認您配置的傳送定義包含來源處置 **delete**。設定此處置表示要在檔案傳送完成時，將其從監視目錄移除，且代理程式也不將該檔案保留在內部清單中。
- 建立目錄監視器時，請在 **fteCreateMonitor** 指令中使用 **-rl** 參數，以限制代理程式遞迴時所必須經歷的目錄層數。使用此參數表示不掃描不必要的低階目錄。

建立資源監視器時的其他考量

資源監視器輪詢的程序會耗用代理程式資源。增加監視器的輪詢間隔可減少代理程式的負載。但輪詢間隔的設定必須適中，以免每次輪詢間隔產生過多的傳送。設定資源監視器的輪詢間隔時，請考慮下列事項：

- 當檔案放入目錄或群組放入佇列後，您需要多快起始傳送。
- 檔案放入目錄或群組放入佇列的速率。
- 代理程式的最大傳送速率。代理程式必須能夠處理監視器所產生的所有傳送。

如果資源監視器是由 **fteCreateMonitor** 指令加上指定 **-pi**（輪詢間隔）與 **-pu**（輪詢間隔單位）參數所建立，便會指定輪詢間隔。您可能需要加以實驗，才能決定您配置的最佳設定。

如果要改善執行資源監視器之高負載代理程式的穩定性，可以選擇降低代理程式內容 **maxSourceTransfers** 的值。如此一來，代理程式便會將資源監視器和傳送檔案的處理時間分開。代理程式內容 **maxSourceTransfers** 的值愈高，傳送檔案所耗用的處理時間愈多，而能夠用於資源監視器的時間便愈少。如果您降低代理程式內容 **maxSourceTransfers** 的值，代理程式在同一時間內就會執行較少的傳送，因而有足夠的處理時間輪詢其資源監視器。由於佇列傳送數可能會增加，因此當您降低此代理程式內容的值時，應考慮增加代理程式內容 **maxQueuedTransfers** 的值。

最佳化監視器之後若發現有傳送進入回復，請考慮增加代理程式逾時值。代理程式負載量大可能表示代理程式在與目的地代理程式協議何時開始傳送時，發生傳送逾時。此逾時會造成傳送進入回復，並導致傳送延後完成。代理程式內容 **maxTransferNegotiationTime** 可指定來源代理程式等待目的地代理程式回應的時間。當超過此時間時，傳送就會進入回復。此內容的預設值為 30000 毫秒（30 秒）。增加內容的值（例如增加到 300000 毫秒，即 5 分鐘）可以讓傳送繼續而不逾時，並避免進入回復。

相關概念

第 133 頁的『使用傳送定義檔』

您可以指定傳送定義檔，它可用來建立檔案傳送。此傳送定義檔是 XML 檔，它定義建立傳送需要的部分或全部資訊。

相關工作

第 141 頁的『監視 MFT 資源』

您可以監視 Managed File Transfer 資源；例如，佇列或目錄。滿足此資源的條件時，資源監視器會啟動作業，例如檔案傳送。您可以在 IBM MQ Explorer 的 Managed File Transfer 外掛程式中使用

fteCreateMonitor 指令或「監視器」視圖來建立資源監視器。

相關參考

第 348 頁的『fteCreateMonitor: 建立 MFT 資源監視器』

fteCreateMonitor 指令可從指令行建立並啟動新的資源監視器。您可以使用 Managed File Transfer 來監視資源（例如目錄的內容），以在符合觸發條件時，啟動指定的作業（例如檔案傳送）。

變數替代導致多個檔案傳送至單一檔名時應該如何處理

對於 Managed File Transfer，如果您正在監視目錄並將多個檔案從來源傳送至目的地位置，而且正在使用 `${FileName}` 變數替代，則必須測試變數替代結果。需要測試結果是因為使用變數替代可能會導致呼叫非預期的檔案傳送指令組合。

若要判斷是否發生此問題，請尋找似乎在傳送多個檔案但只有一個檔案抵達目的地的情況。您可能會在檔案傳送日誌中看到顯示有多個檔案嘗試傳送至相同目的地檔名，但無法傳送至相同檔名的錯誤。

發生此問題的原因

當 MFT 目錄監視器正在處理多個檔案時，對於監視器在所監視目錄中找到的每個檔案，皆會執行作業 XML。如果僅在 XML 作業檔目的地中指定 `${FileName}`，而沒有在來源中進行指定，則會針對每個檔案多次呼叫傳送，針對每個檔名組合則呼叫一次。

例如：

```
<source disposition="delete" recursive="false">
  <file>e:\temp</file>
</source>
<destination exist="overwrite" type="file">
  <file>s:\outdir\${FileName}</file>
</destination>
```

避免此問題

如果您正在來源或目的地中使用 `${FileName}` 變數替代，並且預期相同檔名的變異會抵達目的地，請務必在作業 XML 定義的來源及目的地中同時指定 `${FileName}`。

下列範例從 `e:\temp\<filename>` 擷取檔案並將其傳送至 `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>
```

相關概念

第 149 頁的『使用變數替代來自訂 MFT 資源監視器作業』

滿足作用中資源監視器的觸發條件時，會呼叫所定義的作業。除了每次以相同的目的地代理程式或相同的目的地檔案名稱呼叫傳送或指令作業之外，您還可以在執行時期修改作業定義。您可以將變數名稱插入作業定義 XML 中，來達成此目的。當監視器判斷已滿足觸發條件，且作業定義包含變數名稱時，它會以變數值替換變數名稱，然後呼叫該作業。

第 152 頁的『範例: 資源監視器定義的變數替代』

使用 XML 和 IBM MQ Explorer 來替代資源監視器定義的變數範例。

在 Oracle 資料庫上更新 MFT 資料庫綱目時收到錯誤時應該如何處理

使用 `ftelog_tables_oracle_702_703.sql` 檔案將資料庫綱目更新至最新層次時，您可能會收到下列錯誤訊息：`ERROR at line 1: ORA-02289: sequence does not exist`。發生此錯誤的原因是表格所使用的順序及觸發程式不在與表格相同的綱目中。

關於這項作業

若要修正此問題，您必須在執行 `ftelog_tables_oracle_702_703.sql` 之前先對其內容進行編輯。

程序

1. 找出 Managed File Transfer 資料庫日誌程式表格使用的順序及觸發程式所在的綱目。
 - 在 Db2 上，您可以使用 Control Center 來檢視表格及綱目。
 - 在 Oracle 上，您可以使用 Enterprise Manager 來檢視表格及綱目。
2. 在文字編輯器中開啟 `ftelog_tables_oracle_702_703.sql` 檔案。
3. 對出現的每個 `SELECT FTELOG.sequence_name.nextval` 文字，以現有順序所在綱目的名稱取代 `FTELOG` 文字。
4. 在每次出現文字 `CREATE OR REPLACE TRIGGER FTELOG.trigger_name` 之前，插入文字 `DROP TRIGGER schema_name.trigger_name`，其中 `schema_name` 是現有觸發程式所在的綱目名稱。
5. 使用已編輯的 `ftelog_tables_oracle_702_703.sql` 檔案更新資料庫表格。

MFT 日誌程式錯誤處理及拒絕

Managed File Transfer 日誌程式可識別兩種類型的錯誤：個別訊息錯誤及一般錯誤。

個別訊息錯誤可能是由於一或數個個別訊息發生問題所導致。識別為個別訊息錯誤的一些狀況範例如下：

- 訊息中遺漏作為資料必要項目的結果碼
- 傳送指定工作名稱長度為 3000 個字元，這對相關聯的資料庫直欄而言太大
- 收到傳送的進度訊息，但沒有已開始傳送的記錄（原因可能是傳送開始訊息遭誤送或延遲）
- 收到訊息，但不是 Managed File Transfer 日誌訊息

一般錯誤是指非個別訊息錯誤以外的所有錯誤。這些錯誤可能是因配置問題錯誤所導致。

發現個別訊息錯誤時，日誌程式會透過將訊息放入拒絕佇列來拒絕訊息。由於不會向輸出日誌中寫入任何內容，因此，請定期檢查或持續監視拒絕佇列，以偵測被拒絕的訊息。

如果連續拒絕太多訊息，沒有任何訊息成功寫入資料庫，則會將這種情況視為一般錯誤。例如，請考量一律使用 10 個字元碼作為工作名稱的網站，但意外將工作名稱直欄重新配置為兩個字元寬度。雖然資料太寬通常屬於個別訊息錯誤，但在此案例中是一般的配置問題，該問題被偵測為一般錯誤。您可以使用 `wmqfte.max.consecutive.reject` 內容，調整導致一般錯誤所需的連續個別訊息錯誤數。

如果偵測到一般錯誤，日誌程式會回復尚未交由佇列管理程式確定的任何訊息，然後定期進行重試。識別問題的訊息會寫入輸出日誌及主控台（如果已使用 `-F` 參數以前景模式啟動日誌程式）。

日誌程式的輸出日誌位置，取決於是獨立式還是 JEE 資料庫日誌程式。如果是獨立式資料庫日誌程式，會位在目錄 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name` 中。若為 JEE 資料庫日誌程式，則位於應用程式伺服器的標準輸出日誌。

拒絕佇列

導致個別訊息錯誤的訊息會被移至拒絕佇列。在每個被拒絕訊息上，會設定訊息內容以指示拒絕訊息的原因。該內容的完整名稱是 `usr.WMQFTE_ReasonForRejection`（雖然 `usr`）。在某些環境定義中省略（包括 JMS 和 IBM MQ Explorer）。

如果您是使用「IBM MQ Explorer」，則可以用滑鼠右鍵按一下佇列，並按一下 **瀏覽訊息**，來檢視拒絕佇列的內容。若要查看拒絕訊息的原因，請按兩下訊息以開啟其內容對話框，然後選取「**具名內容**」頁面。您將會看到名為 `WMQFTE_ReasonForRejection` 的內容。或者，您可以撰寫或配置監視工具，以自動取得此資訊。

有時，您可能想要重新處理拒絕佇列中的訊息。在此主題先前說明的範例中，在使用兩個字元工作名稱直欄的資料庫中，訊息可以在增加資料庫直欄寬度後得到成功處理。又如另一個範例，當傳送完成訊息因其相關的傳送開始遺漏而遭到拒絕後，傳送開始訊息可能會在稍後送達。之後，重新處理傳送完成訊息即會成功執行。

若要重新處理訊息，請將其從拒絕佇列移至輸入佇列。在一般安裝中，日誌程式會建立自己的受管理訂閱，輸入佇列由佇列管理程式定義並具有類似 SYSTEM.MANAGED.DURABLE.49998CFF20006204 的名稱。您可以透過查看訂閱 SYSTEM.FTE.DATABASELogger.AUTO 的內容中的目的地名稱來識別輸入佇列，或者使用以下 MQSC 指令：

```
DISPLAY SUB(SYSTEM.FTE.DATABASELogger.AUTO) DEST
```

在佇列間移動訊息的其中一種方式是使用 [MA01 SupportPac](#)，例如：

```
q -IFTE.REJECT -oSYSTEM.MANAGED.DURABLE.49998CFF20006204
```

拒絕佇列會包含因各種原因而遭拒絕的訊息，僅部分訊息已得到解決。在此情況下，您仍可以重新處理所有訊息；現在可以接受的訊息會得到處理，無法接受的訊息會被再次移至拒絕佇列。

日誌程式不會記載傳送日誌中的 Malformed 日誌訊息。這些訊息會被認定為不具重要性，因此會傳送至拒絕佇列。如需傳送日誌訊息的相關資訊，請參閱第 637 頁的『[檔案傳送日誌訊息格式](#)』。

已啟動 MFT 日誌程式，但資料庫中未記載任何傳送資訊時應該如何處理

Managed File Transfer 日誌程式所使用的資料庫表格，需要資料庫具有 8KB 以上的分頁大小。如果資料庫的分頁大小不夠大，則無法適當地建立表格，而且您會看到錯誤 SQLSTATE=42704。

如果您是使用 Java Platform, Enterprise Edition 資料庫日誌程式，可能會在 WebSphere Application Server 系統輸出日誌中看到下列訊息；如果您是使用獨立式資料庫日誌程式，則可能會在 output0.log 檔案中看到下列錯誤：

```
DB2 SQL ERROR: SQLCODE=-204, SQLSTATE=42704
SQLERRMC=FTELOG.TRANSFER_EVENT, DRIVER=3.40.152
```

SQLSTATE 的 42704 值指出日誌程式預期存在的表格（在此案例中，是指 FTELOG.TRANSFER_EVENT）並不存在。

若要修正此問題，請執行下列步驟：

1. 確認表格確實存在且完整。如需日誌程式使用的表格及其直欄的相關資訊，請參閱第 561 頁的『[MFT 資料庫日誌程式表格](#)』。
2. 如果表格不存在或不完整，請檢查資料庫的分頁大小。
3. 如果資料庫大小小於 8KB，請增加資料庫的分頁大小。
 - 如果資料庫位於測試系統上或其中不含資料，您可以捨棄表格，並重建其分頁大小大於 8KB 的資料庫。
 - 如需如何增加分頁大小的相關資訊，請參閱第 17 頁的『[增加 Windows、UNIX 或 Linux 上 Db2 資料庫的分頁大小](#)』或第 19 頁的『[將 Db2 on z/OS 上的資料庫表格移轉至 IBM MQ 8.0](#)』。

MFT 不從 AMS 中的金鑰儲存庫配置檔讀取金鑰儲存庫內容時應該如何處理

如果金鑰儲存庫配置檔未出現在預設位置，則必須透過 MQS_KEYSTORE_CONF 變數指定金鑰儲存庫配置檔位置，Java AMS 才能夠在用戶端模式下執行。如果未指定此位置，則 Managed File Transfer Agent 日誌將顯示錯誤訊息：「無法從金鑰儲存庫配置檔讀取金鑰儲存庫內容」。

金鑰儲存庫配置檔的預設位置為 `home_directory/.mqsc/keystore.conf`。如果金鑰儲存庫配置檔的位置不是預設位置，請完成下列步驟：

1. 在用戶端模式下啟動 FTE 代理程式。

2. 將 AMS 安全套用至 SYSTEM.FTE.DATA.<agent name> 佇列。如果金鑰儲存庫配置檔不在此位置中，則所有傳送都將失敗，而且沒有確認通知。
3. 針對 `fteStartAgent` 指令，將系統變數 `BFG_JVM_PROPERTIES` 設為 `BFG_JVM_PROPERTIES=-DMQS_KEYSTORE_CONF=keystore_config` 檔案的路徑。
4. 對於 `fteStartAgent` 指令，將系統變數 `MQS_KEYSTORE_CONF` 設定為 `MQS_KEYSTORE_CONF=path to keystore_config file`。為了確保所有代理程式均可執行，不論它們是在何種模式下執行，都必須設定此項目。

註: 如果 Java AMS 是在連結模式下執行，而且金鑰儲存庫配置檔不在預設位置中，則佇列管理程式的錯誤日誌中將顯示錯誤 AMQ9062。

如果您的 MFT 代理程式 ABENDSS 由於原生記憶體耗盡而具有 `java.lang.OutOfMemoryError`，怎麼辦？

在處理許多受管理傳送要求 (例如檔案對檔案、訊息對檔案或檔案對訊息傳送) 時，代理程式會異常結束 (ABENDSS) 並報告 `java.lang.OutOfMemoryError`，此時您的 RAM 記憶體總數未完全使用。此異常狀況是由原生記憶體耗盡所造成。

診斷問題

發生此問題時，受影響代理程式 ABEND 會產生兩個檔案，以提供主要原因的詳細資料：

- ABEND 檔案。此檔案的名稱符合命名慣例 `ABEND.FTE.date_timestamp.identifier.log`。

Multi 在 Multiplatforms 上，檔案會寫入 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/ffdc` 目錄。

z/OS 在 z/OS 上，檔案會寫入 USS 位置 `$BFG_CONFIG/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/ffdc`

- Javacore 檔案。此檔案的名稱具有下列格式：

`javacore.datestamp.timestamp.pid.identifier.txt`

Multi 在 Multiplatforms 上，檔案會寫入 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name` 目錄。

z/OS 在 z/OS 上，檔案會寫入 USS 位置 `$BFG_CONFIG/mqft/logs/coordination_qmgr_name/agents/agent_name` 目錄。

ABEND 和 Javacore 配對包含類似下列範例的資訊：

範例: 配對 1

異常終止檔案

```
Filename:
C:\ProgramData\IBM\MQ\mqft\logs\COORDQM\agents\AGENT1\logs\ffdc\ABEND.FTE.20200109113518046.1764802189777
906538.log
Level: p900-005-180821
Time: 09/01/2020 11:35:18:046 GMT
Thread: 96 (TransferSender[414d51204d44424b525030372020202045fbd6532ebfaa02])
Class: com.ibm.wmqfte.thread.FTEThread
Instance: 55b455b4
Method: uncaughtException
Probe: ABEND_001
Cause: java.lang.OutOfMemoryError: native memory exhausted

java.lang.OutOfMemoryError: native memory exhausted
at com.ibm.mq.jmqi.local.internal.base.Native.MQPUT(Native Method)
at com.ibm.mq.jmqi.local.LocalMQ.MQPUT(LocalMQ.java)
at com.ibm.wmqfte.wmqiface.WMQQueueImpl.put(WMQQueueImpl.java)
at com.ibm.wmqfte.wmqiface.WMQQueueImpl.put(WMQQueueImpl.java)
at com.ibm.wmqfte.transfer.impl.TransferSenderRunnable.doTransfer(TransferSenderRunnable.java)
at com.ibm.wmqfte.transfer.impl.TransferSenderRunnable.run(TransferSenderRunnable.java)
```



```
at java.lang.Thread.run(Thread.java)
at com.ibm.wmqfte.thread.FTETHread.run(FTETHread.java)
```

Javacore 檔案

```
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\C0ORDQM\agents\AGENT1\javacore.20200109.113518.14148.0002.txt
```

範例: 配對二

ABEND 檔案

```
Filename:
C:\ProgramData\IBM\MQ\mqft\logs\C0ORDQM\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 檔案

```
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\C0ORDQM\agents\AGENT1\javacore.20200109.143700.2652.0003.txt
```

發生此問題的原因

由於代理程式執行所在系統上的原生資料堆記憶體耗盡而發生此問題。

如需 Java 資料堆記憶體與原生資料堆記憶體之間差異的相關資訊，請參閱 [第 623 頁的『MFT 代理程式如何使用 Java 資料堆及原生資料堆記憶體』](#)。

避免問題

您可以採取一些動作，以協助減少 MFT 代理程式因 `java.lang.OutOfMemoryError` 耗盡原生記憶體而停止的可能性：

1. 減少執行 MFT 代理程式之 JVM 的 Java 資料堆大小。

配置的 Java 資料堆大小越大，原生資料堆可用的記憶體越少。減少代理程式所使用的 Java 資料堆大小，可以釋放更多記憶體給原生資料堆。

依預設，代理程式的 Java 資料堆設為 512 MB。如果您已變更此項以使其成為更大的值，請考慮減少它，並使用類似正式作業的工作量進行測試。

- 在正常處理程序執行代理程式時，若要降低或變更 Java 資料堆，請執行下列動作：

設定 `BFG_JVM_PROPERTIES` 環境變數，以將 `options` 目錄傳遞至 JVM。例如，在 Windows 上，若要将資料堆大小上限設為 1024 MB，請先執行下列指令，然後再使用 **fteStartAgent** 指令：

```
set BFG_JVM_PROPERTIES="-Xmx1024M"
```

如需如何使用 `BFG_JVM_PROPERTIES` 環境變數來設定 Java 系統內容的相關資訊，請參閱 [MFT 的 Java 系統內容](#)。

- 以 Windows 服務方式執行代理程式時，如果要降低或變更 Java 資料堆，請執行下列動作：
若要將選項作為 Windows 服務傳遞至執行代理程式的 JVM，請使用 **fteModifyAgent** 指令上指定的 **-sj** 參數修改代理程式。
下列範例搭配使用 **fteModifyAgent** 指令與 **-sj** 參數，來設定執行 Windows 服務配置代理程式之 JVM 的 Java 資料堆大小上限：

```
fteModifyAgent.cmd -agentName AGENT1 -s -su user1 -sp passwd -sj -Xmx1024M
```

在代理程式重新啟動之後，您可以檢閱代理程式的 `output0.log` 檔案，以檢查是否已順利設定此項目。在開始顯示現行環境區段中，將報告值 1024 MB，如下所示：

```
The maximum amount of memory that the Java virtual machine will attempt to use is: '1024'MB
```

2. 限制原生記憶體使用

通常，如果代理程式使用 `BINDINGS` 傳輸連接至其代理程式佇列管理程式，則會看到原生資料堆耗盡所導致的 `java.lang.OutOfMemoryErrors`。當代理程式已配置成使用 `BINDINGS` 傳輸時，只要代理程式需要與佇列管理程式通訊，代理程式就會呼叫原生方法。

這表示原生記憶體用量會隨著代理程式工作量的增加而增加，因為與佇列管理程式的連線越來越多，且訊息通訊也越來越多。在此狀況下，減少工作量會有幫助。若要執行此動作，請將下列代理程式內容設為低於預設值 25 的值：

- **maxSourceTransfers**
- **maxDestinationTransfers**

這會減少可能發生的並行傳送數目，因此減少代理程式的並行工作量上限。

3. 配置代理程式，以在連接至其代理程式佇列管理程式時使用 `CLIENT` 傳輸。您可以透過設定下列代理程式內容來執行此動作：

- **agentQMgrHost**
- **agentQMgrPort**
- **agentQMgrChannel**

您可以在 [MFT agent.properties](#) 檔主題中找到這些內容的相關資訊。

這可確保代理程式與佇列管理程式之間的所有通訊都透過 TCP/IP 而非原生程式碼進行，這會減少代理程式所使用的原生記憶體數量。

重要：採取此動作也會降低效能。當代理程式需要與佇列管理程式互動時，使用與本端主機的 TCP/IP 連線 (而非原生程式碼)，配置沒有那麼有效率。

BFGSS0023E 錯誤及如何避免錯誤

如果您從安裝中解除安裝修正套件以回到舊版的產品，但在解除安裝時與此安裝相關聯的代理程式涉及受管理傳送，則該代理程式將無法啟動，而且會報告 BFGSS0023E 錯誤。您可以完成一些步驟來避免此錯誤，這些步驟應可在代理程式重新啟動時防止 BFGSS0023E 訊息出現。

對於代理程式目前涉及的每個進行中的受管理傳送，代理程式的 `SYSTEM.FTE.STATE.agent_name` 佇列中會有一則訊息。此訊息儲存受管理傳送的檢查點資訊，萬一受管理傳送進行回復即可使用。受管理傳送完成後，就會移除 `SYSTEM.FTE.STATE.agent_name` 佇列上的對應訊息。

每一個狀態訊息都包含一些內部標頭資訊，指出受管理傳送執行時代理程式所使用的 Managed File Transfer 元件版本。版本資訊會顯示特定的 Fix Pack 層次，因此，例如，如果 IBM MQ 8.0.0 Fix Pack 5 代理程式正在執行受管理傳送，則該受管理傳送的狀態訊息會包含 IBM MQ 8.0.0 Fix Pack 5 的參照。

如果從安裝中解除安裝修正套件，且與該安裝相關聯的代理程式具有相關聯的進行中傳送，則代理程式將無法啟動，而且會報告下列錯誤：

BFGSS0023E：代理程式配置為使用 IBM MQ 佇列，其中包含使用更新版本所建立的資料產品的。代理程式無法在此配置中執行，且將結束。

例如，如果 IBM MQ 8.0.0 Fix Pack 5 代理程式在停止然後降級至 IBM MQ 8.0.0 Fix Pack 4 層次時執行一些進行中傳送，則下次啟動代理程式時，它會檢查其 SYSTEM.FTE.STATE.agent_name 佇列並發現它們在使用 IBM MQ 8.0.0 Fix Pack 5 時已寫入。由於它現在是使用 IBM MQ 8.0.0 Fix Pack 4，因此代理程式會報告前一個段落所描述的 BFGSS0023E 錯誤，並自行關閉。

一般而言，如果要移除任一 Managed File Transfer 元件的修正套件，則完成下列步驟應可在代理程式重新啟動時防止 BFGSS0023E 訊息出現：

1. 確保其所有代理程式都已完成其受管理傳送。
2. 停止代理程式。
3. 移除修正套件。
4. 重新啟動代理程式。

相關工作

第 128 頁的『啟動 MFT 代理程式』

必須先啟動 Managed File Transfer 代理程式，才能將它用來進行檔案傳送。

相關參考

第 522 頁的『MFT 代理程式佇列設定』

fteCreateAgent 指令產生的 MQSC 指令 Script 可建立代理程式佇列，且參數設定為下列值。如果您未使用提供的 MQSC Script 來建立佇列，而是手動建立佇列，請確保將下列參數設定為給定的值。

相關資訊

[BFGSS0001 - BFGSS9999](#)

[在 UNIX 上將佇列管理程式回復至舊版](#)

[在 Windows 上將佇列管理程式回復至舊版](#)

受管理傳送失敗且發生 BFGIO0341E 錯誤時應該如何處理

如果受管理傳送正在將檔案傳送至外部處理程序所監視的位置，則該受管理傳送可能會失敗，並發生下列錯誤：BFGIO0341E：將暫存檔 *destination_filename.part* 重新命名為 *destination_filename* 失敗，因為暫存檔不存在。這是因為受管理傳送的目的地代理程式在寫入目的地檔案時使用暫存檔的方式。

目的地代理程式如何使用暫存檔

依預設，當進行受管理檔案傳送時，目的地代理程式會執行下列步驟：

- 建立一個稱為 *destination_filename.part* 的暫存檔。
- 鎖定暫存檔。
- 從來源代理程式接收到暫存檔時，將檔案資料寫入至暫存檔。
- 在接收並寫出所有檔案資料之後，解除鎖定暫存檔。
- 將暫存檔從 *destination_filename.part* 重新命名為 *destination_filename*。

如果受管理傳送進入回復，則目的地代理程式可以建立稱為 *destination_filename.partnumber* 的暫存檔。然後，目的地代理程式會將檔案資料寫入此檔案，而非稱為 *destination_filename.part* 的檔案。

如果暫存檔名 *destination_filename.partnumber* 已存在，則目的地代理程式會嘗試建立名稱為 *destination_filename.part(number + 1)* 的新暫存檔。如果該檔案已存在，則目的地代理程式會嘗試建立名稱為 *destination_filename.part(number + 2)* 的暫存檔，依此類推，直到它能夠順利

建立檔案為止。在代理程式嘗試建立暫存檔 `destination_filename.part1000` 但失敗的狀況下，它會直接寫入目的地檔案，而不使用暫存檔。

當受管理傳送完成時，目的地代理程式會刪除所有稱為 `destination_filename.partnumber` 的暫存檔，因為假設這些暫存檔是由代理程式在受管理傳送期間建立。

註: 如果代理程式內容 `doNotUseTempOutputFile` 設為值 `true`，則目的地代理程式不會使用暫存檔。相反地，它會直接寫入目的地檔案。如需 `doNotUseTempOutputFile` 內容的相關資訊，請參閱 [MFT agent.properties](#) 檔案。

發生此問題的原因

如果目的地代理程式嘗試重新命名暫存檔，但發現該檔案不再存在，則會產生 BFGIO0341E 錯誤。可能導致此問題的一般實務範例如下：

- 已在目標檔案系統上設定 暫置目錄。
- 外部處理程序配置為監視 暫置目錄，並將它找到的任何檔案移至新位置。
- 目的地代理程式會在 暫置目錄中建立並鎖定暫存檔 `destination_filename.part`。
- 目的地代理程式會將檔案資料寫入暫存檔。
- 將所有檔案資料寫入暫存檔之後，目的地代理程式會解除鎖定檔案。
- 外部處理程序會尋找暫存檔，並將它移至新位置。
- 目的地代理程式嘗試重新命名暫存檔，但發現它已不再存在。因此，傳送項目會標示為 **失敗**，並發生 BFGIO0341E 錯誤。

避免此問題

有兩種方法可防止發生 BFGIO0341E 錯誤：

- 由目的地代理程式寫入的暫存檔一律以 `.part` 或 `.partnumber` 字尾結尾。如果您可以將外部處理程序配置成忽略這些檔案，而不是移動這些檔案，則當目的地代理程式執行重新命名作業時，這些檔案仍會存在於目標目錄中。
- 或者，配置目的地代理程式，使其不使用暫存檔，並直接寫入目的地檔案。只有在所有檔案資料都已寫入目的地檔案時，該目的地檔案才會解除鎖定，此時外部處理程序可以挑選該目的地檔案。

若要將目的地代理程式配置為直接寫入目的地檔案，請設定代理程式內容

`doNotUseTempOutputFile=true`。如需此內容的相關資訊，請參閱 [MFT agent.properties](#) 檔案。

MFT 的回覆碼

Managed File Transfer 指令、Ant 作業及日誌訊息提供回覆碼，指出功能是否順利完成。

下表列出產品回覆碼及其意義：

回覆碼	簡短名稱	說明
0	成功	指令成功
1	指令不成功	指令未順利結束。
2	指令逾時	代理程式未於指定的逾時內回覆指令的狀態。若為受管理的呼叫及傳送指令，此逾時預設為無限制。例如，在 <code>fteCreateTransfer</code> 指令中指定 <code>-w</code> 參數。若為其他指令，此逾時預設為 5 秒。

表 24: 回覆碼 (繼續)

回覆碼	簡短名稱	說明
3	確認逾時	代理程式未於指定的逾時內確認收到指令。此逾時預設為 5 秒。
4	錯誤代理程式	指令傳送至錯誤代理程式。指令 XML 中指定的代理程式不是讀取指令佇列 (訊息放置所在) 的代理程式。
20	傳送局部成功	傳送局部順利完成, 部分檔案已傳送。
21	傳送已停止	其中一個使用者結束程式已停止傳送。
22	取消傳送逾時	代理程式收到取消傳送的要求, 但無法在 30 秒內完成取消。傳送未取消。
26	找不到取消 ID	代理程式收到取消傳送的要求, 但找不到傳送。這可能是因為傳送在代理程式處理取消要求之前即已完成。也可能是因為您提供不正確的傳送 ID 給 fteCancelTransfer 指令。已忽略取消要求。
27	取消進行中	代理程式收到取消傳送的要求, 但傳送已正在取消過程中。已忽略新的取消傳送要求。
40	失敗	傳送失敗, 未傳送任何指定的檔案。
41	已取消	傳送已取消。
42	觸發程式失敗	未執行傳送, 因為傳送為條件式, 且必要條件不符。
43	形態異常的 XML	XML 訊息形態異常。
44	已超出來源代理程式容量	來源代理程式沒有足夠的容量可執行傳送。
45	已超出目的地代理程式容量	目的地代理程式沒有足夠的容量可執行傳送。
46	已超出來源代理程式檔案數目上限	傳送的檔案數目已超出來源代理程式的限制。
47	已超出目的地代理程式檔案數目上限	傳送的檔案數目已超出目的地代理程式的限制。
48	日誌訊息屬性無效	日誌訊息形態異常。此錯誤是內部錯誤。如果收到此回覆碼, 請聯絡 IBM 支援中心尋求進一步協助。

表 24: 回覆碼 (繼續)

回覆碼	簡短名稱	說明
49	無法到達目的地	由於 IBM MQ 發生問題，來源代理程式無法將訊息傳送至目的地代理程式。例如，如果來源代理程式佇列管理程式未正確配置，則無法與目的地代理程式佇列管理程式進行通訊。
50	試用版違規	試用版代理程式嘗試與非試用版代理程式進行通訊。
51	不允許來源傳送	maxSourceTransfers 代理程式內容已設為 0。不允許此代理程式作為任何傳送的來源。
52	不允許目的地傳送	maxDestinationTransfers 代理程式內容已設為 0。不允許此代理程式作為任何傳送的目的地。
53	未獲授權	使用者未獲授權執行作業。請參閱伴隨的訊息，以取得進一步詳細資料。
54	權限層級不符	來源代理程式與目的地代理程式的 authorityChecking 代理程式內容值不符。
55	不支援觸發程式	嘗試在通訊協定橋接器代理程式上以觸發程式建立傳送。不支援此行為。
56	不支援目的地的檔案到訊息傳送	目的地代理程式不支援將檔案寫入目的地佇列
57	不支援檔案空間	目的地代理程式不支援檔案空間。
58	檔案空間遭到拒絕	目的地代理程式拒絕檔案空間傳送。
59	不支援目的地的訊息轉為檔案	目的地代理程式不支援訊息到檔案傳送。
64	不允許皆為佇列	傳送的來源及目的地皆為佇列。
65	一般資料佇列錯誤	在存取 Managed File Transfer Agent 資料佇列時發生錯誤。
66	資料佇列放置授權錯誤	在存取 Managed File Transfer Agent 資料佇列時發生錯誤。Advanced Message Security 未啟用。
67	資料佇列放置 AMS 錯誤	在存取 Managed File Transfer Agent 資料佇列時發生授權錯誤。Advanced Message Security 已啟用。

表 24: 回覆碼 (繼續)

回覆碼	簡短名稱	說明
V 9.0.1 V 9.0.1 69	傳送回復逾時	傳送的回復在指定的 transferRecoveryTimeout 值之後逾時。
70	代理程式已異常結束	應用程式發生無法復原的問題，正在強制終止。
75	佇列管理程式無法使用	應用程式無法繼續，因為應用程式的佇列管理程式無法使用。
78	啟動配置發生問題	應用程式無法繼續，因為啟動配置資料時發生問題。
85 個	資料庫伺服器的問題	應用程式無法繼續，因為資料庫有問題 (通常只有日誌程式才會傳回)
100	監視器替代無效	監視器作業 XML Script 內的變數替代格式形態異常。
101	監視器資源不正確	監視器資源定義的數目無效。
102	監視器觸發程式不正確	監視器觸發程式定義的數目無效。
103	監視器作業不正確	監視器作業定義的數目無效。
104	遺漏監視器	所要求的監視器不存在。
105	監視器已存在	所要求的監視器已存在。
106	監視器使用者結束程式錯誤	在資源監視器輪詢期間，監視器使用者結束程式產生錯誤。
107	監視器使用者結束程式已取消	監視器使用者結束程式已要求取消交易。
108	監視器作業失敗	由於處理作業時發生錯誤，監視器作業無法完成。
109	監視器資源失敗	監視器資源定義無法套用至指定的資源。
110	監視器作業變數替代失敗	已在監視器作業中指定變數，但在 meta 資料中找不到相符的名稱。因此，無法以值來取代變數。
111	監視器作業來源代理程式無效	監視器傳送作業的來源代理程式與資源監視器的代理程式不符。
112	監視器作業來源佇列管理程式無效	監視器傳送作業的來源代理程式佇列管理程式與資源監視器的代理程式佇列管理程式不符。
113	不支援監視器	嘗試在通訊協定橋接器代理程式上建立或刪除資源監視器。不支援此行為。
114	監視資源遭拒	監視資源掃描的目錄存取遭拒。

回覆碼	簡短名稱	說明
115	監視資源佇列在使用中	監視資源佇列已開啟，且對於具有共用存取的輸入不相容。
116	監視資源佇列不明	監視器的相關聯佇列管理程式上不存在監視資源佇列。
118	監視資源表示式無效	評估 XPath 表示式時發生錯誤。已評估 XPath 表示式，以存取訊息標頭中的使用者定義內容。訊息位於資源監視器所監視的佇列上。
119	監視作業來源代理程式佇列管理程式遺漏	監視作業定義中遺漏來源代理程式名稱或來源代理程式佇列管理程式名稱。
120	監視佇列未啟用	監視資源佇列未啟用。
121	存取監視器佇列時發生非預期的錯誤	存取監視器資源佇列時發生非預期的錯誤。
122	環境定義 ID 未啟用監視指令佇列	設定環境定義識別未啟用監視代理程式指令佇列。

下表列出產品中間回覆碼及其意義：

回覆碼	簡短名稱	說明
-2	ACK	要求已接收，但擱置完成。
-3	PROGRESS	要求適用於許多檔案，部分檔案仍在擱置完成。

註：

僅在產生要求的處理程序提供回覆佇列時，才顯示回覆碼。這些是中間回覆，且 Managed File Transfer 指令僅傳回最終回覆碼。

相關參考

第 288 頁的『傳送中檔案的回覆碼』

傳送內的個別檔案各有專屬的結果碼，其意義不同於指令的整體回覆碼。

傳送中檔案的回覆碼

傳送內的個別檔案各有專屬的結果碼，其意義不同於指令的整體回覆碼。

在 <action> 元素設為 "progress" 值的傳送日誌進度訊息中，每一個所報告檔案都有一個包含 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>
```



下表說明 resultCode 的可能值：

表 26: 傳送中的檔案結果碼

結果碼值	說明
0	成功。已順利傳送檔案。
1	失敗。無法傳送檔案。請參閱 <supplement> 元素，以取得錯誤的更多詳細資料。
2	警告。檔案已傳送，但報告警告訊息。例如，雖然來源處置設為刪除，但無法刪除來源檔案。請參閱 <supplement> 元素，以取得警告的更多詳細資料。

Connect:Direct 橋接器疑難排解

使用下列參照資訊及範例，有助於診斷從 Connect:Direct 橋接器傳回的錯誤。

- [第 289 頁的『追蹤 Connect:Direct 橋接器』](#)
- [第 289 頁的『Connect:Direct 橋接器的日誌資訊』](#)
- [第 290 頁的『解決有關 Connect:Direct 節點的權限問題』](#)
- [第 290 頁的『傳入或傳出 Connect:Direct 節點的文字傳送未正確轉換資料時要執行的作業』](#)
-  [第 291 頁的『透過 Connect:Direct 橋接器傳送至 PDS 或 PDS 成員失敗時要執行的作業』](#)
-  [第 291 頁的『指定有雙正斜線的 Connect:Direct 檔案路徑』](#)
- [第 292 頁的『增加 Connect:Direct 橋接器的並行傳送數目』](#)
- [第 293 頁的『對檔案傳送所呼叫的 Connect:Direct 程序進行除錯』](#)

追蹤 Connect:Direct 橋接器

您可以從屬於 Connect:Direct 橋接器的 Connect:Direct 節點擷取追蹤，以協助進行問題判斷。

關於這項作業

若要啟用追蹤，請完成下列步驟：

程序

1. 停止 Connect:Direct 橋接器代理程式。
2. 編輯 Connect:Direct 橋接器代理程式內容檔，以包含下列這一行：

```
cdTrace=true
```

3. 啟動 Connect:Direct 橋接器代理程式。

結果

追蹤資訊將會寫入 Connect:Direct 橋接器代理程式配置目錄中的 output0.log 檔案。

相關參考

[第 463 頁的『MFT agent.properties 檔案』](#)

每一個 Managed File Transfer Agent 都有自己的內容檔 agent.properties，其中必須包含代理程式用來連接至其佇列管理程式的資訊。agent.properties 檔案也可以包含變更代理程式行為的內容。

Connect:Direct 橋接器的日誌資訊

您可以使用 Connect:Direct 橋接器代理程式，在 MFT 代理程式與 Connect:Direct 節點之間傳送檔案。這些傳送中所涉及的 Connect:Direct 節點及程序的相關日誌資訊，將會顯示在「IBM MQ Explorer」外掛程式中，並儲存在日誌資料庫中。

Connect:Direct 橋接器代理程式必須為 IBM WebSphere MQ File Transfer Edition 7.0.4 或更新版本。傳送中所涉及的其他代理程式，可以是任何 Managed File Transfer 版本。但是，若要記載 Connect:Direct 節點及程序的相關資訊，傳送中所涉及的所有 MFT 代理程式，都必須為 IBM WebSphere MQ File Transfer Edition 7.0.4 或更新版本。若要讓此資訊顯示在「IBM MQ Explorer」外掛程式中，該外掛程式必須為 IBM WebSphere MQ File Transfer Edition 7.0.4 或更新版本。若要將此資訊儲存在日誌資料庫中，則資料庫日誌程式及資料庫綱目必須為 IBM WebSphere MQ File Transfer Edition 7.0.4 或更新版本。

檔案傳送中所涉及的 Connect:Direct 節點及 Connect:Direct 程序的相關日誌資訊，包含在發佈至協調佇列管理程式中 SYSTEM.FTE 主題的日誌訊息內。如需相關資訊，請參閱 [第 637 頁的『檔案傳送日誌訊息格式』](#)。

下列資訊將包含在已發佈的訊息內：

- Connect:Direct 橋接器節點名稱
- 主要節點 (PNODE) 名稱
- 次要節點 (SNODE) 名稱
- 處理程序名稱
- 程序 ID 號碼

Connect:Direct 橋接器節點，是與主要節點或次要節點相同的節點。

Connect:Direct 橋接器節點名稱的值，是 MFT Connect:Direct 橋接器代理程式識別該橋接器節點時所依據的名稱。主要及次要節點名稱，是用來參照 Connect:Direct 橋接器節點網路對映中節點的名稱。

相關參考

[第 655 頁的『Connect:Direct 橋接器傳送日誌訊息範例』](#)

當目的地代理程式或來源代理程式為 Connect:Direct 橋接器代理程式時，destinationAgent 或 sourceAgent 元素包含其他屬性。「已啟動」日誌訊息僅包含有關 Connect:Direct 傳送的部分資訊。「進行中」及「已完成」日誌訊息，包含有關 Connect:Direct 傳送的完整資訊。

解決有關 Connect:Direct 節點的權限問題

如果 Managed File Transfer 與 Connect:Direct 之間的傳送，因發生權限不足錯誤而失敗，請使用本主題中的資訊。

對於涉及 Connect:Direct 橋接器的傳送，連接至 Connect:Direct 節點的使用者 ID，是由與傳送要求相關聯的 IBM MQ 訊息描述子 (MQMD) 使用者 ID 所決定。您可以將特定 MQMD 使用者 ID 對映至特定 Connect:Direct 使用者 ID。如需相關資訊，請參閱 [第 120 頁的『對映 Connect:Direct 的認證』](#)。

您可能看到傳送因下列其中一個錯誤而失敗：

- ```
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.
```

如果您看到上述任一錯誤，請判斷與用於傳送要求的 MQMD 使用者 ID 相關聯的 Connect:Direct 使用者 ID。此 Connect:Direct 使用者 ID，必須具有 Connect:Direct 橋接器所需的執行 Connect:Direct 作業的權限。如需所需的職能權限清單，以及如何授與這些權限的指引，請參閱 [第 120 頁的『使用 ConnectDirectCredentials.xml 檔案對映 Connect:Direct 的認證』](#)。

傳入或傳出 Connect:Direct 節點的文字傳送未正確轉換資料時要執行的作業

當您以文字模式在 MFT 代理程式與 Connect:Direct 節點之間傳送檔案時，將會執行字碼頁及行尾字元轉換。該傳送會使用 Connect:Direct 橋接器節點的網路對映中的作業系統資訊，來判斷遠端節點的行尾字元。如果網路對映中的資訊不正確，可能不會正確執行行尾字元轉換。

確保 Connect:Direct 橋接器節點的網路對映，以及用作傳送目的地之任何 Connect:Direct 節點的網路對映，都包含正確的平台說明。

- 如果 Connect:Direct 橋接器節點是在 Windows 系統上，請務必針對網路對映中的每個遠端節點，從**作業系統**清單中選擇正確的值。
 - 如果遠端節點是在 Windows 系統上，請選取 Windows。
 - 如果遠端節點是在 UNIX 或 Linux 系統上，請選取 UNIX。
 - **z/OS** 如果遠端節點是在 z/OS 系統上，請選取 OS/390。

Connect:Direct 橋接器不支援傳送至其他作業系統上的遠端節點。

- 請確定將檔案傳送至或從中的每一個遠端節點，在 Connect:Direct 橋接器代理程式配置目錄中，指定 `ConnectDirectNodeProperties.xml` 檔案中遠端 Connect:Direct 節點的作業系統類型。如需相關資訊，請參閱第 120 頁的『[配置 ConnectDirectNodeProperties.xml 檔案以包含遠端 Connect:Direct 節點的相關資訊](#)』及第 494 頁的『[Connect:Direct 節點內容檔格式](#)』。

相關參考

第 542 頁的『[在 Connect:Direct 與 MFT 之間傳送文字檔](#)』

文字傳送涉及將檔案的字碼頁從一種字碼頁轉換為另一種字碼頁。文字檔傳送還會涉及在系統之間轉換 CRLF (換行, carriage return line feed) 字元。本主題彙總 Managed File Transfer Agent 與 Connect:Direct 節點之間傳送時的文字檔傳送行為。

z/OS 透過 Connect:Direct 橋接器傳送至 PDS 或 PDS 成員失敗時要執行的作業

如果傳送目的地是 z/OS 上的 Connect:Direct 節點，且是 PDS 或 PDS 成員，則當 `-de` 參數未指定改寫值時，傳送會失敗。

關於這項作業

如果您是使用 `fteCreateTransfer` 或 `fteCreateTemplate` 指令提交該傳送，請執行下列步驟：

程序

1. 變更已提交的指令，以包含 `-de overwrite`。
2. 重新提交該指令。

使用 *IBM MQ Explorer* 外掛程式

關於這項作業

如果您是使用「IBM MQ Explorer」外掛程式提交該傳送，請執行下列步驟：

程序

1. 在「**建立新的 Managed File Transfer**」精靈中，指定來源及目的地資訊。
2. 選取改寫目的地檔案系統上具有相同名稱的檔案。
3. 重新提交該指令。

z/OS 指定有雙正斜線的 Connect:Direct 檔案路徑

在檔案傳送過程中，如果您使用以雙正斜線 (//) 開頭的檔案路徑，來指定位於 Connect:Direct 節點上的檔案，則會將該檔案視為資料集。

Connect:Direct 節點上的來源及目的地，以 `cd_node_name:file_path` 格式指定。如果 `file_path` 是以雙正斜線 (//) 開頭，則會將該來源或目的地視為資料集。即使 Connect:Direct 節點不在 z/OS 上也一樣。如果在檔案路徑的開頭意外指定有雙正斜線 (//)，而該檔案又不是資料集，則會導致傳送失敗。

如果您不希望所指定的檔案被視為資料集，請確保不要指定以雙正斜線 (//) 開頭的 `file_path`。

相關概念

第 289 頁的『[Connect:Direct 橋接器疑難排解](#)』

使用下列參照資訊及範例，有助於診斷從 Connect:Direct 橋接器傳回的錯誤。

相關參考

第 534 頁的『[從 Connect:Direct 節點來回傳送資料集](#)』

您可以使用 Connect:Direct 橋接器，在 Managed File Transfer 代理程式與 IBM Sterling Connect:Direct 節點之間傳送資料集。您可以將資料集指定為傳送來源及/或傳送目的地。

增加 Connect:Direct 橋接器的並行傳送數目

若要增加 Connect:Direct 橋接器代理程式可以處理的並行傳送數目，您必須變更三個代理程式內容。您還必須增加 Connect:Direct 節點可接受的連線數目上限。

Connect:Direct 橋接器代理程式可以處理的並行傳送數目上限，取決於特定代理程式內容的值。對於 Connect:Direct 橋接器代理程式，**maxSourceTransfers** 及 **maxDestinationTransfers** 代理程式內容的預設值為五個傳送。此預設值低於其他代理程式類型的預設值（25 個傳送）。在 Connect:Direct 橋接器中，代理程式配置了預設值 **maxSourceTransfers** 及 **maxDestinationTransfers**，在任何一次最多可以處理 10 個傳送：代理程式為來源的 5 個傳送，以及代理程式為目的地的 5 個傳送。

這些預設值可確保 Connect:Direct 橋接器代理程式，不會超出與 Connect:Direct 節點的 API 連線數目上限。具有預設配置的 Connect:Direct 橋接器代理程式，所使用的與 Connect:Direct 節點的 API 連線數目上限為 10。UNIX 上 Connect:Direct 節點接受的連線數目上限由 **api.max.connects** Connect:Direct 參數控制。對於 Windows 上的 Connect:Direct 節點，對等參數為 **max.api.connects**。

如果 Connect:Direct 橋接器執行大量檔案傳送的速率不能滿足需要，您可以增加 Connect:Direct 橋接器代理程式可處理的並行傳送數目。請變更 Connect:Direct 橋接器代理程式的下列代理程式內容：

maxSourceTransfers

將此內容設定為大於 5，但小於或等於 25 的值。如果選擇大於 25 的值，除非對代理程式所用的 JVM 增加可用的記憶體數量，否則代理程式可能會用盡記憶體。

maxDestinationTransfers

將此內容設定為大於 5，但小於或等於 25 的值。如果選擇大於 25 的值，除非對代理程式所用的 JVM 增加可用的記憶體數量，否則代理程式可能會用盡記憶體。

ioThreadPoolSize

ioThreadPoolSize 的預設值為 10。對於 Connect:Direct 橋接器代理程式為來源代理程式的傳送，此內容會限制 Connect:Direct 節點 API 連線數目。這些傳送是從 Connect:Direct 至 Managed File Transfer。請使用下列指引來設定此內容的值：

- 如果 **maxSourceTransfers** 的值小於 **maxDestinationTransfers** 的值，請將 **ioThreadPoolSize** 設定為 **maxSourceTransfers** 值的兩倍或 10（以較大者為準）
- 如果 **maxSourceTransfers** 的值大於 **maxDestinationTransfers** 的值，請將 **ioThreadPoolSize** 設定為 **maxSourceTransfers** 與 **maxDestinationTransfers** 之和

除這些代理程式內容以外，您還必須變更屬於 Connect:Direct 橋接器的 Connect:Direct 節點的並行 API 連線數目上限。如果節點位於 UNIX 上，則控制此數目的 Connect:Direct 參數為 **api.max.connects**，如果節點位於 Windows 上，則為 **max.api.connects**。請對適當參數進行下列變更：

api.max.connects (如果 Connect:Direct 橋接器中的節點是在 UNIX 上)

將此參數設為大於 **maxSourceTransfers** 及 **maxDestinationTransfers** 總和的值。

api.max.connects 參數的預設值為 16。如需如何設定此參數的相關資訊，請參閱 Connect:Direct 文件。

max.api.connects (如果 Connect:Direct 橋接器中的節點是在 Windows 上)

將此參數設為大於 **maxSourceTransfers** 及 **maxDestinationTransfers** 總和的值。

max.api.connects 參數的預設值是 10。如需如何設定此參數的相關資訊，請參閱 Connect:Direct 文件。

相關工作

第 118 頁的『[配置 Connect:Direct 橋接器](#)』

配置 Connect:Direct 橋接器，以在 Managed File Transfer 網路與 Connect:Direct 網路之間傳送檔案。

Connect:Direct 橋接器的元件為 Connect:Direct 節點，以及專門用來與該節點通訊的 Managed File Transfer 代理程式。此代理程式稱為 Connect:Direct 橋接器代理程式。

相關參考

第 463 頁的『[MFT agent.properties 檔案](#)』

每一個 Managed File Transfer Agent 都有自己的內容檔 `agent.properties`，其中必須包含代理程式用來連接至其佇列管理程式的資訊。`agent.properties` 檔案也可以包含變更代理程式行為的內容。

對檔案傳送所呼叫的 Connect:Direct 程序進行除錯

您可以配置 Connect:Direct 橋接器代理程式，以將檔案傳送所呼叫的 Connect:Direct 程序的相關日誌資訊，寫入 Connect:Direct 橋接器代理程式配置目錄中的 `output0.log` 檔案。

關於這項作業

若要配置 Connect:Direct 程序的記載，請完成下列步驟：

程序

1. 停止 Connect:Direct 橋接器代理程式。
2. 編輯 `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` 目錄中的 `agent.properties` 檔案，以包含 `logCDProcess` 內容。
`logCDProcess` 內容可以具有下列其中一個值：
 - None - 不記載任何資訊。這是預設值。
 - Failures - 記載失敗的 Connect:Direct 程序相關資訊。
 - All - 記載所有 Connect:Direct 程序的相關資訊。
3. 啟動 Connect:Direct 橋接器代理程式。

結果

Connect:Direct 程序的相關資訊，將會記載至 Connect:Direct 橋接器代理程式的 `output0.log` 檔案。記載的資訊將包含：

- MFT 傳送 ID
- Connect:Direct 程序名稱
- Connect:Direct 程序號碼
- 產生的程序定義
- 程序範本的檔名（如果是使用者定義的 Connect:Direct 程序）

相關概念

第 289 頁的『[Connect:Direct 橋接器疑難排解](#)』

使用下列參照資訊及範例，有助於診斷從 Connect:Direct 橋接器傳回的錯誤。

相關參考

第 463 頁的『[MFT agent.properties 檔案](#)』

每一個 Managed File Transfer Agent 都有自己的內容檔 `agent.properties`，其中必須包含代理程式用來連接至其佇列管理程式的資訊。`agent.properties` 檔案也可以包含變更代理程式行為的內容。

MFT 安全參照

IBM MQ 中 MFT 的檔案系統許可權

當您安裝及配置 IBM MQ 的 Managed File Transfer 元件時，會建立具有下列許可權的 configuration、installations 及 logs 目錄。

UNIX 和 Linux



表 27: UNIX 及 Linux 上的目錄權限摘要

目錄	許可權
/var/mqm/mqft/config	<ul style="list-style-type: none"> 可由 mqm 群組寫入 可由所有人讀取 mqm 群組中的使用者具有這些目錄及檔案的寫入權
/var/mqm/mqft/installations	<ul style="list-style-type: none"> 可由 mqm 群組寫入 可由所有人讀取
/var/mqm/mqft/logs	可由所有人讀取及寫入

Windows



表 28: Windows 上的目錄權限摘要

目錄	許可權
MQ_DATA_PATH\mqft\config	下列使用者具有完整的讀取權及寫入權： <ul style="list-style-type: none"> • 管理者 • 系統帳戶 • mqm 群組 具有讀取權的其他使用者
MQ_DATA_PATH\mqft\installations	下列使用者具有完整的讀取權及寫入權： <ul style="list-style-type: none"> • 管理者 • 系統帳戶 • mqm 群組 具有讀取權的其他使用者
MQ_DATA_PATH\mqft\logs	下列使用者具有完整的讀取權及寫入權： <ul style="list-style-type: none"> • 管理者 • 系統帳戶 • mqm 群組 具有讀取權及寫入權的其他使用者

表 29: z/OS 上的目錄權限摘要

目錄	許可權
<code>DATA_PATH/mqft/config</code>	<ul style="list-style-type: none"> 可由 <code>mqm</code> 群組寫入，或是由環境變數 <code>BFG_GROUP_NAME</code> 中所識別的群組名稱寫入 可由所有人讀取 <p><code>mqm</code> 群組中的使用者，或是環境變數 <code>BFG_GROUP_NAME</code> 中的值，皆具有這些目錄及檔案的寫入權</p>
<code>DATA_PATH/mqft/installations</code>	<ul style="list-style-type: none"> 可由 <code>mqm</code> 群組寫入，或是由環境變數 <code>BFG_GROUP_NAME</code> 中所識別的群組名稱寫入 可由所有人讀取
<code>DATA_PATH/mqft/logs</code>	可由所有人讀取及寫入

註: `DATA_PATH` 衍生自環境變數 `BFG_DATA`。

管理 MFT 特定資源的權限

對於任何檔案傳送要求，Managed File Transfer 代理程式程序都需要對其本端檔案系統具有某種層次的存取權。此外，與代理程式處理程序相關聯的使用者 ID 以及與執行檔案傳送作業的使用者相關聯的使用者 ID，兩者都必須有使用特定 IBM MQ 物件的權限。

指令是由可能扮演作業角色的使用者發出，這個角色的使用者通常會啟動檔案傳送。或者，他們也可能扮演管理角色，可以另外控制何時建立、啟動、刪除或清除代理程式（亦即，何時移除所有代理程式系統佇列中的訊息）。當使用者發出指令時，包含指令要求的訊息會放置在代理程式的 `SYSTEM.FTE.COMMAND` 佇列上。代理程式程序會從 `SYSTEM.FTE.COMMAND` 佇列中擷取包含指令要求的訊息。代理程式程序也會使用四個其他系統佇列，如下所示：

- `SYSTEM.FTE.DATA.agent_name`
- `SYSTEM.FTE.EVENT.agent_name`
- `SYSTEM.FTE.REPLY.agent_name`
- `SYSTEM.FTE.STATE.agent_name`

因為發出指令的使用者以不同於代理程式處理程序的方式來使用前述佇列，所以，請針對與各佇列相關聯的使用者 ID 或使用者群組，指派不同的 IBM MQ 權限。如需相關資訊，請參閱第 296 頁的『限制 MFT 特定資源的群組權限』。

代理程式有其他佇列，可用來授權使用者執行特定動作。如需如何使用權限佇列的相關資訊，請參閱第 301 頁的『限制 MFT 代理程式動作的使用者權限』。代理程式不會在這些佇列上放置或取得訊息。不過，您必須確保佇列已獲指派正確的 IBM MQ 權限，包含用來執行代理程式處理程序的使用者 ID，以及與獲授權執行特定動作的使用者相關聯的使用者 ID。權限佇列如下所示：

- `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`

如果您要從早於 7.0.2 的 Managed File Transfer 版本移轉至 IBM WebSphere MQ 7.5 或更新版本，並保留現有的代理程式配置，則需要手動建立權限佇列。請使用下列 MQSC 指令來建立佇列：

```
DEFINE QLOCAL(authority_queue_name) DEFPRTY(0) DEFSOPT(SHARED) GET(ENABLED) MAXDEPTH(0) +
  MAXMSGL(0) MSGDLVSQ(PRIORITY) PUT(ENABLED) RETINTVL(999999999) SHARE NOTRIGGER +
  USAGE(NORMAL) REPLACE
```

代理程式處理程序也使用 SYSTEM.FTE 佇列，將訊息發佈至協調佇列管理程式上的 SYSTEM.FTE 主題。視代理程式處理程序扮演的是來源代理程式或目的地代理程式角色而定，代理程式處理程序可能需要讀取、寫入、更新及刪除檔案的權限。

您可以使用 IBM MQ Explorer 來建立及修改 IBM MQ 物件的權限記錄。在物件上按一下滑鼠右鍵，然後按一下 **物件權限 > 管理權限記錄**。您也可以使用 **setmqaut** 指令來建立權限記錄，其詳述於 [setmqaut \(授權或撤銷權限\)](#) 指令中。

相關參考

第 296 頁的『限制 MFT 特定資源的群組權限』

您可以為管理 Managed File Transfer 存取控制目的建立兩個安全群組：FTEUSER 及 FTEAGENT，而非針對可能涉及各種不同物件將權限授與個別使用者。IBM MQ 管理者負責建立及移入這些群組。管理者可以選擇延伸或修改這裡所說明的提議配置。

第 301 頁的『限制 MFT 代理程式動作的使用者權限』

除了使用群組管理對資源的存取外，您還可以啟用額外的一層安全來限制使用者可以採取的 Managed File Transfer 代理程式動作。將代理程式權限佇列的權限授與使用者，使使用者有權執行特定的代理程式動作。

第 304 頁的『MFT 日誌程式的權限』

執行日誌程式的作業系統使用者需要對日誌程式佇列及 SYSTEM.FTE 主題具有特定 IBM MQ 權限。

限制 MFT 特定資源的群組權限

您可以為管理 Managed File Transfer 存取控制目的建立兩個安全群組：FTEUSER 及 FTEAGENT，而非針對可能涉及各種不同物件將權限授與個別使用者。IBM MQ 管理者負責建立及移入這些群組。管理者可以選擇延伸或修改這裡所說明的提議配置。

連接至佇列管理程式的權限

由作業使用者、管理使用者及「IBM MQ 探險家」所執行的指令，需要能夠連接至指令佇列管理程式及協調佇列管理程式。針對建立、變更或刪除代理程式所執行的代理程式程序及指令，需要能夠連接至代理程式佇列管理程式。

- 授與 FTEUSER 群組指令佇列管理程式及協調佇列管理程式的連接權限。例如：

ULW 若為 UNIX、Linux 及 Windows 系統：

```
setmqaut -m command_queue_manager -t qmgr -g FTEUSER +connect
setmqaut -m coordination_queue_manager -t qmgr -g FTEUSER +connect
```

IBM i 若為 IBM i：

```
GRTRMQAUT OBJ('command_queue_manager') OBJTYPE(*MQM) USER(FTEUSER) AUT(*CONNECT)
GRTRMQAUT OBJ('coordination_queue_manager') OBJTYPE(*MQM) USER(FTEUSER) AUT(*CONNECT)
```

z/OS 若為 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)
```

- 對 FTEAGENT 群組授與連接及查詢代理程式佇列管理程式的權限。例如：

ULW 若為 UNIX、Linux 及 Windows 系統：

```
setmqaut -m agent_queue_manager -t qmgr -g FTEAGENT +connect +inq +setid
```


IBM i 若為 **IBM i**:

```
GRTMQMAUT OBJ('agent_queue_manager') OBJTYPE(*MQM) USER(FTEAGENT) AUT(*CONNECT)
```

z/OS 若為 **z/OS**:

```
RDEFINE MQCONN agent_queue_manager.BATCH UACC(NONE)  
PERMIT agent_queue_manager.BATCH CLASS(MQCONN) ID(FTEAGENT) ACCESS(READ)
```

如需各個指令直接對應連接哪些佇列管理程式的相關資訊，請參閱第 307 頁的『[哪些 MFT 指令和程序連接至哪個佇列管理程式](#)』。

將訊息放入 **COMMAND** 佇列（屬於代理程式）的權限

代理程式指令佇列必須可供有權要求代理程式執行動作的任何使用者使用。為能滿足此需求，

- 請只授予 FTEUSER 群組對 SYSTEM.FTE.COMMAND.agent_name 佇列的放置存取權。例如：

ULW 若為 **UNIX、Linux 及 Windows 系統**:

```
setmqaut -m QM1 -n SYSTEM.FTE.COMMAND.agent_name -t queue -g FTEUSER +put
```

IBM i 若為 **IBM i**:

```
GRTMQMAUT OBJ('SYSTEM.FTE.COMMAND.agent_name') OBJTYPE(*Q) USER(FTEUSER) AUT(*PUT)  
MQMNAME('QM1')
```

z/OS 若為 **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)
```

- 授與 FTEAGENT 群組對 SYSTEM.FTE.COMMAND.agent_name 佇列的 PUT、GET 及 SETID 存取權。例如：

ULW 若為 **UNIX、Linux 及 Windows 系統**:

```
setmqaut -m QM1 -n SYSTEM.FTE.COMMAND.agent_name -t queue -g FTEAGENT +browse +put +get  
+setid
```

IBM i 若為 **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 若為 **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)
```

代理程式需要將訊息放置到其他代理程式之指令佇列的存取權。如果有代理程式連接到遠端佇列管理程式，您可能需要授與其他權限，才能容許通道將訊息放置到此佇列。

將訊息放入 DATA、STATE、EVENT 及 REPLY 佇列（屬於代理程式）的權限

只有 Managed File Transfer 代理程式需要能夠使用這些系統佇列，因此需對群組 FTEAGENT 授與放置、取得與查詢存取權。這些系統佇列的名稱如下所示：

- DATA - SYSTEM.FTE.DATA.agent_name
- STATE - SYSTEM.FTE.STATE.agent_name
- EVENT - SYSTEM.FTE.EVENT.agent_name
- REPLY - SYSTEM.FTE.REPLY.agent_name

例如對於 SYSTEM.FTE.DATA.agent_name 佇列，可使用下列指令：

ULW 若為 UNIX、Linux 及 Windows 系統：

```
setmqaut -m QM1 -n SYSTEM.FTE.DATA.agent_name -t queue -g FTEAGENT +put +get +inq
```

IBM i 若為 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 若為 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)
```

代理程式需要將訊息放置到其他代理程式之資料和回覆佇列的存取權。如果有代理程式連接到遠端佇列管理模式，您可能需要授與其他權限，才能容許通道將訊息放置到這些佇列。

執行代理程式程序所使用的授權

執行代理程式程序所使用的授權，會影響代理程式能夠從檔案系統讀取及寫入的檔案，以及代理程式能夠存取的佇列及主題。權限的配置方式和系統有關。將執行代理程式程序所使用的使用者 ID 新增至 FTEAGENT 群組。如需將使用者 ID 新增至群組的相關資訊，請參閱設定安全，並導覽至您作業系統所適用的資訊。

執行指令及 IBM MQ Explorer 所使用的權限

管理指令 (例如 **fteStartAgent** 指令) 及 IBM MQ Explorer 的 Managed File Transfer 外掛程式需要能夠將訊息放置到 SYSTEM.FTE.COMMAND.agent_name 佇列，並從該佇列擷取已發佈的資訊。將授權執行這些指令或「IBM MQ Explorer 探險家」的使用者 ID，新增至 FTEUSER 群組。此發送端使用者 ID 會記錄在傳送日誌中。如需將使用者 ID 新增至群組的相關資訊，請參閱設定安全，並導覽至您作業系統所適用的資訊。

將訊息放入 SYSTEM.FTE 佇列及 SYSTEM.FTE 主題的權限

僅代理程式處理程序需要能夠將訊息放入 SYSTEM.FTE 佇列及 SYSTEM.FTE 主題。對 FTEAGENT 群組授與 SYSTEM.FTE 佇列的放置、取得及查詢權限，並且對 FTEAGENT 群組授與 SYSTEM.FTE 主題的發佈及訂閱權限。例如：

ULW 若為 UNIX、Linux 及 Windows 系統：

```
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 若為 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 若為 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)
```

如果有代理程式連接到遠端佇列管理程式，則可能還需要授與其他權限，才能容許通道將訊息放置到 SYSTEM.FTE 佇列。

若要將訊息發佈至 SYSTEM.FTE 主題，SYSTEM.FTE 主題的權限記錄必須容許訊息的訊息描述子結構 (MQMD) 中所包含的使用者 ID 進行發佈。[發佈日誌及狀態訊息的權限](#)中對此進行了說明。

若要讓使用者能夠發佈至 z/OS 上的 SYSTEM.FTE 主題，您必須授與通道起始程式使用者 ID 發佈至 SYSTEM.FTE 主題的存取權。如果 RESLEVEL 安全設定檔導致必須在通道起始程式連線中檢查兩個使用者 ID，則您還必須為訊息的訊息描述子結構 (MQMD) 中包含的使用者 ID 授與存取權。如需相關資訊，請參閱 [RESLEVEL 安全設定檔](#)

接收 SYSTEM.FTE 主題上發佈資訊的權限

傳送日誌訊息、進度訊息及狀態訊息均用於一般用途，因此請授與 FTEUSER 群組訂閱 SYSTEM.FTE 主題的權限。例如：

ULW 若為 UNIX、Linux 及 Windows 系統:

```
setmqaut -m QM1 -n SYSTEM.FTE -t topic -g FTEUSER +sub
```

IBM i 若為 IBM i:

```
GRTMQMAUT OBJ('SYSTEM.FTE') OBJTYPE(*TOPIC) USER(FTEUSER) AUT(*SUB) MQMNAME('QM1')
```

z/OS 若為 z/OS:

```
RDEFINE MXTOPIC QM1.SUBSCRIBE.SYSTEM.FTE UACC(NONE)
PERMIT QM1.SUBSCRIBE.SYSTEM.FTE CLASS(MXTOPIC) ID(FTEUSER) ACCESS(ALTER)
```

使用傳輸佇列連接至遠端佇列管理程式的權限

在多個佇列管理程式的拓撲中，對於連接至遠端佇列管理程式所使用的傳輸佇列，代理程式需要在這類佇列上具有 Put 權限。

針對檔案傳送建立暫時 REPLY 佇列的權限

檔案傳送要求會等到傳送完成，並依賴正在建立及移入的暫時回覆佇列。對 FTEUSER 群組授與暫時模型佇列定義的 DISPLAY、PUT、GET 及 BROWSE 權限。例如：

ULW 若為 UNIX、Linux 及 Windows 系統:

```
setmqaut -m QM1 -n SYSTEM.DEFAULT.MODEL.QUEUE -t queue -g FTEUSER +dsp +put +get +browse
```

IBM i 若為 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')
GRTMQMAUT OBJ('SYSTEM.DEFAULT.MODEL.QUEUE') OBJTYPE(*Q) USER(FTEUSER) AUT(*BROWSE)
MQMNAME('QM1')
```

z/OS 若為 z/OS:

```
RDEFINE MQQUEUE QM1.SYSTEM.DEFAULT.MODEL.QUEUE UACC(NONE)
PERMIT QM1.SYSTEM.DEFAULT.MODEL.QUEUE CLASS(MQQUEUE) ID(FTEUSER) ACCESS(UPDATE)
```

此佇列預設為 SYSTEM.DEFAULT.MODEL.QUEUE，但您可以透過設定 `command.properties` 檔案中 'modelQueueName' 及 'dynamicQueuePrefix' 內容的值，來配置該名稱。

在 z/OS 上，您也必須授與對 FTEUSER 的暫時佇列進行存取的權限。例如：

```
RDEFINE MQQUEUE QM1.WMQFTE.** UACC(NONE)
PERMIT QM1.WMQFTE.** CLASS(MQQUEUE) ID(FTEUSER) ACCESS(UPDATE)
```

依預設，z/OS 上每個暫時佇列的名稱都會以 WMQFTE 開頭。

下表彙總在所述安全保護方法中，FTEUSER 及 FTEAGENT 的存取控制配置：

物件	物件類型	FTEUSER	FTEAGENT
代理程式佇列管理程式	佇列管理程式		CONNECT、INQ 及 SETID。若要啟用使用者權限檢查，還需要 ALT_USER。
協調佇列管理程式	佇列管理程式		
指令佇列管理程式	佇列管理程式	連接	連接
SYSTEM.FTE	本端佇列		GET 及 PUT
SYSTEM.FTE.COMMAND.agent_name	本端佇列	PUT	BROWSE、GET、PUT 及 SETID
SYSTEM.FTE.DATA.agent_name	本端佇列		GET 及 PUT
SYSTEM.FTE.EVENT.agent_name	本端佇列		BROWSE、GET 及 PUT
SYSTEM.FTE.REPLY.agent_name	本端佇列		GET 及 PUT
SYSTEM.FTE.STATE.agent_name	本端佇列		BROWSE、GET、INQ 及 PUT
SYSTEM.FTE	本端主題	SUBSCRIBE	PUBLISH 及 SUBSCRIBE
SYSTEM.DEFAULT.MODEL.QUEUE (或者 Managed File Transfer 中定義的用於建立暫時 REPLY 佇列的模型佇列。)	模型佇列	BROWSE、DISPLAY、GET 及 PUT	BROWSE、DISPLAY、GET 及 PUT
要與遠端佇列管理程式進行通訊的傳輸佇列	本端佇列		PUT

透過 IBM MQ Explorer 管理傳送的權限

除了在此頁面已提及的狀況中將 MFT 權限授與使用者之外，還需要將進一步權限授與透過 IBM MQ Explorer 管理及執行所有 MFT 作業的 MFT 代理程式使用者。若要發出指令（如建立、取消、排定檔案傳送；建立、刪除資源監視器；以及建立傳送範本），IBM MQ Explorer 使用者必須擁有如下權限：

- 協調佇列管理程式：連接、查詢、顯示
- 指令佇列管理程式：連接、查詢、顯示
- SYSTEM.FTE 主題：發佈、訂閱

- SYSTEM.MQEXPLORER.REPLY.MODEL: display, inquire, get, browse, put
- SYSTEM.ADMIN.COMMAND.QUEUE: 查詢、放置、顯示
- SYSTEM.DEFAULT.MODEL.QUEUE: 取得、放置、查詢、顯示、瀏覽

如需各個指令直接對應連接哪些佇列管理程式的相關資訊，請參閱第 307 頁的『[哪些 MFT 指令和程序連接至哪個佇列管理程式](#)』。

相關參考

第 301 頁的『[限制 MFT 代理程式動作的使用者權限](#)』

除了使用群組管理對資源的存取外，您還可以啟用額外的一層安全來限制使用者可以採取的 Managed File Transfer 代理程式動作。將代理程式權限佇列的權限授與使用者，使使用者有權執行特定的代理程式動作。

第 304 頁的『[MFT 日誌程式的權限](#)』

執行日誌程式的作業系統使用者需要對日誌程式佇列及 SYSTEM.FTE 主題具有特定 IBM MQ 權限。


限制 MFT 代理程式動作的使用者權限

除了使用群組管理對資源的存取外，您還可以啟用額外的一層安全來限制使用者可以採取的 Managed File Transfer 代理程式動作。將代理程式權限佇列的權限授與使用者，使使用者有權執行特定的代理程式動作。

啟用使用者權限管理

若要開啟對代理程式動作的使用者權限檢查，請完成下列步驟：

1. 在 `agent.properties` 檔案中，將 `authorityChecking` 值設定為 `true`。如需相關資訊，請參閱第 463 頁的『[MFT agent.properties 檔案](#)』。
2. 確保執行代理程式的使用者，具有代理程式佇列管理程式的 IBM MQ 替代使用者 (ALT_USER) 權限。

 在 z/OS 平台上，執行代理程式的使用者對於可要求執行代理程式動作之權限的使用者 ID 必須具有 ALT_USER 權限。

傳送所涉及的兩個代理程式都必須已啟用相同的安全等級，也就是說，兩個代理程式內容檔案中所設定的 `authorityChecking` 值必須相同。如果兩個代理程式具有不同的 `authorityChecking` 內容值，則這兩個代理程式之間的傳送將失敗。

代理程式權限佇列

代理程式具有權限佇列，這些佇列用於管理有權執行特定代理程式動作的使用者。代理程式不會在這些佇列中放置或取得訊息。代理程式權限佇列如下所示：

- 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

當透過設定代理程式內容 `authorityChecking=true` 啟用使用者權限管理時，使用者在代理程式權限佇列上所具有的權限，指定授權該使用者採取的動作。

重要：   從 IBM MQ 9.0.0 Fix Pack 3 及 IBM MQ 9.0.4 開始，`inquire` 為所有代理程式權限佇列的必要權限。

下表彙總使用者或群組在代理程式權限佇列上需要除了 `查詢` 許可權之外才能執行特定動作的 IBM MQ 存取權。

表 31: 使用者或群組在代理程式權限佇列上執行特定動作所需的 IBM MQ 存取權層次。

使用者動作	Managed File Transfer 存取權	權限佇列	IBM MQ 存取權 (多平台)	RACF 存取層次 (僅限 z/OS)
在 fteStopAgent 指令中使用 -m 選項，以關閉代理程式。	管理	SYSTEM.FTE.AUTHADM1.agent_name	BROWSE	READ
啟動來源為此代理程式的檔案傳送	傳送來源	SYSTEM.FTE.AUTHTRN1.source_agent_name	BROWSE	READ
在此代理程式上執行受管理呼叫				
取消由同一使用者啟動的來源為此代理程式的檔案傳送				
啟動目的地為此代理程式的檔案傳送	傳送目的地	SYSTEM.FTE.AUTHTRN1.destination_agent_name	PUT	UPDATE
取消由同一使用者啟動目的地為此代理程式的檔案傳送				
建立資源監視器	監視者	SYSTEM.FTE.AUTHMON1.monitor_agent_name	BROWSE	READ
刪除由相同使用者建立的資源監視器				
刪除由任何使用者建立的資源監視器	監視作業	SYSTEM.FTE.AUTHOPS1.agent_name	設定	ALTER
建立排定的傳送	排程	SYSTEM.FTE.AUTHSCH1.source_agent_name	BROWSE	READ
刪除由相同使用者建立的排程傳送				
刪除由任何使用者或群組建立的排程傳送	排程作業	SYSTEM.FTE.AUTHOPS1.agent_name	PUT	UPDATE
取消由啟動傳送之相同使用者或群組或者由其他使用者或群組建立的傳送	傳送作業	SYSTEM.FTE.AUTHOPS1.source_agent_name SYSTEM.FTE.AUTHOPS1.destination_agent_name	BROWSE	READ

註: 若要授與使用者或群組許可權來設定啟動傳送的資源監視器或排定傳送，使用者需要 **監視器** 或 **排程** 權限，以及 **傳送** 來源和 **傳送** 目的地權限。

V9.0.0.3 **V9.0.4** 從 IBM MQ 9.0.0 Fix Pack 3 及 IBM MQ 9.0.4 開始，變更了當代理程式收到取消檔案傳送要求時執行的權限檢查順序，以便代理程式最先檢查發出取消要求的使用者是否為啟動傳送之相同使用者。如果取消傳送的使用者不是要求該傳送之相同使用者，代理程式會接著檢查執行取消作業的使用者是否具有 SYSTEM.FTE.AUTHOPS1.agent_name 佇列的 browse 權限。在要求檔案傳送的使用者與要求取消的使用者相同時，檢查順序的變更可避免代理程式及佇列管理程式錯誤日誌中出現非預期的錯誤。

使用者可以啟動一個代理程式，並讓它和另一個代理程式進行互動。兩個代理程式的互動方式，取決於使用者對於另一個代理程式權限佇列的存取權層次。

表 32: IBM MQ 存取權層次，啟動代理程式的使用者在另一個代理程式權限佇列上需要具有該權限，才能使檔案能夠在兩個代理程式間傳送。

代理程式動作	Managed File Transfer 存取權	權限佇列	IBM MQ 存取權 (多平台)	RACF 存取層次 (僅限 z/OS)
接受來自來源代理程式	代理程式來源	SYSTEM.FTE.AUTHAGT1.source_agent_name	BROWSE	READ
將傳送傳送至 destination_agent	代理程式目的地	SYSTEM.FTE.AUTHAGT1.destination_agent_name	PUT	UPDATE

配置使用者權限管理

若要授權使用者使其能夠對代理程式執行動作，請授與該使用者相關權限佇列上的適當權限。若要將權限授與使用者，請完成下列步驟：

1. 在代理程式佇列管理程式所在的系統上建立使用者，該使用者名稱與您要授權執行代理程式動作的使用者名稱相同。此使用者不必處理作用中狀態。
2. 授與使用者相關權限佇列上的適當權限。如果您是使用 Linux、UNIX 或 Windows，您可以使用 `setmqaut` 指令。
3. 重新整理佇列管理程式的安全配置。您可以使用 `REFRESH SECURITY MQSC` 指令。

範例

z/OS 在 z/OS 系統上，不使用 `setmqaut` 指令。若為 z/OS，請改用 RACF。如需相關資訊，請參閱設定 z/OS 上的安全。

IBM i 在 IBM i 系統上，不使用 `setmqaut` 指令。對於 IBM i，請參閱 IBM MQ 物件的存取權，其中說明如何完成 IBM MQ 物件的授權。IBM i 有三個相關的 CL 指令可用：**Grant MQ Object Authority (GRMQMAUT)**、**Revoke MQ Object Authority (RVKMQMAUT)** 及 **Refresh MQ Authority (RFRMQMAUT)**。

屬於 `requestor_group` 群組成員的某使用者想要在 AGENT1 上設定資源監視器，以將檔案從 AGENT1 傳送至 AGENT2，其中 AGENT1 由使用者 `user1`（屬於 `user1_group` 群組成員）執行，AGENT2 由使用者 `user2`（屬於 `user2_group` 群組成員）執行。AGENT1 連接至 QM1；AGENT2 連接至 QM2。這兩個代理程式均已啟用權限檢查。若要使此傳送變為可能，請採取下列步驟：

1. `requestor` 在 AGENT1 上必須具有監視權限。透過在執行 QM1 的系統上執行下列指令，來設定此權限：

```
setmqaut -m QM1 -t queue -n SYSTEM.FTE.AUTHMON1.AGENT1 -g requestor_group +browse
```

2. `requestor` 在 AGENT1 上必須具有傳送來源權限。透過在執行 QM1 的系統上執行下列指令，來設定此權限：

```
setmqaut -m QM1 -t queue -n SYSTEM.FTE.AUTHTRN1.AGENT1 -g requestor_group +browse
```

3. `requestor` 在 AGENT2 上必須具有傳送目的地權限。透過在執行 QM2 的系統上執行下列指令，來設定此權限：

```
setmqaut -m QM2 -t queue -n SYSTEM.FTE.AUTHTRN1.AGENT2 -g requestor_group +put
```

4. `user2` 在 AGENT1 上必須具有代理程式來源權限。透過在執行 QM1 的系統上執行下列指令，來設定此權限：

```
setmqaut -m QM1 -t queue -n SYSTEM.FTE.AUTHAGT1.AGENT1 -g user2_group +browse
```

5. `user1` 在 AGENT2 上必須具有代理程式目的地權限。透過在執行 QM2 的系統上執行下列指令，來設定此權限：

```
setmqaut -m QM2 -t queue -n SYSTEM.FTE.AUTHAGT1.AGENT2 -g user1_group +put
```

記載

如果已啟用使用者權限檢查，失敗的權限檢查會導致將未獲授權的日誌訊息發佈至協調佇列管理程式。如需相關資訊，請參閱第 695 頁的『安全的 MFT 訊息格式』。

有關使用者權限的訊息會被寫入代理程式事件日誌。透過設定代理程式內容檔中的 `logAuthorityChecks` 內容，您可以配置寫入代理程式事件日誌的資訊量。依預設，權限檢查記載的層次為 `None`。您也可以將

logAuthorityChecks 值設定為 Failures，指定僅報告失敗的權限檢查，或設定為 All，指定報告失敗及成功的權限檢查。

如需相關資訊，請參閱第 463 頁的『MFT agent.properties 檔案』。

相關參考

第 296 頁的『限制 MFT 特定資源的群組權限』

您可以為管理 Managed File Transfer 存取控制目的建立兩個安全群組：FTEUSER 及 FTEAGENT，而非針對可能涉及各種不同物件將權限授與個別使用者。IBM MQ 管理者負責建立及移入這些群組。管理者可以選擇延伸或修改這裡所說明的提議配置。

第 304 頁的『MFT 日誌程式的權限』

執行日誌程式的作業系統使用者需要對日誌程式佇列及 SYSTEM.FTE 主題具有特定 IBM MQ 權限。

第 449 頁的『fteStopAgent』

使用 **fteStopAgent** 指令，以受管制的方式停止 Managed File Transfer 代理程式，或在必要時使用 **-i** 參數立即停止代理程式。

MFT 日誌程式的權限

執行日誌程式的作業系統使用者需要對日誌程式佇列及 SYSTEM.FTE 主題具有特定 IBM MQ 權限。

執行日誌程式的作業系統使用者需要具有下列 IBM MQ 權限：

- 協調佇列管理程式的 CONNECT 及 INQUIRE。
- 對 SYSTEM.FTE 主題的 SUBSCRIBE 權限。
- 對 SYSTEM.FTE.LOG.RJCT.logger_name 佇列的 PUT 權限。
- 對 SYSTEM.FTE.LOG.CMD.logger_name 佇列的 GET 權限。

相關參考

第 296 頁的『限制 MFT 特定資源的群組權限』

您可以為管理 Managed File Transfer 存取控制目的建立兩個安全群組：FTEUSER 及 FTEAGENT，而非針對可能涉及各種不同物件將權限授與個別使用者。IBM MQ 管理者負責建立及移入這些群組。管理者可以選擇延伸或修改這裡所說明的提議配置。

第 301 頁的『限制 MFT 代理程式動作的使用者權限』

除了使用群組管理對資源的存取外，您還可以啟用額外的一層安全來限制使用者可以採取的 Managed File Transfer 代理程式動作。將代理程式權限佇列的權限授與使用者，使使用者有權執行特定的代理程式動作。

存取機密配置資訊的 MFT 權限

任何用來儲存機密性配置資訊的檔案（即任何利用 IBM MQ 配置樹狀結構參照的檔案）皆不得具有全系統的讀取、寫入或刪除（如果適用）權限。這些限制也適用於信任儲存庫檔案及金鑰儲存庫檔。

如果 Managed File Transfer 程序偵測到配置檔包含機密性資訊、屬於金鑰儲存庫檔或信任儲存庫檔案，且具有全系統的讀取、寫入或刪除權限的狀況，此程序會執行下列其中一個動作：

- 如果是在啟動時偵測到此狀況，則會無法啟動。
- 如果是在執行時期偵測到此狀況，則會產生警告訊息，並忽略配置檔的內容。這關係到通訊協定橋接器及 Connect:Direct 橋接器，這些橋接器會在配置於程序執行期間有所變更時予以重新載入。

在具有 UNIX 類型檔案系統的系統上

用來判斷檔案具有無法接受之全系統權限的準則為：

- 其他類別已被授與檔案的讀取權
- 其他類別已被授與檔案的寫入權
- 其他類別已被授與檔案所在目錄的寫入權

在 Windows 系統上

用來判斷檔案具有無法接受之全系統權限的準則為：

- 任何 Everyone、Guests 或 Users 群組都具有下列任何權限：
 - 檔案的讀取資料權限
 - 檔案的附加資料權限
 - 檔案的寫入資料權限
- 任何 Everyone、Guests 或 Users 群組都具有檔案所在資料夾的「建立檔案」權限，且同時具有下列任何權限：
 - 檔案所在資料夾的「刪除子資料夾及檔案」權限
 - 檔案的刪除權限

發佈 MFT 代理程式日誌及狀態訊息的權限

Managed File Transfer 代理程式會發出各種可在協調佇列管理程式上發佈的日誌、進度及狀態訊息。這些訊息的發佈會隨著 IBM MQ 安全模型而有不同，在某些情況下，您可能必須執行進一步的配置才能啟用發佈。

如需 IBM MQ 安全的相關資訊，請參閱開頭為[安全](#)的小節。

Managed File Transfer 代理程式會將要發佈的訊息，傳送至協調佇列管理程式上的 SYSTEM.FTE 佇列。每一個訊息在其訊息描述子 (MQMD) 中都附有使用者 ID。訊息會透過主題物件（亦稱為 SYSTEM.FTE）進行發佈。若要順利發佈指定的訊息，SYSTEM.FTE 主題的權限記錄必須允許由訊息的 MQMD 中所包含的使用者 ID 來發佈。

在 z/OS 上，通道起始程式使用者 ID 需要用來發佈至 SYSTEM.FTE 主題的存取權。如果 RESLEVEL 安全設定檔造成要檢查兩個使用者 ID 是否有通道起始程式連線，訊息的 MQMD 中的使用者 ID 也需要發佈至此主題的存取權。

訊息中最初包含的使用者 ID，取決於代理程式連接至其專屬佇列管理程式的方式。連接連結的代理程式所發出的訊息，包含代理程式據以執行的使用者 ID。連接用戶端的代理程式所發出的訊息則會包含內部 IBM MQ 使用者 ID。

您可以變更訊息中的使用者 ID。無論是連接用戶端還是連接連結的代理程式，您都可以使用 publicationMDUser 內容（在 agent.properties 檔中）來指定使用者 ID，而此 ID 將用於所有來自該代理程式的日誌及狀態訊息。代理程式必須獲得其專屬佇列管理程式的權限，才能使用此替代使用者 ID；只要對代理程式據以執行的使用者 ID 授與 setid 權限，即可指定此權限。

您也可以使用代理程式用來連接至其佇列管理程式之通道上的 MCAUSER 內容，對連接用戶端的代理程式所發出的所有訊息變更其所包含的使用者 ID。

您可以使用通道結束程式變更訊息中的使用者 ID，例如，在接收端通道上，將訊息導入協調佇列管理程式中。

視 IBM MQ 拓撲及原則而定，IBM MQ 管理者可透過多種方式使用此主題中的資訊，以確保狀態及日誌訊息能夠順利發佈。兩個範例如下：

- 決定網路中的代理程式所使用的所有使用者 ID。明確地為這些 ID 授與權限記錄。
- 建立一個以上用來發佈日誌及狀態訊息的一般使用者名稱。建立這些使用者名稱在協調佇列管理程式上的權限記錄。將每一個代理程式的 publicationMDUser 內容設為一般使用者名稱。在每一個代理程式佇列管理程式上，對代理程式據以執行的使用者 ID 授與 setid 權限，使之可以接受 publicationMDUser 內容。

MFT 存取檔案系統的權限

對於任何檔案傳送要求，Managed File Transfer 代理程式程序都需要對其本端檔案系統具有某種層次的存取權。

- 若要從來源檔案進行傳送，執行來源代理程式所使用的使用者 ID 必須具有來源檔案的讀取權。此外，根據來源處置屬性，您可能還需要授與來源代理程式刪除或寫入權限。
- 若要傳送至檔案或目錄，執行目的地代理程式所使用的使用者 ID 必須具有指定路徑的寫入權限。此外，根據目的地存在屬性，您可能還需要授與目的地代理程式更新權限。

- 除了您授與代理程式程序的檔案存取權外，您還可以使用沙盤推演來指定並強制執行受限檔案路徑區域。如需相關資訊，請參閱第 39 頁的『MFT 沙盤推演』。
- 如果您要來回傳送的檔案位於代理程式無法存取的位置（例如 **z/OS** VSAM 資料集，或是沙盤推演功能限制的位置），則可以使用 Managed File Transfer 使用者結束程式對可供代理程式存取的位置來回移動檔案。如需相關資訊，請參閱第 214 頁的『利用使用者結束程式自訂 MFT』。

commandPath MFT 內容

使用 commandPath 內容來指定 Managed File Transfer 可以從中執行指令的位置。在設定此內容時請特別小心，因為可以從遠端用戶端系統有效地呼叫其中一個指定的 commandPaths 中的任何指令，從而可以將指令傳送至代理程式。

您可以指定要在系統上執行的指令，在此系統中，代理程式從 Managed File Transfer 的受管理傳送及受管理呼叫功能執行。請參閱程式呼叫，以取得相關資訊。但是，指令必須在 commandPath 代理程式內容參照的路徑上。

如果指定的指令不完整，Managed File Transfer 會嘗試在指令路徑上尋找相符的指令。如果指令路徑上存在多個相符的指令，則會使用第一個相符項。

依預設，commandPath 內容為空白，因此代理程式無法呼叫任何指令。

指定 commandPath 代理程式內容，如下所示：

```
commandPath=command_directory_name
separator...command_directory_name
```

z/OS 或者僅對於 z/OS，請指定：

```
commandPath=command_directory_name_or_data_set_name_prefix
separator...command_directory_name_or_data_set_name_prefix
```

其中：

- *command_directory_name* 是可以執行指令的目錄路徑。
- **z/OS** *command_directory_name_or_data_set_name_prefix* 是可以執行指令的「z/OS UNIX 系統服務」目錄路徑，或是資料集名稱字首（以 // 開頭）。您可以選擇使用完整或不完整的資料集名稱字首（亦即，格式：// 'HLQ...' 或 //HLQ...）。以格式 // 'HLQ()...' 或 //HLQ()... 指定分割的資料集。僅使用資料集指定 JCL Script 指令。
- *separator* 是平台專用的分隔字元。

UNIX 例如，在 UNIX 系統上，如果要執行位於目錄 /home/user/cmds1 及 /home/user/cmds2 中的指令，請將 commandPath 代理程式內容設定如下：

```
commandPath=/home/user/cmds1:/home/user/cmds2
```

Windows 例如，在 Windows 系統上，如果要執行位於目錄 C:\File Transfer\commands 及 C:\File Transfer\agent commands 中的指令，請將 commandPath 代理程式內容設定如下：

```
commandPath=C:\\File Transfer\\commands;C:\\File Transfer\\agent commands
```

在 Windows 系統上，分隔字元反斜線 (\) 必須跳出並輸入為雙反斜線 (\\)。反斜線字元 (\) 也可以取代為正斜線 (/)。

z/OS 例如，在 z/OS 上，如果您要執行的指令：

- 在目錄 /home/user/cmds1 及 /home/user/cmds2 中
- 在以 // 'USER.COMD1' , //CMD2, 開頭的資料集中
- 是稱為 // 'USER.CMDS' 的完整 PDS 的成員

請將 commandPath 代理程式內容設定如下：

```
commandPath=/home/user/cmds1:/home/user/cmds2://'USER.CMD1'://CMD2://'USER.CMDS()'
```

重要：在設定此內容時請特別小心，因為可以從遠端用戶端系統呼叫其中一個指定的 commandPaths 中的任何指令，從而可以將指令傳送至代理程式。因此，指定 commandPath 時，依預設會配置沙盤推演，以在傳送時自動拒絕存取所有 commandPath 目錄（及其子目錄）：

- 如果代理程式是配置為使用代理程式沙盤推演，則當代理程式啟動時，commandPath 目錄會自動新增至被拒絕的目錄清單中。
- 如果代理程式是配置為使用一個以上的使用者沙盤推演，則當代理程式啟動時，會針對每一個使用者沙盤推演，將 commandPath 目錄作為 <exclude> 元素新增至 <read> 及 <write> 元素。
- 如果代理程式未配置為使用代理程式沙盤推演或使用者沙盤推演，當代理程式啟動時，會建立新的代理程式沙盤推演，將 commandPath 目錄指定為被拒絕的目錄。

V9.0.0.1 為了取得與下列版次的相容性，您可以置換此行為：

- IBM WebSphere MQ File Transfer Edition.
- IBM WebSphere MQ 7.5.0 Fix Pack 1 Managed File Transfer 元件（或更舊版本）。
- IBM WebSphere MQ 7.5.0 Fix Pack 2 Managed File Transfer 元件（或更舊版本），其所在的安裝環境中並未設定 enableFunctionalFixPack=7502 安裝內容。

您可以透過在 agent.properties 檔案中新增下列內容，來置換此行為：

```
addCommandPathToSandbox=false
```

V9.0.0.1 當 addCommandPathToSandbox 內容存在並設為 false 時，會發生下列行為：

- 如果代理程式是配置為使用代理程式沙盤推演，但沙盤推演沒有指定任何容許的目錄，則當代理程式啟動時，commandPath 目錄會自動新增至被拒絕的目錄清單中。
- 如果代理程式是配置為使用代理程式沙盤推演，且沙盤推演已指定一個以上容許的目錄，則當代理程式啟動時，commandPath 目錄不會新增至被拒絕的目錄清單中。
- 如果代理程式是配置為使用一個以上的使用者沙盤推演，則使用者沙盤推演不變，且對於每一個使用者沙盤推演，不會將 commandPath 目錄作為 <exclude> 元素新增至 <read> 及 <write> 元素。
- 如果代理程式未配置為使用代理程式沙盤推演或使用者沙盤推演，當代理程式啟動時，會建立新的代理程式沙盤推演，將 commandPath 目錄指定為被拒絕的目錄。

相關參考

第 463 頁的『MFT agent.properties 檔案』

每一個 Managed File Transfer Agent 都有自己的內容檔 agent.properties，其中必須包含代理程式用來連接至其佇列管理程式的資訊。agent.properties 檔案也可以包含變更代理程式行為的內容。

哪些 MFT 指令和程序連接至哪個佇列管理程式

Managed File Transfer 拓撲由一些不同元件組成。

這些元件如下：

- 一或多個代理程式及其相關聯的代理程式佇列管理程式
- 協調佇列管理程式
- 指令佇列管理程式
- 若干指令，用來管理拓撲及提交受管理傳送
- 選用的日誌程式，用於收集拓撲中代理程式所執行之受管理傳送的相關資訊
- IBM MQ Explorer Managed File Transfer 外掛程式，可用來執行一些管理作業以及檢視受管理傳送的相關資訊。

代理程式、日誌程式、指令及 IBM MQ Explorer Managed File Transfer 外掛程式在執行時會連接至一或多個佇列管理程式。

下列表格彙總了代理程式、日誌程式、指令及 IBM MQ Explorer Managed File Transfer 外掛程式在執行時會連接至哪個佇列管理程式。

如果表格中的某個指令或程序沒有 X 字元，則表示此指令在執行時不連接至任何佇列管理程式或程序。



指令名稱	代理程式佇列管理程式	指令佇列管理程式	協調佇列管理程式	日誌程式佇列管理程式
fteAnt				
fteCancelTransfer		X		
fteChangeDefaultConfigurationOptions				
fteCleanAgent	X			
fteCreateAgent	X			
fteCreateBridgeAgent	X			
fteCreateCDAgent	X			
 fteCreate 環境				
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 層次				

表 33: Managed File Transfer 指令連接至哪個佇列管理程式的摘要 (繼續)

指令名稱	代理程式佇列管理程式	指令佇列管理程式	協調佇列管理程式	日誌程式佇列管理程式
fteSetAgentTraceLevel				
fteSetLoggerTraceLevel				
 fteSetProductID				
fteSetupCommands				
fteSetupCoordination				
fteShowAgentDetails			X	
fteShowLoggerDetails				
fteStartAgent				
fteStartLogger				
fteStopAgent		X		
fteStopLogger		X		

表 34: Managed File Transfer 程序連接至哪個佇列管理程式的摘要

Processes	代理程式佇列管理程式	指令佇列管理程式	協調佇列管理程式	日誌程式佇列管理程式
Managed File Transfer 代理程式	X			
Managed File Transfer 外掛程式 (適用於「IBM MQ Explorer」)		X	X	
Managed File Transfer 日誌程式			X	X

該檔案包含連接至每種佇列管理程式類型所需的認證資訊，亦即可以在相關聯的內容檔中指定的代理程式、指令及協調佇列管理程式。例如，協調佇列管理程式有一個 `coordination.properties` 檔案。在此檔案中，您可以設定 `coordinationQMgrAuthenticationCredentialsFile` 內容來指向認證檔。

連接至協調佇列管理程式的指令使用在該檔案中所指定的認證資訊。如果佇列管理程式上已啟用安全，但未正確設定此內容，則 MFT 指令無法順利完成。如需相關資訊，請參閱 [MFT 和 IBM MQ 連線鑑別](#)。

相關資訊

[已安裝的 MFT 指令集](#)

哪些 MFT 指令連接至哪個佇列管理程式的詳細資料

關於哪些 Managed File Transfer 指令連接至哪個佇列管理程式的進一步資訊。

本主題會擴充 [第 307 頁的『哪些 MFT 指令和程序連接至哪個佇列管理程式』](#) 中的資訊，以及一些圖解。

連接至協調佇列管理程式的指令

下列指令會連接至協調佇列管理程式：

- [fteCreateTemplate](#)
- [fteDeleteAgent](#)
- [fteDeleteTemplates](#)
- [fteListAgents](#)
- [fteListMonitors](#)
- [fteListScheduledTransfers](#)
- [fteListTemplates](#)

- `fteShowAgentDetails`

Managed File Transfer 拓撲的協調佇列管理程式是對整個拓撲具有知識的中央中心。協調佇列管理程式會透過傳送端及接收端通道，連接至拓撲中的所有代理程式佇列管理程式。代理程式會定期將狀態資訊發佈至協調佇列管理程式，並將其傳送範本儲存在該處。

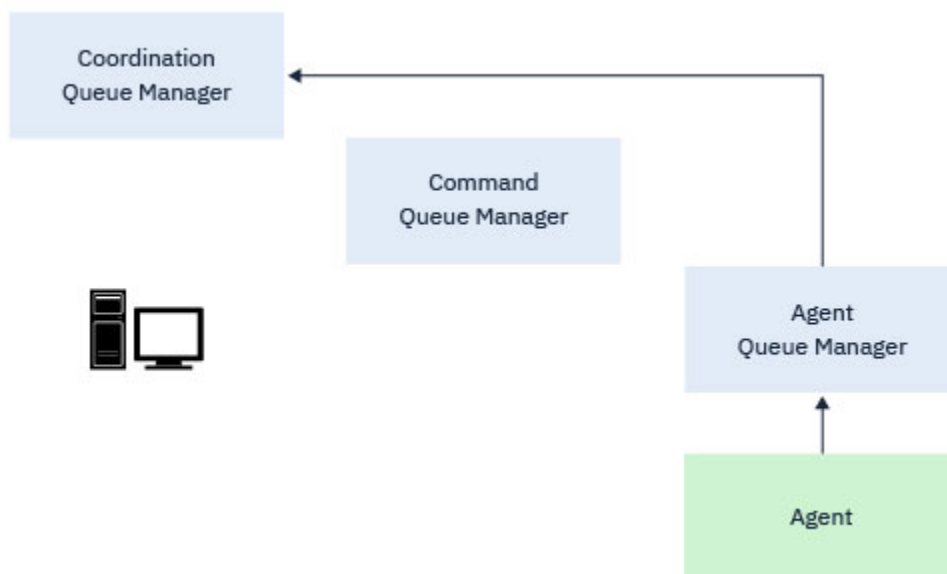


圖 9: 代理程式在協調佇列管理程式上發佈狀態資訊或儲存傳送範本

當執行上述任何連接至協調佇列管理程式的指令時，它們會直接連接至協調佇列管理程式，並執行下列任一動作：

- 建立或刪除傳送範本。
- 查詢代理程式、監視器或排程傳送的相關狀態資訊，並向使用者顯示該資訊。

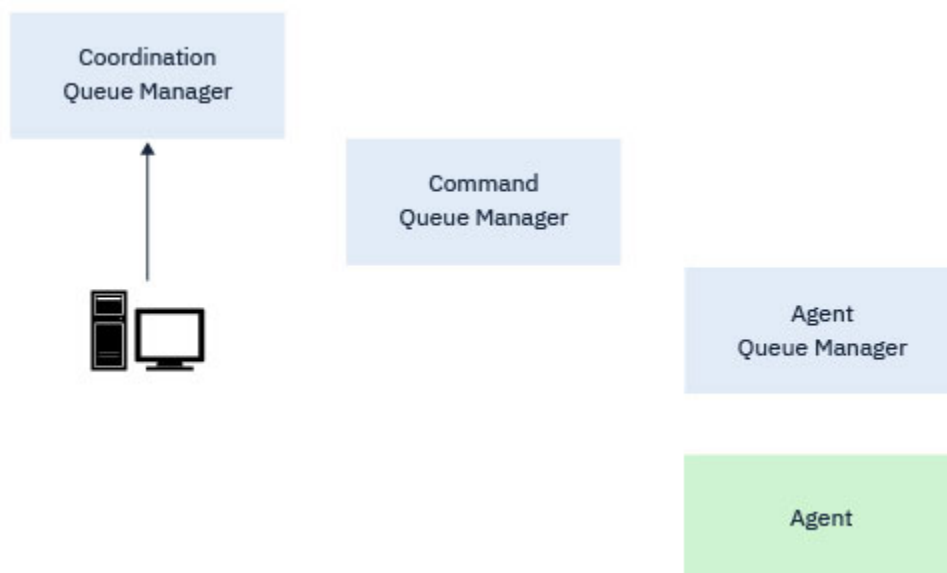


圖 10: 然後指令會直接連接至協調佇列管理程式，以取得適當的狀態資訊或使用傳送範本。

連接至指令佇列管理程式的指令

下列指令會連接至指令佇列管理程式：

- [fteCancelTransfer](#)
- [fteCreateMonitor](#)
- [fteCreateTransfer](#)
- [fteDeleteMonitor](#)
- [fteDeleteScheduledTransfer](#)
- [ftePingAgent](#)
- [fteStopAgent](#)

您可以將指令佇列管理程式視為 Managed File Transfer 拓撲中閘道的一種形式。它使用傳送端及接收端通道連接至代理程式佇列管理程式。

當執行任何先前連接至佇列管理程式的指令時，它們：

- 連接至指令佇列管理程式。
- 建立暫時回覆佇列。
- 將包含指令詳細資料的訊息傳送至適當的代理程式。

訊息會透過 IBM MQ 網路遞送至代理程式佇列管理程式，由代理程式挑選並處理。

代理程式處理指令之後，代理程式會將回覆傳送回指令佇列管理程式，其中指令會挑選回覆。

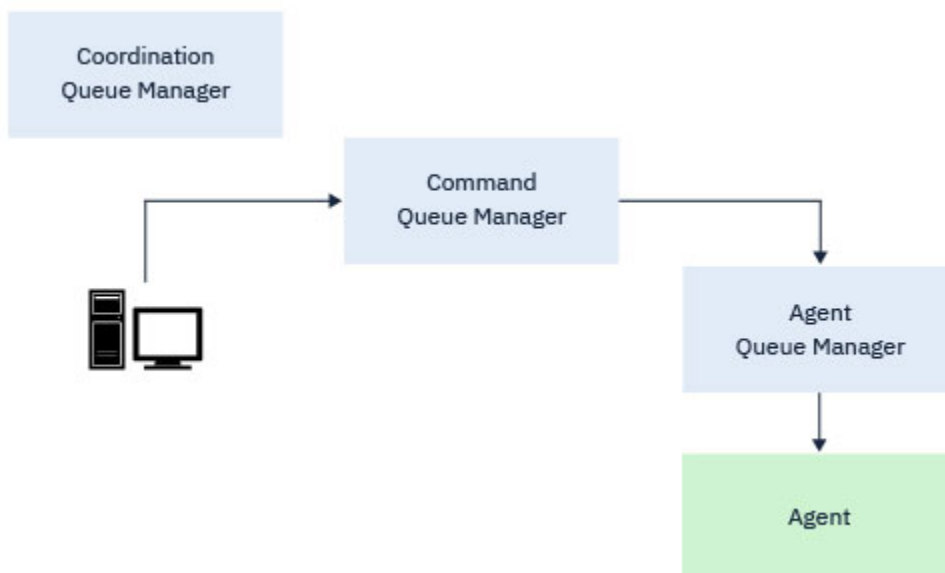


圖 11: 指令會連接至指令佇列管理程式。然後，包含指令的訊息會透過 IBM MQ 網路遞送至正確的代理程式佇列管理程式，代理程式會在其中挑選該訊息。

連接至代理程式佇列管理程式的指令

下列指令會連接至代理程式佇列管理程式：

- [fteCleanAgent](#)
- [fteCreateAgent](#)
- [fteCreateBridgeAgent](#)
- [fteCreateCDAgent](#)
- [fteDeleteAgent](#)

每個代理程式都有自己的代理程式佇列管理程式。代理程式使用在此佇列管理程式上管理的系統佇列來維護狀態資訊及接收工作要求。

單一佇列管理程式可以作為多個代理程式的代理程式佇列管理程式。代理程式佇列管理程式會使用傳送端及接收端通道連接至協調佇列管理程式、指令佇列管理程式及其他代理程式佇列管理程式。

fteCreateAgent、**fteCreateBridgeAgent** 及 **fteCreateCDAgent** 指令會採用代理程式佇列管理程式名稱作為引數。

執行這些指令時，它們會連接至指定的佇列管理程式，並將訊息傳送至協調佇列管理程式，指出代理程式已新增至 Managed File Transfer 拓撲。

同樣地，當 **fteDeleteAgent** 執行時，它會連接至代理程式佇列管理程式，並將訊息傳送至協調佇列管理程式，通知它現在已從 Managed File Transfer 拓撲中移除代理程式。

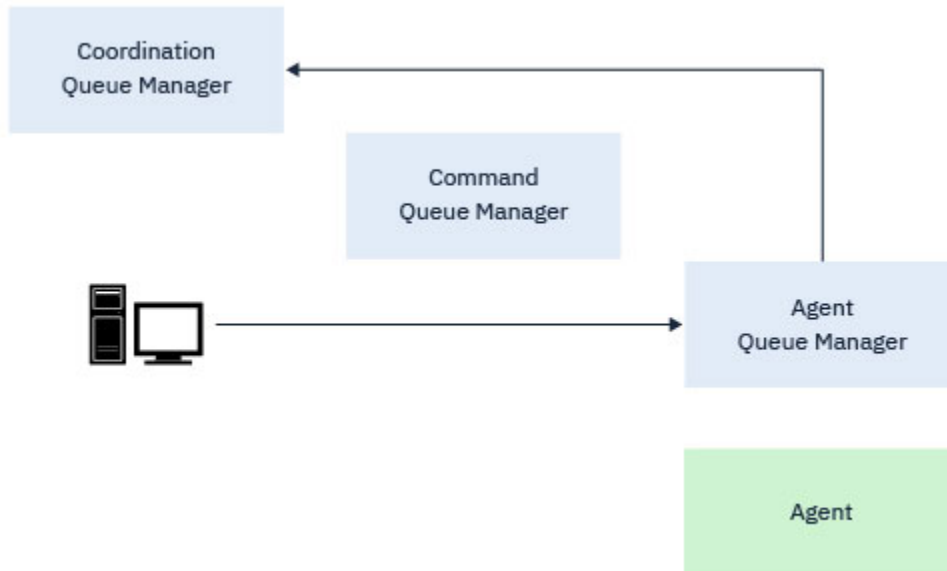


圖 12: 各種 **fteCreate** 指令以及 **fteDeleteAgent** 指令會連接至代理程式佇列管理程式，並將訊息傳送至協調佇列管理程式，以從拓撲登錄或取消登錄代理程式。



小心: **fteCleanAgent** 會連接至代理程式佇列管理程式，並從其系統佇列中移除該代理程式的任何狀態資訊。

執行此指令可能會影響整個拓撲。因此，您應該僅在 IBM 的指引下執行此指令。

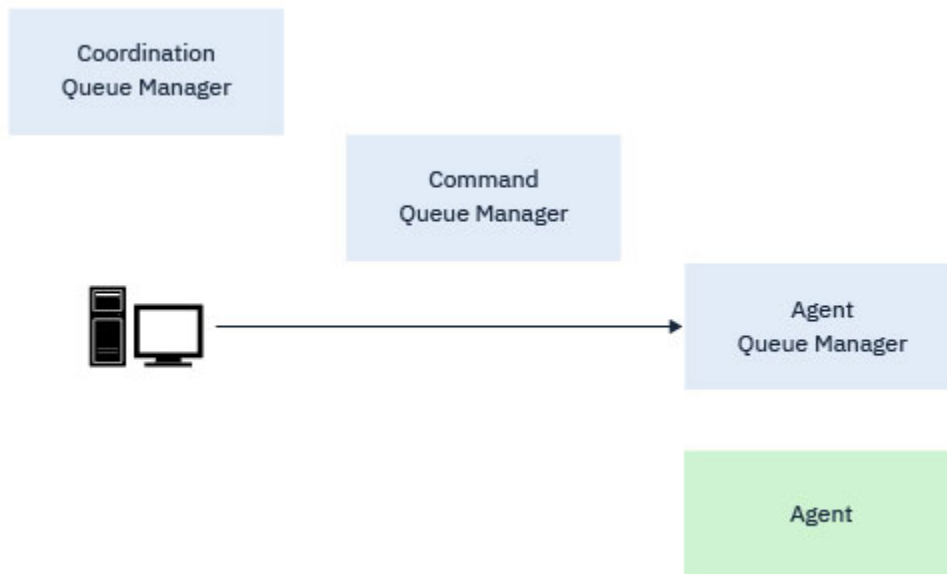


圖 13: **fteCleanAgent** 指令會直接連接至代理程式佇列管理程式，並從代理程式的系統佇列中移除狀態資訊

相關資訊

[已安裝的 MFT 指令集](#)

MFT 指令

列出了所有 Managed File Transfer 指令，及其詳細說明的鏈結。

表 35: Managed File Transfer 指令及其用途	
指令名稱	用途
移轉的指令:	
fteMigrateAgent	將代理程式及其配置從 IBM WebSphere MQ File Transfer Edition 7.0.4 移轉至 Managed File Transfer 7.5 或更新版本
fteMigrateConfigurationOptions	將 IBM WebSphere MQ File Transfer Edition 7.0 配置移轉至 IBM WebSphere MQ 7.5 或更新版本中的 Managed File Transfer。
fteMigrateLogger	將獨立式資料庫日誌程式的配置從 IBM WebSphere MQ File Transfer Edition 7.0.1 或更新版本移轉至 Managed File Transfer 7.5 或更新版本。
用於配置的指令:	
fteChangeDefaultConfigurationOptions	變更您希望 Managed File Transfer 使用的預設配置選項
fteCreateAgent	建立 Managed File Transfer Agent
fteCreateBridgeAgent	建立 Managed File Transfer 通訊協定橋接器代理程式
fteCreateCDAgent	建立 Managed File Transfer Connect:Direct 橋接器代理程式
V9.0.1 fteCreate 環境	為 Redistributable Managed File Transfer Agent 設定檔案配置及傳送的環境變數。
fteCreateLogger	建立 Managed File Transfer 日誌程式
fteDefine	產生定義指定物件所需的配置 Script。
fteDelete	產生移除指定物件所需的配置 Script。
fteDeleteAgent	刪除特定的 Managed File Transfer Agent


表 35: Managed File Transfer 指令及其用途 (繼續)	
指令名稱	用途
fteDeleteLogger	刪除 Managed File Transfer 日誌程式
fteModifyAgent	僅適用於 Windows。修改要以 Windows 服務方式執行的代理程式、Connect:Direct 橋接器代理程式或通訊協定橋接器代理程式。
fteModifyLogger	僅適用於 Windows。修改要以 Windows 服務方式執行的日誌程式。
fteSetupCommands	指定在發出指令時，連接至 IBM MQ 網路的佇列管理程式的詳細資料
fteSetupCoordination	配置 Managed File Transfer 協調佇列管理程式
用於管理的指令：	
fteAnt	在具有可用檔案傳送 Ant 作業的環境中執行 Ant Script。
fteCancelTransfer	取消檔案傳送
fteCleanAgent	清除代理程式使用的佇列
fteCreateMonitor	建立並啟動新的資源監視器
fteCreateTemplate	建立傳送範本以備將來使用
fteCreateTransfer	建立並啟動新的檔案傳送
fteDeleteMonitor	停止並移除現有的資源監視器
fteDeleteScheduledTransfer	刪除您先前排定的特定檔案傳送
fteDeleteTemplates	刪除現有的檔案傳送範本
fteListAgents	列出向特定協調佇列管理程式登錄的所有代理程式
fteListMonitors	列出向特定協調佇列管理程式登錄的所有資源監視器
fteListScheduledTransfers	列出您先前使用指令行或 IBM MQ Explorer 建立的所有 Managed File Transfer 傳送。
fteListTemplates	列出用於協調佇列管理程式的所有檔案傳送範本
ftePingAgent	對代理程式進行連線測試，以判斷代理程式是否處於作用中狀態，且可以處理傳送。
 fteSetProductID	設定 z/OS SCRT 錄製產品 ID
 fteShowAgentDetails	顯示特定代理程式的詳細資料
fteShowLoggerDetails	顯示特定日誌程式的詳細資料
fteStartAgent	先啟動特定代理程式，然後再使用此代理程式傳送檔案
fteStartLogger	啟動日誌程式
fteStopAgent	停止特定代理程式
fteStopLogger	停止日誌程式
用於安全的指令：	
fteObfuscate	為認證檔中的機密資料加密。
用於疑難排解的指令：	
fteDisplayVersion	顯示產品版本
 fteSetAgentLog 層次	針對 Managed File Transfer 通訊協定橋接器代理程式與 FTP/SFTP/FTPS 檔案伺服器之間的檔案傳送，啟用或停用診斷資訊記載。
fteSetAgentTraceLevel	設定要執行的代理程式追蹤的層次
fteSetLoggerTraceLevel	設定要執行的日誌程式追蹤層次
fteRAS	執行 RAS 收集工具


請參閱已安裝的 MFT 指令集，以取得顯示哪些指令隨哪些 Managed File Transfer 供應項目一起安裝的表格。

每個指令及其參數的語法，會顯示在名為鐵路圖的語法圖中。如需如何解譯鐵路圖的相關資訊，請參閱[如何看懂鐵路圖](#)。

使用 MFT 指令的權限

如果您想要發出 Managed File Transfer 指令，則使用者 ID 必須是 mqm 群組的成員（除非您已經將 IBM MQ 配置成容許不屬於 mqm 群組的使用者發出指令）。

 如需在 z/OS 上定義 mqm 替代群組的相關資訊，請參閱 [發出指令至 IBM MQ for z/OS](#)

如需授權的相關資訊，請參閱[管理 IBM MQ 的權限](#)。 如果您是使用 IBM i，請從下列主題開始：[IBM MQ 權限](#)。

可以使用 IBM MQ Explorer 發出部分 Managed File Transfer 指令。

從 Windows 和 UNIX 系統發出指令

請留意下列關於發出指令的環境特定資訊：

Managed File Transfer for Windows

所有指令均可從指令行發出。指令名稱不區分大小寫：您可以輸入大寫、小寫或大小寫混用的格式。不過，控制指令的引數（例如佇列名稱）及參數（例如佇列管理程式名稱的 **-m**）則須區分大小寫。

在語法說明中，以連字號 (-) 作為旗標指示器。

Managed File Transfer for UNIX 系統

所有 Managed File Transfer 指令均可從 Shell 發出。所有指令都須區分大小寫。

從 z/OS 系統發出指令



Managed File Transfer 指令會安裝在安裝產品時所選位置的 bin 子目錄中。可以透過下列任一選項執行指令：

- 直接從 USS 環境執行，方法是指定指令的路徑，或在使用者指令路徑中加入 bin 子目錄。
- 針對特定的代理程式或日誌程式，從配置自 PDSE 指令範本庫之指令的 PDSE 資料集執行。如需相關資訊，請參閱第 54 頁的『[建立 MFT 代理程式或日誌程式指令資料集](#)』。

從 IBM i 平台發出指令



請留意下列關於在 IBM i 上發出指令的環境特定資訊：

- 您可以使用 Qshell 直譯器來啟動 Managed File Transfer 指令。若要啟動 Qshell 直譯器，請從 IBM i 系統指令行發出 **STRQSH** 指令。
- 當您在 Qshell 環境中執行指令時，指令名稱不區分大小寫：您可以輸入大寫、小寫或大小寫混用的格式。不過，控制指令的引數（例如佇列名稱）及參數（例如佇列管理程式名稱的 **-m**）則須區分大小寫。

相關參考

第 284 頁的『[MFT 的回覆碼](#)』

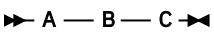
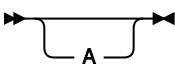
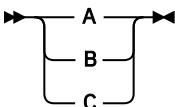
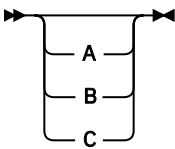
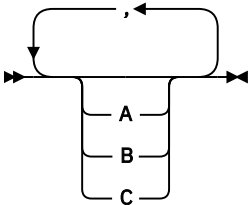
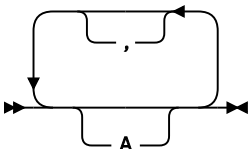
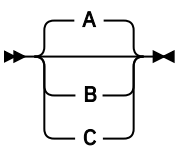
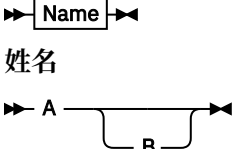
Managed File Transfer 指令、Ant 作業及日誌訊息提供回覆碼，指出功能是否順利完成。

如何閱讀鐵路（語法）圖

每個鐵路圖皆以一個右雙箭頭開始，而以一個右箭頭/左箭頭組合結束。起始於單一右箭頭的線條是接續線。從左到右、從上到下，遵循箭頭的方向閱讀鐵路圖。

在鐵路圖中使用的其他慣例如下所示：

表 36: 如何閱讀鐵路圖

慣例	意義
	您必須指定值 A、B 和 C。必要值會顯示在鐵路圖的主要線上。
	您可以指定值 A。選用值顯示在鐵路圖的主線下方。
	A、B 及 C 是替代值，您必須指定其中一個值。
	A、B 及 C 是替代值，您可指定其中一個值。
	您可以指定一或多個值 A、B 及 C。箭頭會顯示多個或重複值的任何必要分隔字元 (在此範例中，是逗點 (,))。
	您可多次指定值 A。本範例中的分隔字元是選用項目。
	A、B 及 C 是替代值，您可指定其中一個值。如果不指定如上所示的任何值，則會使用預設 A (在主要線條上方顯示的值)。
	鐵路片段 Name 與主要鐵路圖分開顯示。
標點及大寫值	按原樣指定所顯示的內容。
小寫值 (例如, <i>name</i>)	提供您自己的文字來取代 <i>name</i> 變數。

追蹤 MFT 指令

您可以追蹤任何 Managed File Transfer 指令，以協助從指令行進行問題判斷。

用途

對任何 MFT 指令使用 **-trace** 參數，以啟用指定層次的追蹤。除非包括了 **-tracePath** 參數來識別不同目錄，否則產生的追蹤檔案位於現行工作目錄中。

因為執行追蹤會大幅影響效能，而且會產生大量追蹤資料，因此，請謹慎執行追蹤，並且僅在必要時才執行。一般而言，只需要在 IBM 客戶服務代表要求時，再啟用追蹤即可。

您可以在 `agent.properties` 檔中設定進一步追蹤內容，例如，追蹤檔大小及要保留的追蹤檔數目。[進階代理程式內容](#)中說明了這些內容。

語法

```
► fteCommandName — -trace — (classes=level) — -tracePath — (directory path)
```

參數

-trace (classes=level)

必要項目。設定追蹤的層次，以及要套用追蹤的類別。請指定下列格式：

```
classes=level
```

例如：

```
com.ibm.wmqfte=all
```

其追蹤所有 Managed File Transfer 類別。

指定您要套用追蹤層次的類別規格清單（以冒號區隔）。如果未指定此參數，則追蹤層次會套用至所有代理程式類別。

如果 (*classes*) 的開頭有加號 (+)，則加號之後的追蹤類別清單，都會加入目前正在追蹤的任何現有追蹤類別中。

有效的追蹤層次選項如下，並且會依照追蹤檔案大小及其詳細資料遞增列出：

off

關閉代理程式追蹤，但繼續將資訊寫入日誌檔。這是預設選項。

flow

擷取與代理程式中的程序流程相關聯的追蹤點資料。

moderate

擷取追蹤中的適量診斷資訊。

verbose

擷取追蹤中的大量診斷資訊。

全部

設定對所有代理程式類別執行代理程式追蹤。

-tracePath (directory path)

選用項目。請指定您想要將追蹤寫入其中的目錄。例如，`c:\temp`。

z/OS 如果未指定此參數，則該值為發出指令的目錄。例如，在 z/OS 上：

```
z/OS /u/smith/fte/wmqft/mqft/logs/MQPV/loggers/BFGLG1/logs/
```

只有在已指定 **-trace** 參數時，此參數才有效。

範例

在此範例中，追蹤層次設為 all，表示 **fteStartAgent** 指令會追蹤屬於 AGENT.NAME 的所有類別：

註：代理程式啟動時，追蹤會進入 *mft_config/logscoordination_qmgr/agents/agent*

```
fteStartAgent -trace com.ibm.wmqfte=all -tracePath /u/mft/trace AGENT.NAME
```

在此範例中，針對代理程式 AGENT.NAME 的 com.ibm.wmqfte.common 類別，追蹤層次設為 moderate。**ftePingAgent** 指令會擷取適當追蹤量：

```
ftePingAgent -trace com.ibm.wmqfte.common=moderate AGENT.NAME
```

在此範例中，針對代理程式 AGENT.NAME 的 com.ibm.wmqfte.common 類別，追蹤層次設為 moderate，且追蹤寫入 c:\\$user 目錄中。**ftePingAgent** 指令會擷取適當追蹤量：

```
ftePingAgent -trace com.ibm.wmqfte.common=moderate -tracePath c:\$user AGENT.NAME
```

相關參考

第 227 頁的『疑難排解 MFT』

使用下列參照資訊來協助您診斷 Managed File Transfer 中的錯誤：

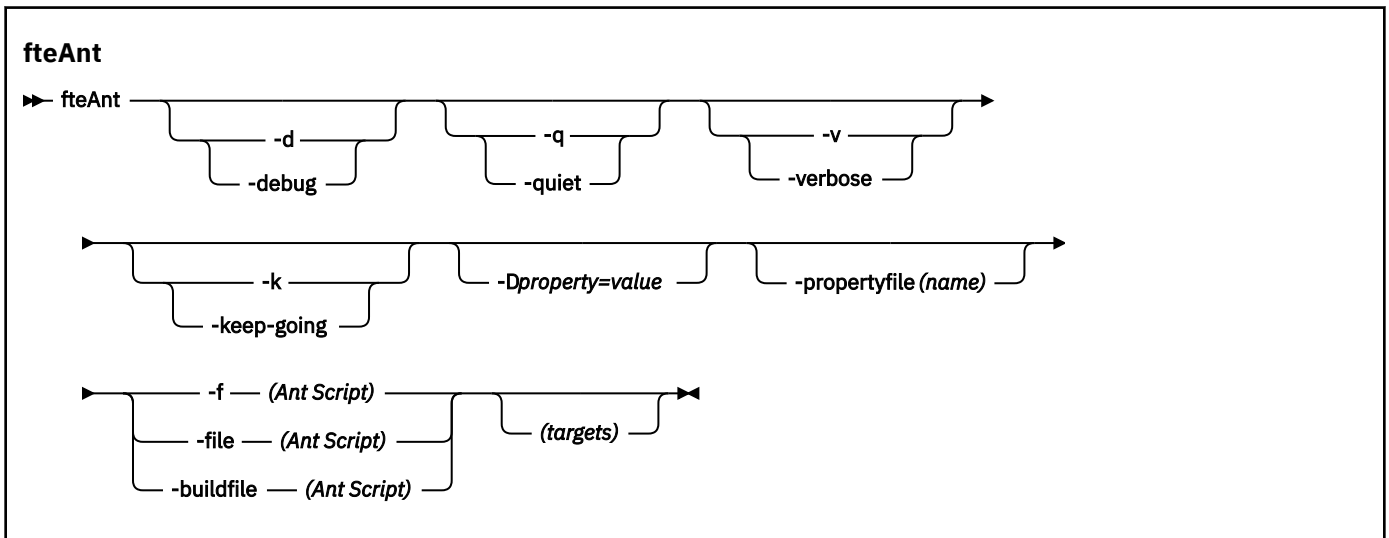
fteAnt: 在 MFT 中執行 Ant 作業

fteAnt 指令可在提供 Managed File Transfer Ant 作業的環境中執行 Ant Script。

用途

使用 **fteAnt** 指令可在具有 Managed File Transfer 的環境中執行 Ant Script。與標準 **ant** 指令不同，**fteAnt** 需要您定義 Script 檔。

語法



參數

-debug 或 **-d**

選用項目。產生除錯輸出。

-quiet 或 **-q**

選用項目。產生最小輸出。

-verbose 或 **-v**

選用項目。產生詳細輸出。

-keep-going 或 -k

選用項目。執行不依賴於失敗目標的所有目標。

-D *property*=*value*

選用項目。針對給定的 *property* 使用 *value*。設有 **-D** 的內容會優先於內容檔中所設定的內容。

使用 **com.ibm.wmqfte.propertyset** 內容可指定 Ant 作業所要使用的配置選項集。使用非預設協調佇列管理程式的名稱作為此內容的值。Ant 作業會使用與此非預設協調佇列管理程式相關聯的配置選項集。如果未指定此內容，則會使用基於預設協調佇列管理程式的預設配置選項集。如果為 Ant 作業指定 **cmdqm** 屬性，則此屬性優先於為 **fteAnt** 指令指定的配置選項集。無論您是使用預設配置選項集還是使用 **com.ibm.wmqfte.propertyset** 內容指定集，此行為均適用。

-propertyfile (*name*)

選用項目。載入檔案中的所有內容，並優先載入 **-D** 內容。

-f (*Ant script*)、-file (*Ant script*) 或 -buildfile (*Ant script*)

必要項目。指定要執行的 Ant Script 的名稱。

targets

選用項目。要從 Ant Script 執行的一個以上目標的名稱。如果您沒有指定此參數的值，則會執行 Script 的預設目標。

-version

選用項目。顯示 Managed File Transfer 指令及 Ant 版本。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，會執行 Ant Script `fte_script.xml` 中的目標 **copy**，指令會將除錯輸出寫入標準輸出。

```
fteAnt -d -f fte_script.xml copy
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

也可以從 Ant Script 指定其他狀態回覆碼（例如，透過使用 Ant 失敗作業來指定）。

如需相關資訊，請參閱 [失敗](#)。

z/OS

fteBatch、fteCommon 及 ftePlatform

z/OS

fteBatch、fteCommon 及 ftePlatform 是 Managed File Transfer 在 `MQ_INSTALLATION_PATH/bin` 目錄中提供作為 Helper Script 的 Script。並非每個平台上都會存在所有這些 Script。

fteBatch Script (僅限 z/OS)

z/OS

fteBatch 是一種 Helper Script，用於從「JZOS 批次啟動程式」執行 Managed File Transfer。fteBatch 只安裝在 z/OS 上。一般而言，Managed File Transfer 是使用所提供的指令 Shell Script 來啟動，這些 Script 會先執行一些環境配置，然後再啟動適用於該功能的 Java 類別。使用「JZOS 批次啟動程式」啟動 Managed

File Transfer 時，會直接從該啟動程式啟動 Java 類別。fteBatch 可以作為啟動程式設定的一部分進行呼叫，以將所需的類別名稱置於環境變數中，並執行一般指令 Shell Script 所執行的設定工作，然後再啟動 Java。這可在您的工作與 Managed File Transfer 所使用的內部類別名稱之間提供一層隔離。

Managed File Transfer 8.0 已淘汰 fteBatch 指令，因為您可以透過新的 PDSE 指令資料集來執行 Managed File Transfer。如需相關資訊，請參閱第 54 頁的『[建立 MFT 代理程式或日誌程式指令資料集](#)』。

fteCommon

fteCommon 是一種由其他 Managed File Transfer 指令 Script 啟動的 Helper Script，用於在啟動 Java 之前先執行一般設定處理作業。

ftePlatform

ftePlatform 是一種由 fteCommon Script 啟動的 Helper Script，用於執行平台專用的設定處理作業。

fteCancelTransfer

使用 **fteCancelTransfer** 指令來取消 Managed File Transfer 傳送。您可以對傳送的來源或目的地代理程式發出此指令。

用途

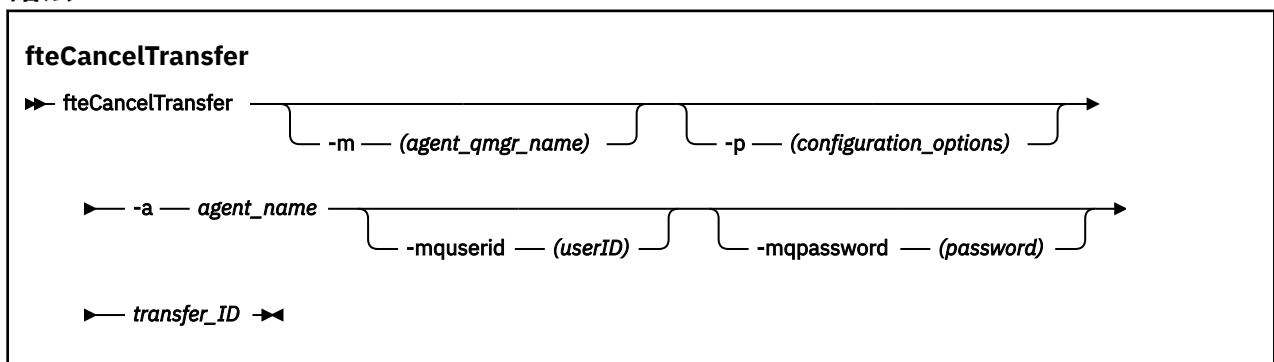
當該傳送目前正在進行時，如果您發出 **fteCancelTransfer** 指令，則在該傳送過程中已經傳送的任何檔案，均會保留在目的地系統中，並且不會刪除。在該傳送過程中部分傳送的任何檔案，均會從目的地系統中刪除。以 "已取消" 身分傳送之傳送日誌的目的地端。

如果取消傳送至 Connect:Direct 節點，則在取消的該傳送過程中部分傳送的任何檔案，均會保留在目的地系統中，並且不會刪除。

您可以從任何可以連接至 IBM MQ 網路的系統執行 **fteCancelTransfer** 指令，然後遞送至代理程式佇列管理程式。具體而言，為了讓指令能夠執行，您必須在此系統上安裝 Managed File Transfer，且必須已在此系統上配置 Managed File Transfer，才能與 IBM MQ 網路進行通訊。如果沒有可用的連線功能詳細資料，則只要有可用的代理程式佇列管理程式詳細資料，即會改用這些詳細資料進行連線。

僅在要使用一組不同於您的預設集的配置選項時，才為此指令指定選用的 **-p** 參數。如需相關資訊，請參閱 [配置選項](#)。

語法



參數

-m (agent_qmgr_name)

選用項目。代理程式佇列管理程式的名稱。此代理程式必須是您要取消的傳送之來源或目的地代理程式。如果未指定此參數，則會將取消要求傳送至您正在使用的配置選項集所識別的佇列管理程式。

-p (configuration_options)

選用項目。此參數會決定用於取消傳送的配置選項集。依照慣例，會使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，指令即會使用與此非預設協調佇列管理程式相關聯的一組內容檔。

如果未指定此參數，則會使用一組以預設協調佇列管理程式為基礎的配置選項。

-a (agent_name)

必要項目。您要取消的傳送之來源或目的地代理程式名稱。

-mquserid (userID)

選用項目。指定用來向指令佇列管理程式進行鑑別的使用者 ID。

-mqpassword (password)

選用項目。指定用來向指令佇列管理程式進行鑑別的密碼。您還必須指定 **-mquserid** 參數。如果指定了 **-mquserid**，卻未指定 **-mqpassword**，則系統會提示您提供相關聯的密碼。將不會顯示密碼。

transfer_ID

必要項目。您要取消的傳送 ID。發出 **fteCreateTransfer** 指令後，傳送 ID（也稱為要求 ID）即會顯示在指令行中。傳送 ID 也會包含在檔案傳送日誌訊息中，或顯示在「IBM MQ 探險家」的「傳送日誌」畫面中。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，AGENT1 為要取消的傳送之來源代理程式。

```
fteCancelTransfer -a AGENT1 414d5120514d5f4c4d343336303920201159c54820027102
```

回覆碼

0

指令已順利完成，或代理程式無法識別指定的傳送 ID。如果代理程式無法識別該傳送 ID，最有可能的原因是該傳送已完成或已取消。

1

指令未順利結束。


相關參考

第 367 頁的『[fteCreateTransfer: 啟動新的檔案傳送](#)』



fteCreateTransfer 指令從指令行建立並啟動新的檔案傳送。此指令可立即啟動檔案傳送、將檔案傳送排定在未來的時間和日期、將排定的傳送重複一次以上，以及根據特定條件觸發檔案傳送。

fteChangeDefaultConfigurationOptions

使用 **fteChangeDefaultConfigurationOptions** 指令，可變更您希望 Managed File Transfer 使用的預設配置選項。這些配置選項的值定義 Managed File Transfer 使用的內容檔群組。

重要:  在 IBM MQ for UNIX, Linux, and Windows 上，只有 IBM MQ 管理者（及 mqm 群組的成員）身分的使用者才能執行此指令。如果您嘗試以非 IBM MQ 管理者的使用者身分執行此指令，則會收到錯誤訊息 BFGCL0502E：您未獲授權執行所要求的作業。，且指令將不會執行。

 在 z/OS 系統上，使用者必須滿足（至少）下列其中一項條件才能執行移轉指令：

- 是 mqm 群組的成員（如果 mqm 群組已定義在系統上）。
-  是 BFG_GROUP_NAME 環境變數所指名之群組的成員（如果已指名群組的話）。
-  指令執行時，BFG_GROUP_NAME 環境變數中未設定任何值。

用途

當您第一次使用 `fteSetupCoordination` 指令將佇列管理程式配置為協調佇列管理程式時，會建立預設 Managed File Transfer 配置選項。在安裝 MFT 產品期間，如果 `mqft` 目錄尚未存在，則會在 `<MQ_DATA_PATH>` 下建立它。此外，如果配置、安裝及日誌目錄尚未存在，則會在 `mqft` 目錄下建立它們。

透過使用 `fteChangeDefaultConfigurationOptions` 指令，您可以變更 `installation.properties` 檔案中定義的預設協調佇列管理程式。如果您變更此協調佇列管理程式，Managed File Transfer 依預設會使用結構化目錄集，以及此目錄中所含用作 `configuration_options` 輸入的內容檔提供的配置選項。此目錄名稱與代理程式在此配置下使用的協調佇列管理程式相同。

如需 `installation.properties` 檔的相關資訊，請參閱 [第 49 頁的『多平台上的 MFT 配置選項』](#)。

語法

`fteChangeDefaultConfigurationOptions`

```
►► fteChangeDefaultConfigurationOptions — configuration_options ◄◄
```

參數

`configuration_options`

必要項目。此參數可指定您要變更至的預設配置選項。請使用非預設協調佇列管理程式的名稱作為此參數的輸入。

`-?` 或 `-h`

選用項目。顯示指令語法。

範例

在此範例中，預設配置選項將變更為 `QM_COORD2`：

```
fteChangeDefaultConfigurationOptions QM_COORD2
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

相關概念

[第 49 頁的『多平台上的 MFT 配置選項』](#)

Managed File Transfer 提供了一組內容檔，這些檔案包含有關您的設定的金鑰資訊，是作業所必要的。這些內容檔位於安裝產品時所定義的配置目錄中。

`fteCleanAgent`：清除 MFT 代理程式

使用 `fteCleanAgent` 指令，可透過從代理程式所使用的持續性及非持續性佇列中刪除訊息，來清除 Managed File Transfer Agent 使用的佇列。如果因資訊保留在代理程式使用的佇列中，而可能導致您在啟動代理程式時遇到問題，請使用 `fteCleanAgent` 指令。

用途

使用 `fteCleanAgent` 指令，可從代理程式使用的持續性及非持續性佇列中刪除訊息。具體而言，此指令可以執行下列動作：

- 在停止傳送之前，移除要傳入此代理程式或要從中傳出的任何進行中的傳送。當代理程式重新啟動時，不會回復這些傳送
- 移除已提交至代理程式，但尚未執行的任何指令
- 刪除代理程式上儲存的所有資源監視器
- 刪除代理程式上儲存的所有排定的傳送
- 刪除代理程式上儲存的所有無效訊息

如果代理程式是 Connect:Direct 橋接器代理程式，**-ms**、**-ss** 及 **-ims** 參數將無效。對於 Connect:Direct 橋接器代理程式，此指令還會執行下列動作：

- 從 Connect:Direct 橋接器代理程式在傳送檔案時暫時儲存檔案的目錄中，刪除所有檔案。此目錄的位置由 **cdTmpDir** 參數定義
- 顯示與任何進行中傳送相關聯的 Connect:Direct 程序的相關資訊

在 IBM MQ 9.0.3 及 IBM MQ 9.0.0 Fix Pack 2 之前，如果您只使用 **agent_name** 參數來執行 **fteCleanAgent** 指令，則依預設，該指令會如同已提供 **-all** 參數一樣執行，這會導致清除代理程式上的所有排定傳送、資源監視器及排定傳送定義。

V 9.0.0.2 **V 9.0.3** 從 IBM MQ 9.0.3 和 IBM MQ 9.0.0 Fix Pack 2，依預設，您必須將適當的參數傳遞至 **fteCleanAgent** 指令，並提供代理程式名稱，以指定要清除的 Managed File Transfer 狀態。這表示，依預設，**fteCleanAgent** 不會清除所指定代理程式的所有進行中和擱置中傳送、資源監視器定義及排定傳送定義。若要啟用或停用此行為，則可以將 `command.properties` 檔案中的 `failCleanAgentWithNoArguments` 內容設為適當的值：

- 依預設，`failCleanAgentWithNoArguments` 的值為 `true`，這表示如果僅指定 **agent_name** 參數，則 **fteCleanAgent** 指令無法執行。
- 如果 `failCleanAgentWithNoArguments` 設為 `false` 並且僅指定 **agent_name** 參數，則 **fteCleanAgent** 的行為方式和您指定 **-all** 參數時的行為相同。

必須在已停止的代理程式執行 **fteCleanAgent** 指令。如果嘗試在目前處於執行中狀態的代理程式上執行此指令，則會收到錯誤。此指令便不會啟動該代理程式。**fteCleanAgent** 指令可清除您發出該指令所在系統上的代理程式。您無法清除遠端系統上的代理程式。若要執行 **fteCleanAgent** 指令，您必須具有代理程式鎖定檔的寫入權，該檔案位於

`MQ_DATA_PATH\mqft\logs\coordination_QMgr_name\agents\agent_name\agent.lck`

FTEAGENT 群組必須在下列佇列上具有 GET 及 BROWSE 權限，才能順利執行 **fteCleanAgent**：

- `SYSTEM.FTE.COMMAND.agent_name`
- `SYSTEM.FTE.EVENT.agent_name`
- `SYSTEM.FTE.STATE.agent_name`

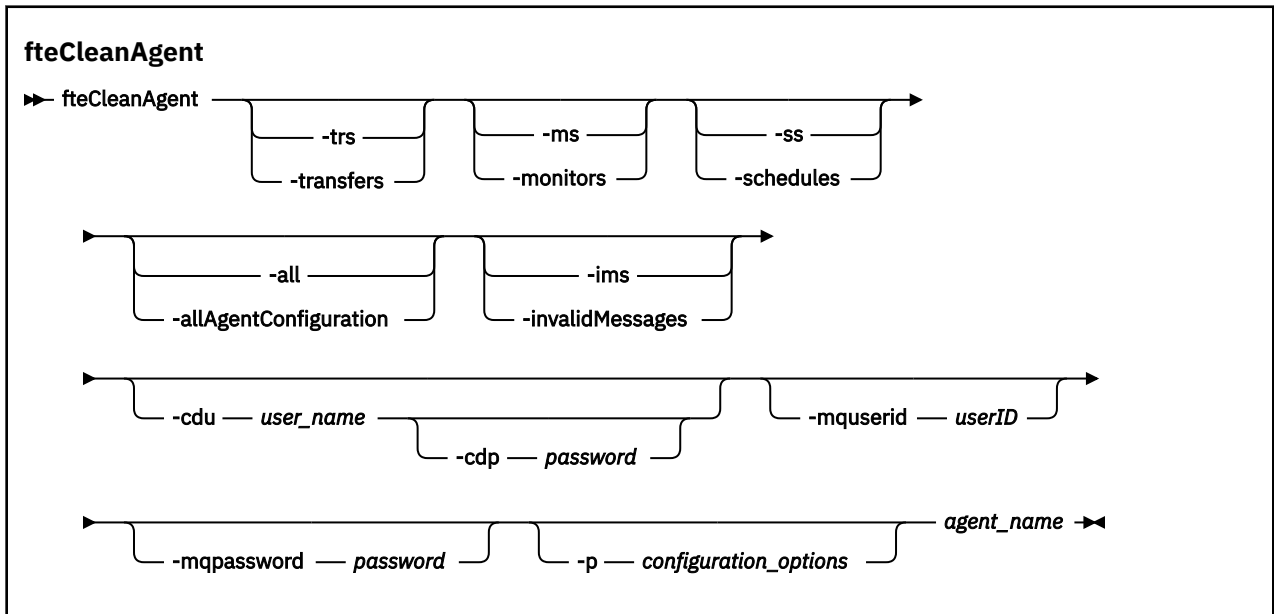
如需 FTEAGENT 群組及限制群組權限的進一步相關資訊，請參閱第 296 頁的『限制 MFT 特定資源的群組權限』。

如果要對以連結模式連接至其佇列管理程式的代理程式執行 **fteCleanAgent** 指令，且該代理程式剛剛停止執行，**fteCleanAgent** 指令可能會報告傳訊問題：MQRC 2042。發生此 MQRC 問題，是因為該代理程式的佇列控點仍存在於佇列管理程式中。在短暫的延遲之後，佇列管理程式會移除此控點，您即可重新發出 **fteCleanAgent**。

僅在要使用一組不同於您的預設集的配置選項時，才為此指令指定選用的 **-p** 參數。如需相關資訊，請參閱 [配置選項](#)。

註：清除 Connect:Direct 橋接器代理程式時，用來執行 **fteCleanAgent** 指令的使用者 ID，必須對 Connect:Direct 橋接器代理程式暫存目錄具有讀取權及寫入權。

語法



參數

您可以使用 **fteCleanAgent** 指令來刪除特定的構件。例如，您可以指定 **-trs** 指令來刪除擱置傳送，但不變更任何資源監視器及排定傳送。

-trs 或 **-transfers**

選用項目。指定要從代理程式刪除進行中以及擱置中的傳送。您不得與 **-all** 或 **-ims** 參數一起指定此參數。

-ms 或 **-monitors**

選用項目。指定要從代理程式刪除所有資源監視器定義。您不得與 **-all** 或 **-ims** 參數一起指定此參數。

-ss 或 **-schedules**

選用項目。指定要從代理程式刪除所有排定的傳送。您不得與 **-all** 或 **-ims** 參數一起指定此參數。

-all 或 **-allAgentConfiguration**

選用項目。指定要從代理程式刪除所有傳送、資源監視器定義及排定的傳送定義。您不得與 **-trs**、**-ss**、**-ms** 或 **-ims** 參數一起指定此參數。



小心: 只有在沒有其他選項可用時，才應該使用 **all** 參數。刪除傳送、資源監視器定義及排程傳送定義的動作可能會對您的企業產生重大影響。

-ims 或 **-invalidMessages**

選用項目。指定要從代理程式刪除所有無效訊息。您不得與 **-trs**、**-ss**、**-ms** 或 **-all** 參數一起指定此參數。

-cdu user_name

選用項目。僅在要清除的代理程式是 Connect:Direct 橋接器代理程式時才有效。如果指定此參數，此指令即會使用提供的使用者名稱，與 Connect:Direct 橋接器節點建立連線，並擷取現有 Connect:Direct 程序的其他相關資訊。如果未指定此參數，則會清除代理程式，而不顯示 Connect:Direct 程序的相關資訊。

-cdp password

選用項目。僅在要清除的代理程式是 Connect:Direct 橋接器代理程式，且您已指定 **-cdu** 參數時才有效。如果您指定 **-cdp** 參數，此指令即會使用提供的密碼，與 Connect:Direct 橋接器節點建立連線，並

擷取現有 Connect:Direct 程序的其他相關資訊。如果未指定此參數，但已指定 **-cdu** 參數，則會要求您以互動方式提供密碼。

-mquserid (userID)

選用項目。指定用來向代理程式佇列管理程式進行鑑別的使用者 ID。

-mqpassword (password)

選用項目。指定用來向代理程式佇列管理程式進行鑑別的密碼。您還必須指定 **-mquserid** 參數。如果指定了 **-mquserid**，卻未指定 **-mqpassword**，則系統會提示您提供相關聯的密碼。將不會顯示密碼。

-p (configuration_options)

選用項目。此參數會決定用於清除代理程式的配置選項集。依照慣例，會使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，指令即會使用與此非預設協調佇列管理程式相關聯的一組內容檔。

如果未指定此參數，則會使用一組以預設協調佇列管理程式為基礎的配置選項。

agent_name

必要項目。您要清除的 Managed File Transfer 代理程式的名稱。

-? 或 -h

選用項目。顯示指令語法。

範例

在以下基本範例中，將會清除 AGENT2 使用的所有佇列：

```
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.
```

在此範例中，將會清除 AGENT2 使用的所有無效訊息佇列：

```
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.
```

在此範例中，將會清除 Connect:Direct 橋接器代理程式 AGENT_CD_BRIDGE 使用的傳送佇列。

```
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.
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

相關參考

第 449 頁的『fteStopAgent』

使用 **fteStopAgent** 指令，以受管制的方式停止 Managed File Transfer 代理程式，或在必要時使用 **-i** 參數立即停止代理程式。

第 389 頁的『fteDeleteAgent』

fteDeleteAgent 指令會刪除 Managed File Transfer Agent 及其配置。如果代理程式為通訊協定橋接器代理程式，則使用者認證檔會留在檔案系統中。


第 460 頁的『MFT command.properties 檔』

command.properties 檔指定當您發出指令時要連接的指令佇列管理程式，以及 Managed File Transfer 聯絡該佇列管理程式所需的資訊。


fteCreateAgent (建立 MFT 代理程式)

fteCreateAgent 指令會建立 Managed File Transfer Agent 及其相關聯的配置。

您可以控制代理程式的存取權。如需進一步資訊，請參閱 [限制 MFT 代理程式動作的使用者權限](#)。您需要使用 **-ac** 參數，並提供存取部分佇列的權限。

重要：  在 IBM MQ for UNIX, Linux, and Windows 上，只有 IBM MQ 管理者（及 mqm 群組的成員）身分的使用者才能執行此指令。如果您嘗試以非 IBM MQ 管理者的使用者身分執行此指令，則會收到錯誤訊息 BFGCL0502E：您未獲授權執行所要求的作業。，且指令將不會執行。

 在 z/OS 系統上，使用者必須滿足（至少）下列其中一項條件才能執行移轉指令：

- 是 mqm 群組的成員（如果 mqm 群組已定義在系統上）。
-  是 BFG_GROUP_NAME 環境變數所指名之群組的成員（如果已指名群組的話）。

- **V9.0.0.1** 指令執行時，BFG_GROUP_NAME 環境變數中未設定任何值。

用途

使用 **fteCreateAgent** 指令可建立代理程式。此指令提供一些 MQSC 指令，必須針對代理程式佇列管理程式執行這些指令，才能建立下列代理程式佇列：

- 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

這些佇列是內部系統佇列，您不得從中修改、刪除或讀取訊息，除非您刪除代理程式。下列位置的檔案也提供了要執行的 MQSC 指令：

`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name\agent_name_create.mqsc`。

如果您稍後想要刪除代理程式，此指令也提供了您必須執行的 MQSC 指令，以清除並刪除代理程式使用的佇列。MQSC 指令位於下列位置的檔案中：

`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name\agent_name_delete.mqsc`。

Managed File Transfer 提供協助您配置代理程式的進階代理程式內容。[agent.properties](#) 檔中說明了這些內容。

您可能必須建立 `MQMFTCredentials.xml` 認證檔，才能使用您的代理程式。此檔案的範例位於 `MQ_INSTALLATION_PATH/mqft/samples/credentials/` 中。如需相關資訊與範例，請參閱第 696 頁的『MFT 認證檔格式』。

重要：

在 UNIX 平台及 Linux Managed File Transfer 指令上，使用 Socket 檔案與在相同主機上執行的代理程式處理程序進行通訊。

這些 Socket 檔案建立在代理程式的日誌目錄中，並在代理程式停止時刪除。在 IBM MQ Managed File Transfer 安裝中，此 Socket 檔案是以下列檔案路徑建立：`<MQ_DATA_PATH>/mqft/logs/<COORDINATION_QM_NAME>/agents/<AGENT_NAME>/logs/<AGENT_NAME>@<AGENT_QM_NAME>`，其中 `MQ_DATA_PATH` 依預設為 `/var/mqm`。

若為可重新配送的代理程式，會在下列目錄下建立此 Socket 檔案：`<RE_DISTRIBUTABLE_DIRECTORY>/mqft/logs/<COORDINATION_QM_NAME>/agents/<AGENT_NAME>/logs/<AGENT_NAME>@<AGENT_QM_NAME>`。

例如，如果代理程式名稱為 `SRCAGENT`，代理程式佇列管理程式名稱為 `SRCAGENTQM`，協調佇列管理程式名稱為 `COORDQM`，且可重新配送代理程式從目錄 `/home/myuser/mqmft-redis` 執行，則此 Socket 檔案的完整路徑為：`/home/myuser/mqmft-redis/mqft/logs/COORDQM/agents/SRCAGENT/logs/SRCAGENT@SRCAGENTQM`

檔案路徑長度總計為 85 個字元。

Socket 檔案的這些作業系統所容許的路徑長度上限為 107 個字元。因此，在建立代理程式時，請小心確定 Socket 檔案路徑不超過 107 個字元。這對於可重新配送的代理程式尤其重要，代理程式的日誌目錄可以位於任意目錄位置。如需設定配置目錄的詳細資料，請參閱 [fteCreateEnvironment](#) 指令。

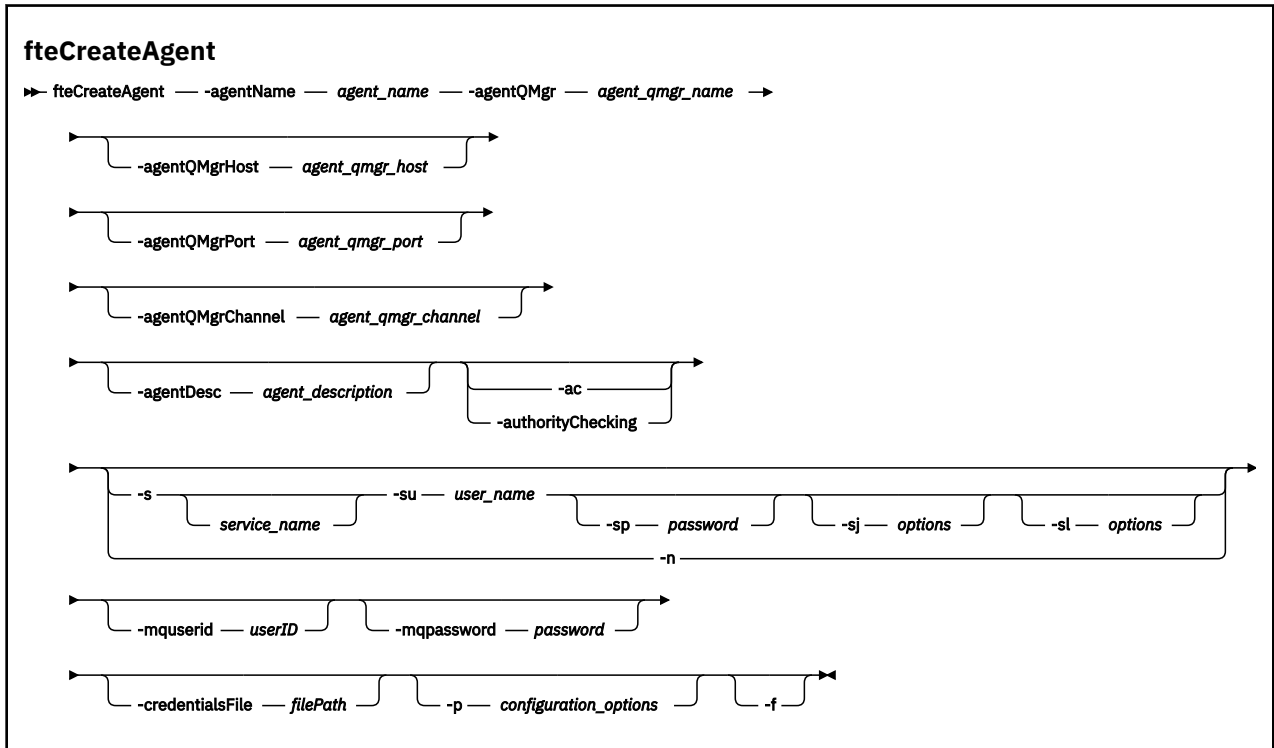
如果您啟動代理程式，或執行連接至代理程式的其他指令，且您的路徑長度超過 107 個字元，則會收到下列訊息：

BFGNV0159E: 嘗試連結至具有 FFDC 的 Socket 檔案失敗

特殊字元

如果使用的參數值包含特殊字元，請格外小心，以免指令 Shell 解譯字元的方式不符合預期。例如，如果完整檔案路徑和名稱包含諸如空格、引號（單引號或雙引號）、正斜線或反斜線字元等字元，則可能會被指令 Shell 解譯，而不是直接傳遞至指令本身。若要避免字元被指令 Shell 解譯，請用雙/單引號將整個參數括住，或者使用指令 Shell 的 ESC 序列來跳出特殊字元。

語法



參數

-agentName (agent_name)

必要項目。要建立的代理程式的名稱。代理程式名稱對其協調佇列管理程式必須是唯一的。

如需命名代理程式的相關資訊，請參閱物件命名慣例。

-agentQMGr (agent_qmgr_name)

必要項目。代理程式佇列管理程式的名稱。

-agentQMGrHost (agent_qmgr_host)

選用項目。代理程式佇列管理程式的主機名稱或 IP 位址。

-agentQMGrPort (agent_qmgr_port)

選用項目。用戶端連接代理程式佇列管理程式所使用的埠號。

-agentQMGrChannel (agent_qmgr_channel)

選用項目。用於連接至代理程式佇列管理程式的通道名稱。

-agentDesc (agent_description)

選用項目。代理程式的說明，顯示在「IBM MQ Explorer」中。

-ac 或 -authorityChecking

選用項目。此參數會啟用權限檢查。如果您指定此參數，代理程式會檢查是否已授權提交要求的使用者執行所要求的動作。如需相關資訊，請參閱 [限制 MFT 代理程式動作的使用者權限](#)。

Windows -s (service_name)

選用項目（僅限 Windows）。指出代理程式是以 Windows 服務方式執行，該指令必須從 Windows 管理者使用者 ID 執行。如果未指定 *service_name*，則此服務會命名為 `mqmftAgentAGENTQMGR`，其中 *AGENT* 是代理程式名稱，而 *QMGR* 是代理程式佇列管理程式名稱。

服務的顯示名稱（顯示在 Windows 服務視窗中的名稱直欄內）一律為 **Managed File Transfer Agent AGENT@QMGR**。

註：如果可轉散發的代理程式將以 Windows 服務形式來執行，則需要在系統環境中設定 **BFG_DATA** 環境變數，服務才能運作。

Windows -su (user_name)

選用項目（僅限 Windows）。代理程式以 Windows 服務方式執行時，此參數會指定服務執行時所使用的帳戶名稱。若要使用 Windows 網域使用者帳戶執行此代理程式，請以 `DomainName\UserName` 格式指定此值。若要使用本端內建網域中的帳戶執行服務，請以 `UserName` 格式指定值。

您使用 **-su** 參數指定的 Windows 使用者帳戶必須具有 **Log on as a service** 權限。如需如何授與此權限的相關資訊，請參閱 [以 Windows 服務方式執行 MFT 代理程式或日誌程式的指引](#)。

指定 **-s** 時需要。

Windows -sp (password)

選用項目（僅限 Windows）。

此參數僅在指定 **-s** 時有效。如果在指定 **-s** 參數時不指定此參數，會產生一則警告訊息。此訊息警告您，您必須先使用「Windows 服務」工具設定密碼，然後服務才能順利啟動。

Windows -sj (options)

選用項目（僅限 Windows）。以 Windows 服務方式啟動代理程式時，會以 **-D** 或 **-X** 的格式定義傳遞至 JVM 的選項清單。這些選項會使用 **#** 記號或分號 (**;**) 字元來區隔。如果您必須內含任何 **#** 或分號 (**;**) 字元，請將這些字元放在單引號內。

此參數僅在指定 **-s** 時有效。

Windows -sl (options)

選用項目（僅限 Windows）。設定 Windows 服務日誌層次。有效的選項是：`error`、`info`、`warn`、`debug`。預設值是 `info`。如果您在使用 Windows 服務時遇到問題，則此選項非常有用。將它設為 `debug`，可以在服務日誌檔中提供更詳細資訊。

此參數僅在指定 **-s** 時有效。

Windows -n

選用項目（僅限 Windows）。指示代理程式以正常程序執行。此參數與 **-s** 選項互斥。如果未指定其中一個 **-s** 參數及 **-n** 參數，則代理程式會配置為一般 Windows 處理程序。

-p (configuration_options)

選用項目。此參數會決定用於建立代理程式的配置選項集。依照慣例，會使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，`fteCreateAgent` 指令會使用與此非預設協調佇列管理程式相關聯的內容檔集。

只有在您想要使用不同於預設值的配置選項時，才需指定選用的 **-p** 參數。如果未指定此參數，則會使用一組以預設協調佇列管理程式為基礎的配置選項。

-mquserid (userID)

選用項目。指定用來向協調佇列管理程式進行鑑別的使用者 ID。

-mqpassword (password)

選用項目。指定用來向協調佇列管理程式進行鑑別的密碼。您還必須指定 **-mquserid** 參數。如果指定了 **-mquserid**，卻未指定 **-mqpassword**，則系統會提示您提供相關聯的密碼。將不會顯示密碼。

-credentialsFile (filePath)

選用項目。現有或新認證檔的完整檔案路徑，IBM MQ 鑑別詳細資料會新增到該認證檔中。

此指令支援將 IBM MQ 鑑別詳細資料的集合新增至名為 Managed File Transfer 的認證檔。已啟用 IBM MQ 連線鑑別時，請使用此指令。如果更新現有的詳細資料，則必須使用 **-f** 強制參數。

-credentialPath (credentials_path)

此指令定義要將認證資訊移轉至的位置。此參數可以是現有認證檔的目錄路徑，也可以是新認證檔的目錄路徑。 **z/OS** 在 z/OS 平台上，認證檔可以是預先存在的分割資料集延伸 (PDSE)。PDSE 可以包含現有成員，或認證檔的新成員。必須更新 PDSE 的現有成員，以包含認證檔。PDSE 的格式必須為分區塊變數。

-f

選用項目。強制指令改寫不符合的現有參數。指定此參數，並不會強制取代現有 Windows 服務代理程式。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，AGENT3 使用代理程式佇列管理程式 QM_NEPTUNE 建立，並使用預設協調佇列管理程式：

```
fteCreateAgent -agentName AGENT3 -agentQMGr QM_NEPTUNE  
-agentQMGrHost myhost.ibm.com -agentQMGrPort 1415 -agentQMGrChannel CHANNEL1
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

fteCreateBridgeAgent (建立及配置 MFT 通訊協定橋接器代理程式)

fteCreateBridgeAgent 指令會建立 Managed File Transfer 通訊協定橋接器代理程式及其相關聯的配置。為您要傳送檔案至其中及從中接收檔案的每一台檔案伺服器建立通訊協定橋接器代理程式。

重要: **ULW** 在 IBM MQ for UNIX, Linux, and Windows 上，只有 IBM MQ 管理者（及 mqm 群組的成員）身分的使用者才能執行此指令。如果您嘗試以非 IBM MQ 管理者的使用者身分執行此指令，則會收到錯誤訊息 BFGCL0502E：您未獲授權執行所要求的作業。，且指令將不會執行。

z/OS 在 z/OS 系統上，使用者必須滿足（至少）下列其中一項條件才能執行移轉指令：

- 是 mqm 群組的成員（如果 mqm 群組已定義在系統上）。
- **V9.0.0.1** 是 BFG_GROUP_NAME 環境變數所指名之群組的成員（如果已指名群組的話）。
- **V9.0.0.1** 指令執行時，BFG_GROUP_NAME 環境變數中未設定任何值。

用途

使用 **fteCreateBridgeAgent** 指令可建立通訊協定橋接器代理程式。如需如何使用通訊協定橋接器的概觀，請參閱 [通訊協定橋接器](#)。此 **fteCreateBridgeAgent** 指令提供一些 MQSC 指令，必須針對代理程式佇列管理程式執行這些指令，才能建立下列代理程式佇列：

- 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

這些佇列是內部系統佇列，您不得從中修改、刪除或讀取訊息，除非您刪除代理程式。下列位置的檔案中也提供了要執行的 MQSC 指令：

`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name\agent_name_create.mqsc`

如果您稍後想要刪除代理程式，此指令也提供了您必須執行的 MQSC 指令，以清除並刪除代理程式使用的佇列。MQSC 指令位於下列位置的檔案中：

`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name\agent_name_delete.mqsc`。

fteCreateBridgeAgent 指令會在下列目錄中建立 `ProtocolBridgeProperties.xml` XML 檔案：

`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name`。

使用者必須自行手動建立 `ProtocolBridgeCredentials.xml` 檔案，因為 **fteCreateBridgeAgent** 指令不會再建立此檔案。

`ProtocolBridgeCredentials.xml` 檔案容許您定義使用者名稱及認證資訊，通訊協定橋接器代理程式使用這些資訊自行向通訊協定伺服器授權；`ProtocolBridgeProperties.xml` 檔案容許您定義多台通訊協定檔案伺服器，以便您可以傳送至多個端點。

`MQ_INSTALLATION_PATH/mqft/samples/credentials/` 目錄中有範例

`ProtocolBridgeCredentials.xml`。如需相關資訊，請參閱 [第 481 頁的『通訊協定橋接器認證檔案格式』](#) 及 [第 484 頁的『通訊協定橋接器內容檔案格式』](#)。

如果您執行 **fteCreateBridgeAgent** 指令並指定預設通訊協定檔案伺服器 (`-bt` 參數)，此預設伺服器會包含在 `ProtocolBridgeProperties.xml` 檔案中，且其主機名稱用於伺服器名稱。使用 `-bt` 參數時，您需要指定下列參數：

- `-bh`
- `-btz`
- `-bm`
- `-bsl`
- `-bfe`
- `-bts`

如果您沒有指定預設伺服器，則 `ProtocolBridgeProperties.xml` 檔案中沒有任何項目；您必須先手動新增至少一台伺服器，然後才能進行傳送。

Managed File Transfer 提供進階代理程式內容，協助您配置通訊協定橋接器代理程式。與通訊協定橋接器相關的內容以 `protocol` 開頭。這些內容在 [進階代理程式內容: 通訊協定橋接器](#) 及 [進階代理程式內容: 通訊協定橋接器代理程式](#) 記載中有說明。如果您在通訊協定橋接器中看到非預期的行為，請檢閱這些 `protocol` 內容，並確保您已針對您的系統正確設定這些內容。

如果您看到 **fteCreateBridgeAgent** 指令的下列輸出：

```
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.
```

它指出無法聯絡協調佇列管理程式，並提供 IBM MQ 原因碼來解釋原因。此參考訊息可能指示目前無法使用協調佇列管理程式，或您未正確定義配置。

重要:

在 UNIX 平台及 Linux Managed File Transfer 指令上，使用 Socket 檔案與在相同主機上執行的代理程式處理程序進行通訊。

這些 Socket 檔案建立在代理程式的日誌目錄中，並在代理程式停止時刪除。在 IBM MQ Managed File Transfer 安裝中，此 Socket 檔案是以下列檔案路徑建立: <MQ_DATA_PATH>/mqft/logs/<COORDINATION_QM_NAME>/agents/<AGENT_NAME>/logs/<AGENT_NAME>@<AGENT_QM_NAME>，其中 MQ_DATA_PATH 依預設為 /var/mqm。

若為可重新配送的代理程式，會在下列目錄下建立此 Socket 檔案: <RE_DISTRIBUTABLE_DIRECTORY>/mqft/logs/<COORDINATION_QM_NAME>/agents/<AGENT_NAME>/logs/<AGENT_NAME>@<AGENT_QM_NAME>。

例如，如果代理程式名稱為 SRCAGENT，代理程式佇列管理程式名稱為 SRCAGENTQM，協調佇列管理程式名稱為 COORDQM，且可重新配送代理程式從目錄 /home/myuser/mqmft-redist 執行，則此 Socket 檔案的完整路徑為: /home/myuser/mqmft-redist/mqft/logs/COORDQM/agents/SRCAGENT/logs/SRCAGENT@SRCAGENTQM

檔案路徑長度總計為 85 個字元。

Socket 檔案的這些作業系統所容許的路徑長度上限為 107 個字元。因此，在建立代理程式時，請小心確定 Socket 檔案路徑不超過 107 個字元。這對於可重新配送的代理程式尤其重要，代理程式的日誌目錄可以位於任意目錄位置。如需設定配置目錄的詳細資料，請參閱 [fteCreateEnvironment](#) 指令。

如果您啟動代理程式，或執行連接至代理程式的其他指令，且您的路徑長度超過 107 個字元，則會收到下列訊息:

BFGNV0159E: 嘗試連結至具有 FFDC 的 Socket 檔案失敗

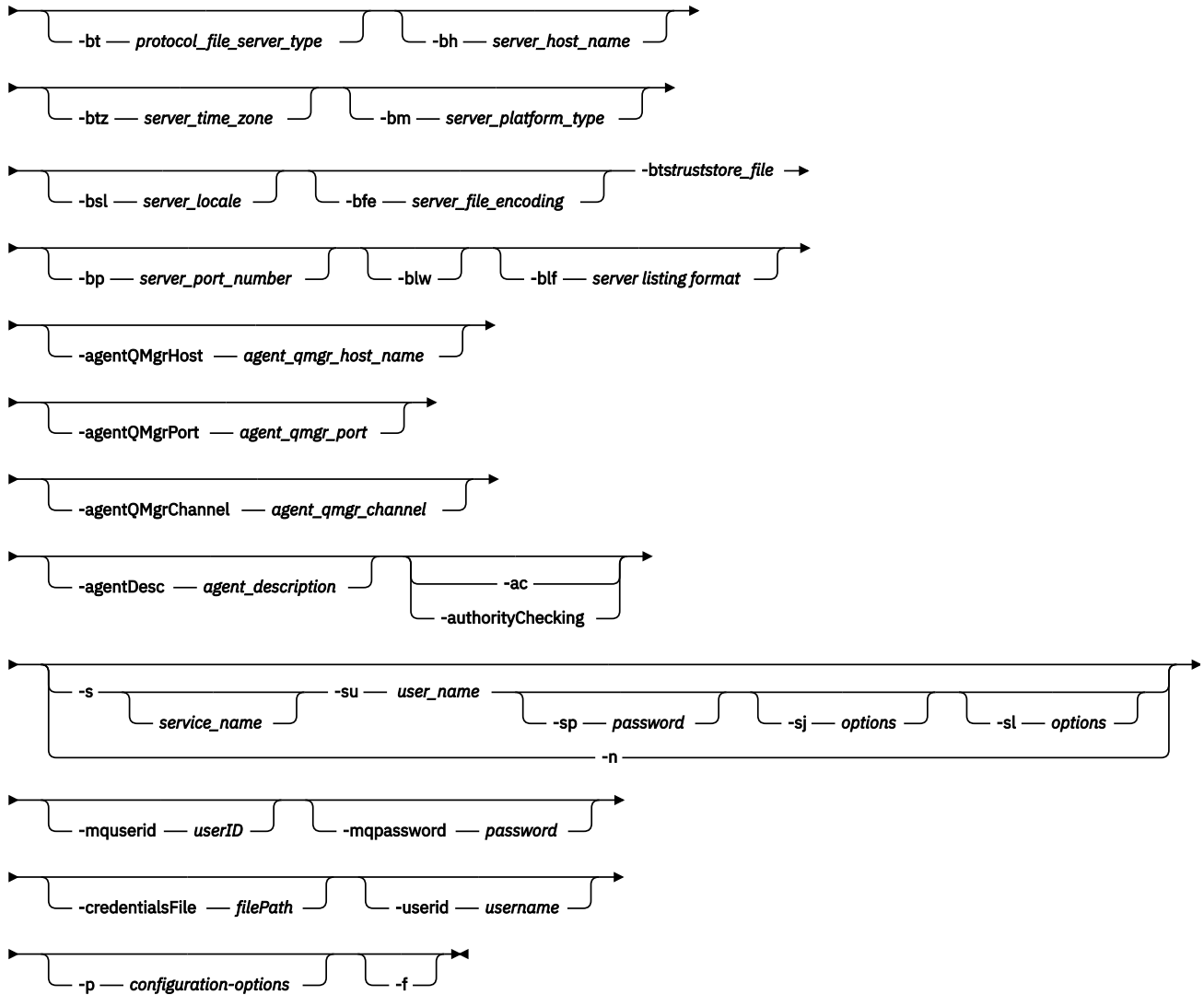
特殊字元

如果使用的參數值包含特殊字元，請格外小心，以免指令 Shell 解譯字元的方式不符合預期。例如，如果完整檔案路徑和名稱包含諸如空格、引號（單引號或雙引號）、正斜線或反斜線字元等字元，則可能會被指令 Shell 解譯，而不是直接傳遞至指令本身。若要避免字元被指令 Shell 解譯，請用雙/單引號將整個參數括住，或者使用指令 Shell 的 ESC 序列來跳出特殊字元。

語法

fteCreateBridgeAgent

► fteCreateBridgeAgent — -agentName *agent_name* -agentQMgr *agent_qmgr_name* →



參數

-agentName (*agent_name*)

必要項目。要建立的代理程式的名稱。代理程式名稱在其管理網域中必須是唯一的。

如需命名代理程式的相關資訊，請參閱物件命名慣例。

-agentQMgr (*agent_qmgr_name*)

必要項目。代理程式佇列管理程式的名稱。

-bt (*protocol_file_server_type*)

選用項目。指出您要定義預設通訊協定檔案伺服器。請指定下列其中一個選項：

FTP

標準 FTP 伺服器

SFTP

SSH FTP 伺服器

FTPS

使用 SSL 或 TLS 保護 FTP 伺服器

如果未指定此參數，將不會定義預設通訊協定伺服器。

-bh (server_host_name)

僅當您也使用 **-bt** 參數指定預設通訊協定檔案伺服器時才需要。通訊協定檔案伺服器的 IP 主機名稱或 IP 位址。

-btz (server_time_zone)

僅當您也指定 **-bt** 參數時才需要（僅限於 FTP 和 FTPS 伺服器）。通訊協定檔案伺服器的時區。請按下列格式指定時區：Area/Location。例如：Europe/London。

您可以使用 **-htz** 參數來列出 **-btz** 的可能值。例如：fteCreateBridgeAgent -htz

-bm (server_platform)

僅當您也使用 **-bt** 參數指定預設通訊協定檔案伺服器時才需要。通訊協定檔案伺服器的平台類型。請指定下列其中一個選項：

UNIX

一般 UNIX 和 Linux 平台

WINDOWS

一般 Windows 平台

-bsl (server_locale)

僅當您也指定 **-bt** 參數時才需要（僅限於 FTP 和 FTPS 伺服器）。通訊協定檔案伺服器的語言環境。請按下列格式指定語言環境：*xx_XX*。例如：en_GB。

- *xx* 是 ISO 語言碼。如需有效值的清單，請參閱[語言名稱表示法的代碼](#)
- *XX* 是 ISO 國碼。如需有效值的清單，請參閱[國家/地區名稱及代碼元素](#)

-bfe (server_file_encoding)

僅當您也使用 **-bt** 參數指定預設通訊協定檔案伺服器時才需要。通訊協定檔案伺服器上儲存的檔案的字元編碼格式。例如：UTF-8。

您可以使用 **-hcs** 參數來列出 **-bfe** 的可能值。例如：fteCreateBridgeAgent -hcs

-bts (truststore_file)

僅當您也指定 **-bt** 參數時才需要（僅限於 FTPS 伺服器）。指定用於驗證 FTPS 伺服器呈現的憑證的信任儲存庫路徑。

只有您也指定了 **-bt** 參數上的 FTPS 選項時，才能指定 **-bts** 參數。

-bp (server_port)

選用項目。通訊協定檔案伺服器所連接的 IP 埠。僅在您的通訊協定檔案伺服器未使用該通訊協定的預設埠時，才指定此參數。如果未指定此參數，Managed File Transfer 會使用檔案伺服器的通訊協定類型的預設埠。

-blw

選用項目。將通訊協定檔案伺服器定義為具有有限的寫入能力。依預設，通訊協定橋接器代理程式需要通訊協定檔案伺服器允許刪除檔案、重新命名檔案，以及開啟檔案來附加寫入。指定此參數可以讓通訊協定檔案伺服器不允許這些檔案動作。檔案伺服器改為僅允許讀取及寫入檔案。如果您指定此參數，在傳送岔斷時可能無法回復任何傳送，並可能會導致目前正在傳送的檔案失敗。

-blf (server_listing_format)

選用項目，僅限於 FTP 和 FTPS 伺服器。定義預設通訊協定檔案伺服器所傳回之列示檔案資訊的伺服器清單格式。選項如下所示：

UNIX

一般 UNIX 和 Linux 平台

WINDOWS

一般 Windows 平台

若要確認所應選取的格式，則可使用 FTP 用戶端程式執行目錄列示作業，從而選取最適用的格式。例如：

UNIX UNIX 會顯示下列類型的清單：

```
-rwxr-xr-x 2 userid groupId 4096 2009-07-23 09:36 filename
```

Windows Windows 會顯示下列類型的清單：

```
437,909 filename
```

IBM i IBM i 會顯示下列類型的清單：

```
OS400IFS -rwxrwsrwx 3 USERID 0 8192 Mar 7 08:33 filename
```

預設值為 UNIX，這是大部分伺服器所使用的格式。

-agentQMGrHost (agent_qmgr_host)

選用項目。代理程式佇列管理程式的主機名稱或 IP 位址。

-agentQMGrPort (agent_qmgr_port)

選用項目。用戶端連接代理程式佇列管理程式所使用的埠號。

-agentQMGrChannel (agent_qmgr_channel)

選用項目。用於連接至代理程式佇列管理程式的通道名稱。

-agentDesc (agent_description)

選用項目。代理程式的說明，顯示在「IBM MQ Explorer」中。

-ac 或 -authorityChecking

選用項目。此參數會啟用權限檢查。如果您指定此參數，代理程式會檢查是否已授權提交要求的使用者執行所要求的動作。如需相關資訊，請參閱 [限制 MFT 代理程式動作的使用者權限](#)。

Windows -s (service_name)

選用項目（僅限 Windows）。指出代理程式是以 Windows 服務方式執行。如果未指定 *service_name*，則此服務會命名為 *mqmftAgentAGENTQMGR*，其中 *AGENT* 是代理程式名稱，而 *QMGR* 是代理程式佇列管理程式名稱。

服務的顯示名稱（顯示在 Windows 服務視窗中的名稱直欄內）一律為 **Managed File Transfer Agent AGENT@QMGR**。

Windows -su (user_name)

選用項目（僅限 Windows）。代理程式以 Windows 服務方式執行時，此參數會指定服務執行時所使用的帳戶名稱。若要使用 Windows 網域使用者帳戶執行此代理程式，請以 *DomainName\UserName* 格式指定此值。若要使用本端內建網域中的帳戶執行服務，請以 *UserName* 格式指定值。

您使用 **-su** 參數指定的 Windows 使用者帳戶必須具有 **Log on as a service** 權限。如需如何授與此權限的相關資訊，請參閱 [以 Windows 服務方式執行 MFT 代理程式或日誌程式的指引](#)。

指定 **-s** 時需要。

Windows -sp (password)

選用項目（僅限 Windows）。**-su** 參數所設定使用者帳戶的密碼。

此參數僅在指定 **-s** 時有效。如果在指定 **-s** 參數時不指定此參數，會產生一則警告訊息。此訊息警告您，您必須先使用「Windows 服務」工具設定密碼，然後服務才能順利啟動。

Windows -sj (options)

選用項目（僅限 Windows）。以 Windows 服務方式啟動代理程式時，會以 **-D** 或 **-X** 的格式定義傳遞至 JVM 的選項清單。這些選項會使用 **#** 記號或分號 (**;**) 字元來區隔。如果您必須內含任何 **#** 或分號 (**;**) 字元，請將這些字元放在單引號內。

只有在指定 **-s** 時，此參數才有效。

Windows -sl (options)

選用項目（僅限 Windows）。設定 Windows 服務日誌層次。有效的選項是：error、info、warn、debug。預設值是 info。如果您在使用 Windows 服務時遇到問題，則此選項非常有用。將它設為 debug，可以在服務日誌檔中提供更詳細資訊。

此參數僅在指定 **-s** 時有效。

Windows -n

選用項目（僅限 Windows）。指示代理程式以正常程序執行。此參數與 **-s** 選項互斥。如果未指定其中一個 **-s** 參數及 **-n** 參數，則代理程式會配置為一般 Windows 處理程序。

-p (configuration-options)

選用項目。此參數會決定用於建立代理程式的配置選項集。依照慣例，會使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，**fteCreateBridgeAgent** 指令會使用與此非預設協調佇列管理程式相關聯的內容檔集。

只有在您想要使用不同於預設值的配置選項時，才需指定選用的 **-p** 參數。如果您沒有指定 **-p**，則會使用 `installation.properties` 檔案中定義的配置選項。如需相關資訊，請參閱[配置選項](#)。

-f

選用項目。強制指令改寫現有的配置。

-htz

選用項目。顯示您可以用作 **-btz** 參數的輸入的受支援時區清單。

-hcs

選用項目。顯示可用作 **-bfe** 參數的輸入的受支援字集清單。

執行 **fteCreateBridgeAgent -hcs** 指令，可列出 JVM 的已知字碼頁。這項資訊無法從外部來源取得，因為已知字碼頁會隨 JVM 而改變。

-mquserid (userID)

選用項目。指定用來向指令佇列管理程式進行鑑別的使用者 ID。

-mqpassword (password)

選用項目。指定用來向指令佇列管理程式進行鑑別的密碼。您還必須指定 **-mquserid** 參數。如果指定了 **-mquserid**，卻未指定 **-mqpassword**，則系統會提示您提供相關聯的密碼。將不會顯示密碼。

-credentialsFile (filePath)

選用項目。現有或新認證檔的完整檔案路徑，IBM MQ 鑑別詳細資料會新增到該認證檔中。

此指令支援將 IBM MQ 鑑別詳細資料的集合新增至名為 Managed File Transfer 的認證檔。已啟用 IBM MQ 連線鑑別時，請使用此指令。如果更新現有的詳細資料，則必須使用 **-f** 強制參數。

-userid (username)

選用項目。用來與認證詳細資料相關聯的使用者 ID。如果未指定使用者 ID，則認證詳細資料將套用至所有使用者。您還必須指定 **-credentialsFile** 參數。

-? 或 -h

選用項目。顯示指令語法。

淘汰的參數

下列參數已淘汰，且在 IBM WebSphere MQ 7.5 或 IBM WebSphere MQ File Transfer Edition 7.0.2 或更新版本上不受支援。

-brd (reconnect_delay)

已棄用。選用項目。指定重新建立中斷的通訊協定檔案伺服器連線嘗試之間的延遲期間（以秒為單位）。預設值為 10 秒。

-brr (reconnect_retries)

已棄用。選用項目。指定當預設通訊協定檔案伺服器連線中斷時，能夠嘗試重建連線的次數上限。達到此次數上限時，現行檔案傳送會分類為失敗。預設值為 2。

範例

在此範例中，新的通訊協定橋接器代理程式 ACCOUNTS1 使用代理程式佇列管理程式 QM_ACCOUNTS 建立，並使用預設協調佇列管理程式。ACCOUNTS1 連接至 FTP 伺服器 accountshost.ibm.com。此 FTP 伺服器是在 Windows 上執行，並使用歐洲/柏林的時區、de_DE 的語言環境，以及 UTF-8 的檔案編碼。重新連接重試次數為 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
```

在此範例中，新的通訊協定橋接器代理程式 ACCOUNTS2 使用代理程式佇列管理程式 QM_ACCOUNTS 建立，並使用預設協調管理程式。這時會建立沒有預設通訊協定檔案伺服器的 ACCOUNTS2。

```
fteCreateBridgeAgent -agentName ACCOUNTS2 -agentQMGr QM_ACCOUNTS
```

註：以上不適用於 Managed File Transfer Agent 可重新配送套件。

其他自訂作業

如果已使用 -bt 參數（及其他必要參數），則 ProtocolBridgeProperties.xml 檔案中會存在預設伺服器名稱。

如果您要新增其他 ftp 伺服器，或變更認證檔的位置，請參閱 [使用 ProtocolBridgeProperties.xml 檔案定義通訊協定檔案伺服器的內容](#)。

回覆碼

0

指令已順利完成。

1

指令未順利結束。

使用 **fteStartAgent** 指令可啟動通訊協定橋接器代理程式。如需相關資訊，請參閱 [第 445 頁的](#)

『fteStartAgent』。  另請參閱 [在 z/OS](#)。

相關參考

[第 481 頁的『通訊協定橋接器認證檔案格式』](#)

Managed File Transfer Agent 配置目錄中的 ProtocolBridgeCredentials.xml 檔案定義通訊協定橋接器代理程式用來向通訊協定伺服器授權其本身的使用者名稱及認證資訊。

[第 484 頁的『通訊協定橋接器內容檔格式』](#)

代理程式配置目錄中的 ProtocolBridgeProperties.xml 檔案，會定義通訊協定檔案伺服器的內容。

相關資訊

[通訊協定橋接器](#)

fteCreateEnvironment (為 Redistributable Managed File Transfer Agent 設定環境)

fteCreateEnvironment 指令會設定 Redistributable Managed File Transfer Agent 的檔案配置及傳送環境。

用途

使用 **fteCreateEnvironment** 指令來設定環境以使用 Redistributable Managed File Transfer Agent。您可以搭配 **-d** 參數來執行此指令，為 MFT 代理程式資料檔指定位置。如果您沒有指定 **-d** 參數，則該指令會在 Redistributable Managed File Transfer Agent 下載位置建立資料檔，並設定資料路徑。

語法

fteCreateEnvironment

```

▶ fteCreateEnvironment -d data_location -? 或 -h
  
```

參數

-d (*data_location*)

選用項目。在設定環境時，此參數用於指定資料檔的位置。

如果您沒有指定此參數，則會在 Redistributable Managed File Transfer Agent 進行解壓縮的位置建立資料目錄（如果此目錄尚未存在），並為此位置設定環境變數 (BFG_DATA)。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，Windows 上的 **-d** 參數指定建立資料夾的位置：

```
fteCreateEnvironment -d C:\mftRedistributable\mftData
```

在 Linux 上，作為必要條件，必須在 Bash Shell 上執行此指令。在 Bash Shell 中，可以按不同方式執行此指令，但需要設定指令檔的來源。

```
source Path_of_MFTZipBin/fteCreateEnvironment
```

替代方法如下：

```
. Path_of_MFTZipBin/fteCreateEnvironment
```

或者，如果是從指令檔所在的目錄執行：

```
./fteCreateEnvironment
```



小心： 請注意第一個句點字元 (.) 後面的空格

回覆碼

0

指令已順利完成。

1

指令未順利結束。

fteCreateCDAgent (建立 Connect:Direct 橋接器代理程式)

fteCreateCDAgent 指令可建立 Managed File Transfer Agent 及其與 Connect:Direct 橋接器搭配使用的關聯配置。

重要: `ULW` 在 IBM MQ for UNIX, Linux, and Windows 上，只有 IBM MQ 管理者（及 mqm 群組的成員）身分的使用者才能執行此指令。如果您嘗試以非 IBM MQ 管理者的使用者身分執行此指令，則會收到錯誤訊息 BFGCL0502E：您未獲授權執行所要求的作業。，且指令將不會執行。

z/OS 在 z/OS 系統上，使用者必須滿足（至少）下列其中一項條件才能執行移轉指令：

- 是 mqm 群組的成員（如果 mqm 群組已定義在系統上）。
- `V9.0.0.1` 是 BFG_GROUP_NAME 環境變數所指名之群組的成員（如果已指名群組的話）。
- `V9.0.0.1` 指令執行時，BFG_GROUP_NAME 環境變數中未設定任何值。

用途

使用 **fteCreateCDAgent** 指令來建立 Connect:Direct 橋接器代理程式。此類型的代理程式專用於從 Connect:Direct 節點來回傳送檔案。如需相關資訊，請參閱 [Connect:Direct 橋接器](#)。如需 Connect:Direct 橋接器所支援作業系統版本的詳細資料，請參閱 [IBM MQ 系統需求網頁](#)。

此指令提供一些 MQSC 指令，必須針對代理程式佇列管理程式執行這些指令，才能建立下列代理程式佇列：

- 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

這些佇列是內部系統佇列，您不得從中修改、刪除或讀取訊息，除非您刪除代理程式。下列位置的檔案也提供了要執行的 MQSC 指令：

`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name\agent_name_create.mqsc`。

如果您稍後想要刪除此代理程式，這個指令也提供您 MQSC 指令，您必須執行它們，才能清除及刪除屬於此代理程式的佇列。MQSC 指令位於下列位置的檔案中：

`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name\agent_name_delete.mqsc`。

Managed File Transfer 提供協助您配置代理程式的進階代理程式內容。這些內容在 [MFT agent.properties](#) 檔案中有說明。

fteCreateCDAgent 指令會在代理程式內容目錄中建立兩個 XML 檔案。

`ConnectDirectNodeProperties.xml`，用於定義傳送中遠端節點的相關資訊，以及

`ConnectDirectProcessDefinitions.xml`，用於指定由傳送啟動哪些使用者定義 Connect:Direct 處理程序。

若要定義 Connect:Direct 橋接器代理程式用來連接至 Connect:Direct 節點的使用者名稱及密碼，您必須手動建立 `ConnectDirectCredentials.xml` 檔案。範例 XML 檔案位於 `MQ_INSTALLATION_PATH/mqft/`

samples/credentials/ 中。如需相關資訊與範例，請參閱第 491 頁的『[Connect:Direct 認證檔格式](#)』。

重要:

在 UNIX 平台及 Linux Managed File Transfer 指令上，使用 Socket 檔案與在相同主機上執行的代理程式處理程序進行通訊。

這些 Socket 檔案建立在代理程式的日誌目錄中，並在代理程式停止時刪除。在 IBM MQ Managed File Transfer 安裝中，此 Socket 檔案是以下列檔案路徑建立: <MQ_DATA_PATH>/mqft/logs/<COORDINATION_QM_NAME>/agents/<AGENT_NAME>/logs/<AGENT_NAME>@<AGENT_QM_NAME>，其中 MQ_DATA_PATH 依預設為 /var/mqm。

若為可重新配送的代理程式，會在下列目錄下建立此 Socket 檔案: <RE_DISTRIBUTABLE_DIRECTORY>/mqft/logs/<COORDINATION_QM_NAME>/agents/<AGENT_NAME>/logs/<AGENT_NAME>@<AGENT_QM_NAME>。

例如，如果代理程式名稱為 SRCAGENT，代理程式佇列管理程式名稱為 SRCAGENTQM，協調佇列管理程式名稱為 COORDQM，且可重新配送代理程式從目錄 /home/myuser/mqmft-redist 執行，則此 Socket 檔案的完整路徑為: /home/myuser/mqmft-redist/mqft/logs/COORDQM/agents/SRCAGENT/logs/SRCAGENT@SRCAGENTQM

檔案路徑長度總計為 85 個字元。

Socket 檔案的這些作業系統所容許的路徑長度上限為 107 個字元。因此，在建立代理程式時，請小心確定 Socket 檔案路徑不超過 107 個字元。這對於可重新配送的代理程式尤其重要，代理程式的日誌目錄可以位於任意目錄位置。如需設定配置目錄的詳細資料，請參閱 [fteCreateEnvironment](#) 指令。

如果您啟動代理程式，或執行連接至代理程式的其他指令，且您的路徑長度超過 107 個字元，則會收到下列訊息:

BFGNV0159E: 嘗試連結至具有 FFDC 的 Socket 檔案失敗

特殊字元

如果使用的參數值包含特殊字元，請格外小心，以免指令 Shell 解譯字元的方式不符合預期。例如，如果完整檔案路徑和名稱包含諸如空格、引號（單引號或雙引號）、正斜線或反斜線字元等字元，則可能會被指令 Shell 解譯，而不是直接傳遞至指令本身。若要避免字元被指令 Shell 解譯，請用雙/單引號將整個參數括住，或者使用指令 Shell 的 ESC 序列來跳出特殊字元。

fteCreateCDAgent

► fteCreateCDAgent —agentName — agent_name — -agentQMGr — agent_qmgr_name →

► -cdNode — cd_node_name — -agentQMGrHost — agent_qmgr_host

► -agentQMGrPort — agent_qmgr_port

► -agentQMGrChannel — agent_qmgr_channel

► -agentDesc — agent_description — -ac — -authorityChecking

► -p — 配置選項 — -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

參數

-agentName (agent_name)

必要項目。要建立的代理程式的名稱。代理程式名稱對其協調佇列管理程式必須是唯一的。

如需命名代理程式的相關資訊，請參閱物件命名慣例。

-agentQMGr (agent_qmgr_name)

必要項目。代理程式佇列管理程式的名稱。

-cdNode cd_node_name

必要項目。Connect:Direct 節點的名稱，該節點用於將訊息從此代理程式傳送至目的地 Connect:Direct 節點。此參數的值用於記載，而非用於指定節點要連接的 Connect:Direct 橋接器代理程式。

-cdNodeHost 及 -cdNodePort 的值，會指定屬於 Connect:Direct 橋接器的 Connect:Direct 節點。

-agentQMGrHost (agent_qmgr_host)

選用項目。代理程式佇列管理程式的主機名稱或 IP 位址。

-agentQMGrPort (agent_qmgr_port)

選用項目。用戶端連接代理程式佇列管理程式所使用的埠號。

-agentQMGrChannel (agent_qmgr_channel)

選用項目。用於連接至代理程式佇列管理程式的通道名稱。

-agentDesc (agent_description)

選用項目。代理程式的說明，顯示在「IBM MQ Explorer」中。

-ac 或 -authorityChecking

選用項目。此參數會啟用權限檢查。如果您指定此參數，代理程式會檢查是否已授權提交要求的使用者執行所要求的動作。如需相關資訊，請參閱 [限制 MFT 代理程式動作的使用者權限](#)。

-p (configuration_options)

選用項目。此參數會決定用於建立代理程式的配置選項集。依照慣例，會使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，**fteCreateCDAgent** 指令會使用與此非預設協調佇列管理程式相關聯的內容檔集。

只有在您想要使用不同於預設值的配置選項時，才需指定選用的 **-p** 參數。如果未指定此參數，則會使用一組以預設協調佇列管理程式為基礎的配置選項。

-f

選用項目。強制指令改寫不符合的現有參數。指定此參數，並不會強制取代現有 Windows 服務代理程式。

-cdNodeHost cd_node_host_name

選用項目。由 **-cdNode** 參數指定的 Connect:Direct 節點所在系統的主機名稱或 IP 位址。如果您沒有指定 **-cdNodeHost** 參數，則會使用本端系統主機名稱或 IP 位址的預設值。

在大部分情況下，Connect:Direct 節點會與 Connect:Direct 橋接器代理程式位於相同的系統上。在這些情況下，此內容的預設值（即本端系統的 IP 位址）是正確的。如果系統具有多個 IP 位址，或 Connect:Direct 節點是在與 Connect:Direct 橋接器代理程式不同的系統上，且其系統均共用一個檔案系統，請使用此內容為 Connect:Direct 節點指定正確的主機名稱。

-cdNodePort cd_node_port_name

選用項目。Connect:Direct 節點的埠號，用戶端應用程式會使用此節點與 **-cdNode** 參數指定的節點通訊。在 Connect:Direct 產品文件中，此埠也稱為 API 埠。如果您沒有指定 **-cdNodePort** 參數，則會假設預設埠號為 1363。

-cdTmpDir cd_tmp_directory

選用項目。在將檔案傳送至目的地 Connect:Direct 節點之前，此代理程式要用於暫時儲存這些檔案的目錄。此參數會指定暫時儲存檔案的目錄之完整路徑。比方說，例如將 **cdTmpDir** 設定為 /tmp，則會將檔案暫時放置在 /tmp 目錄中。如果您沒有指定 **-cdTmpDir** 參數，則會將檔案暫時儲存在名為 **cdbridge-agent_name** 的目錄中。此預設目錄建立於 **java.io.tmpdir** 內容值所定義的位置中。

註: 如果您執行 **fteCleanAgent** 指令，則會刪除此目錄中的所有檔案。

Windows -s (service_name)

選用項目（僅限 Windows）。指出代理程式是以 Windows 服務方式執行，該指令必須從 Windows 管理者使用者 ID 執行。如果未指定 **service_name**，則此服務會命名為 **mqmftAgentAGENTQMGR**，其中 **AGENT** 是代理程式名稱，而 **QMGR** 是代理程式佇列管理程式名稱。

服務的顯示名稱（顯示在 Windows 服務視窗中的名稱直欄內）一律為 **Managed File Transfer Agent AGENT@QMGR**。

註: 如果可轉散發的代理程式將以 Windows 服務形式來執行，則需要在系統環境中設定 **BFG_DATA** 環境變數，服務才能運作。

Windows -su (user_name)

選用項目（僅限 Windows）。代理程式以 Windows 服務方式執行時，此參數會指定服務執行時所使用的帳戶名稱。若要使用 Windows 網域使用者帳戶執行此代理程式，請以 **DomainName\UserName** 格式指定此值。若要使用本端內建網域中的帳戶執行服務，請以 **UserName** 格式指定值。

您使用 **-su** 參數指定的 Windows 使用者帳戶必須具有 **Log on as a service** 權限。如需如何授與此權限的相關資訊，請參閱 [以 Windows 服務方式執行 MFT 代理程式或日誌程式的指引](#)。

指定 **-s** 時需要。

Windows -sp (password)

選用項目（僅限 Windows）。

此參數僅在指定 **-s** 時有效。如果在指定 **-s** 參數時不指定此參數，會產生一則警告訊息。此訊息警告您，您必須先使用「Windows 服務」工具設定密碼，然後服務才能順利啟動。

Windows **-sj (options)**

選用項目（僅限 Windows）。以 Windows 服務方式啟動代理程式時，會以 **-D** 或 **-X** 的格式定義傳遞至 JVM 的選項清單。這些選項會使用 **#** 記號或分號 **(;)** 字元來區隔。如果您必須內含任何 **#** 或分號 **(;)** 字元，請將這些字元放在單引號內。

此參數僅在指定 **-s** 時有效。

Windows **-sl (options)**

選用項目（僅限 Windows）。設定 Windows 服務日誌層次。有效的選項是：error、info、warn、debug。預設值是 info。如果您在使用 Windows 服務時遇到問題，則此選項非常有用。將它設為 debug，可以在服務日誌檔中提供更詳細資訊。

此參數僅在指定 **-s** 時有效。

Windows **-n**

選用項目（僅限 Windows）。指示代理程式以正常程序執行。此參數與 **-s** 選項互斥。如果未指定其中一個 **-s** 參數及 **-n** 參數，則代理程式會配置為一般 Windows 處理程序。

-mquserid (userID)

選用項目。指定用來向指令佇列管理程式進行鑑別的使用者 ID。

-mqpassword (password)

選用項目。指定用來向指令佇列管理程式進行鑑別的密碼。您還必須指定 **-mquserid** 參數。如果指定了 **-mquserid**，卻未指定 **-mqpassword**，則系統會提示您提供相關聯的密碼。將不會顯示密碼。

-credentialsFile (filePath)

選用項目。現有或新認證檔的完整檔案路徑，IBM MQ 鑑別詳細資料會新增到該認證檔中。

此指令支援將 IBM MQ 鑑別詳細資料的集合新增至名為 Managed File Transfer 的認證檔。已啟用 IBM MQ 連線鑑別時，請使用此指令。如果更新現有的詳細資料，則必須使用 **-f** 強制參數。

-userid (username)

選用項目。用來與認證詳細資料相關聯的使用者 ID。如果未指定使用者 ID，則認證詳細資料將套用至所有使用者。您還必須指定 **-credentialsFile** 參數。

範例

在以下範例中，會透過代理程式佇列管理程式 QM_NEPTUNE，來建立新的 Connect:Direct 橋接器代理程式 CD_BRIDGE。此代理程式使用 Connect:Direct 節點 BRIDGE_NODE，將檔案傳送至其他 Connect:Direct 節點。BRIDGE_NODE 節點位於與此代理程式相同的系統上，並且使用預設埠進行用戶端連線。傳送至 Connect:Direct 或從其中傳送的檔案，暫時儲存在 /tmp/cd-bridge 目錄中。

```
fteCreateCDAgent -agentName CD_BRIDGE -agentQMgr QM_NEPTUNE
                 -cdNode BRIDGE_NODE -cdTmpDir /tmp/cd-bridge
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

fteCreate 日誌程式 (建立 MFT 檔案或資料庫日誌程式)

使用 **fteCreateLogger** 指令來建立 Managed File Transfer 檔案或資料庫日誌程式。

重要: **ULW** 在 IBM MQ for UNIX, Linux, and Windows 上，只有 IBM MQ 管理者（及 mqm 群組的成員）身分的使用者才能執行此指令。如果您嘗試以非 IBM MQ 管理者的使用者身分執行此指令，則會收到錯誤訊息 BFGCL0502E：您未獲授權執行所要求的作業。 ，且指令將不會執行。

z/OS 在 z/OS 系統上，使用者必須滿足（至少）下列其中一項條件才能執行移轉指令：

- 是 mqm 群組的成員（如果 mqm 群組已定義在系統上）。
- **V9.0.0.1** 是 BFG_GROUP_NAME 環境變數所指名之群組的成員（如果已指名群組的話）。
- **V9.0.0.1** 指令執行時，BFG_GROUP_NAME 環境變數中未設定任何值。

IBM i 上的日誌程式

IBM i

IBM i 平台不支援 Managed File Transfer 日誌程式。

用途

fteCreateLogger 指令提供一些 MQSC 指令，必須針對日誌程式指令佇列管理程式執行這些指令，才能建立下列日誌程式佇列：

- SYSTEM.FTE.LOG.COMD.*logger_name*
- SYSTEM.FTE.LOG.RJCT.*logger_name*

這些佇列是內部系統佇列，您不得從中修改、刪除或讀取訊息，除非您刪除日誌程式。下列位置的檔案中也提供了要執行的 MQSC 指令：

```
MQ_DATA_PATH\mqft\config\coordination_qmgr\loggers\logger_name\logger_name_create.mqsc
```

如果您稍後想要刪除日誌程式，請使用 **fteDeleteLogger** 指令。

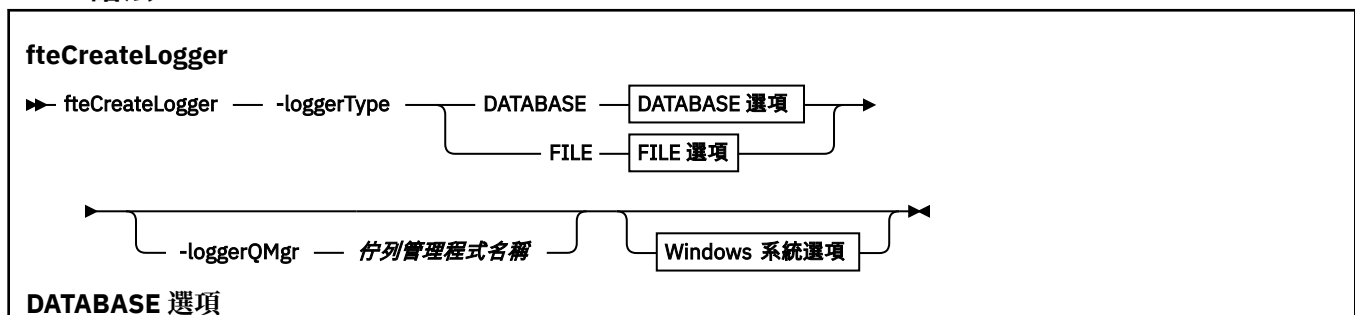
Managed File Transfer 提供進階日誌程式內容，協助您配置日誌程式。請參閱 [MFT 日誌程式配置內容](#)

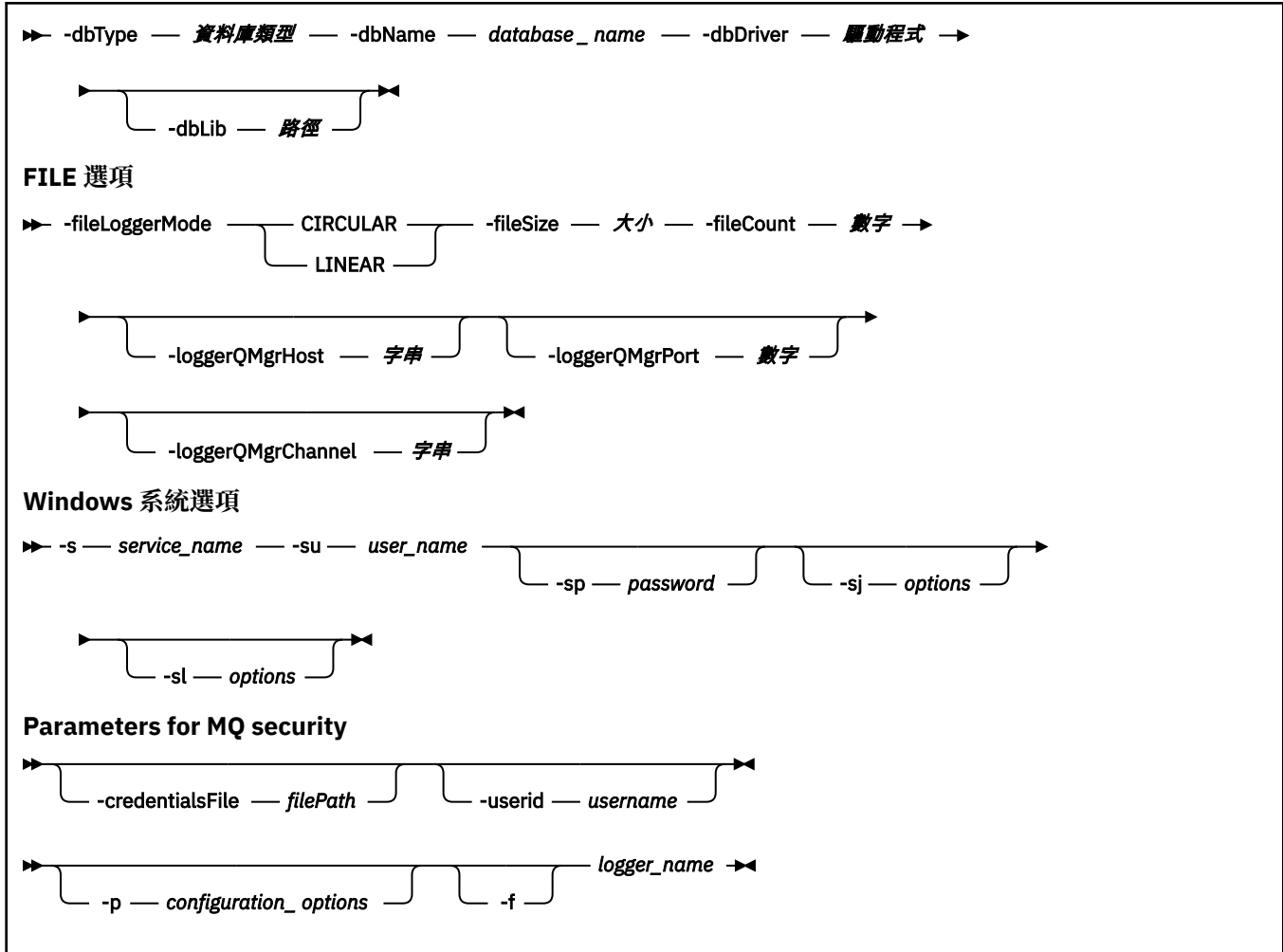
註: 如果要建立的日誌程式是資料庫日誌程式，而且並未連接至本端 Db2 資料庫時，您將需要手動建立 MQMFTCredentials.xml 檔案。該檔案包含用來連接資料庫的使用者名稱及密碼。您應在 logger.properties 檔中使用 wmqfte.database.credentials 內容檔，來指定 MQMFTCredentials.xml 檔的路徑。此認證檔的範例位於 MQ_INSTALLATION_PATH/mqft/samples/credentials/ 中。

特殊字元

如果使用的參數值包含特殊字元，請格外小心，以免指令 Shell 解譯字元的方式不符合預期。例如，如果完整檔案路徑和名稱包含諸如空格、引號（單引號或雙引號）、正斜線或反斜線字元等字元，則可能會被指令 Shell 解譯，而不是直接傳遞至指令本身。若要避免字元被指令 Shell 解譯，請用雙/單引號將整個參數括住，或者使用指令 Shell 的 ESC 序列來跳出特殊字元。

語法





參數

-loggerType (type)

必要項目。指定將記載 Managed File Transfer 資訊的位置。類型的選項為 DATABASE（如果傳送資訊將記載至資料庫）或 FILE（如果資訊將記載至檔案）。

-loggerQMgr (qmgr_name)

選用項目。決定所連接的佇列管理程式，以接收包含 Managed File Transfer 的相關資訊的訊息。佇列管理程式必須在與日誌程式相同的系統上。如果您沒有指定 **-loggerQMgr** 參數，則與為此日誌程式設定的配置選項相關聯的協調佇列管理程式將用作預設值。

-dbType (database_type)

-loggerType 為 DATABASE 時需要。指定用於儲存 Managed File Transfer 資訊的資料庫管理系統的類型。選項為 db2 或 oracle

註：您需要使用 SQL 檔案建立表格。 .sql 檔案可從 MQ_INSTALLATION_PATH_/mqft/sql 中取得：

- 若為 Db2 資料庫: ftelog_tables_db2.sql
- 若為 Oracle 資料庫: ftelog_tables_oracle.sql

-dbName (database_name)

-loggerType 為 DATABASE 時需要。儲存 Managed File Transfer 資訊的資料庫名稱。資料庫必須使用 Managed File Transfer 日誌表格進行配置。

-dbDriver (driver)

-loggerType 為 DATABASE 時需要。資料庫的 JDBC 驅動程式類別位置。這通常是 JAR 檔的路徑及檔名。

-dbLib (path)

選用項目，若 -loggerType 為 DATABASE。您選擇的資料庫所需的任何原生程式庫的位置。

-fileLoggerMode (mode)

-loggerType 為 FILE 時需要。指定用於儲存受管理檔案傳送資訊的檔案系統的類型。選項為 LINEAR 或 CIRCULAR。

選項 LINEAR 表示檔案日誌程式會將資訊寫入檔案，直至檔案達到 -fileSize 定義的大小上限。達到大小上限時，檔案日誌程式將會啟動新的檔案。先前寫入的檔案不會被刪除，從而容許這些檔案保留為日誌訊息的歷程記錄。檔案在此模式下執行時不會被刪除，因此 -fileCount 會被忽略，因為沒有可以建立的檔案數目上限。因為在此模式下執行時沒有上限，因此需要追蹤日誌檔使用的磁碟空間，以避免磁碟空間不足。

選項 CIRCULAR 表示檔案日誌程式會將資訊寫入檔案，直至檔案達到 -fileSize 定義的大小上限。達到大小上限時，檔案日誌程式將會啟動新的檔案。在此模式下寫入的檔案數目上限由使用 -fileCount 定義的值控制。達到此檔案數目上限時，檔案日誌程式即會刪除第一個檔案，然後將它重建作為目前作用中的檔案。如果 -fileSize 中定義的值是固定大小位元組單位，則此模式下使用的磁碟空間上限將等於 fileSize x fileCount。如果 -fileSize 中定義的值是時間單位，則大小上限將視您的系統在這些時段中的日誌訊息傳輸量而定。

如需相關資訊，請參閱 [MFT 日誌程式配置內容](#)

-fileSize (size)

-loggerType 為 FILE 時需要。容許日誌檔增長到的大小上限。值是一個大於零的正整數，後面接下列其中一個單位：KB、MB、GB、m (分鐘)、h (小時)、d (天)、w (週)。例如：-fileSize 5MB (指定大小上限 5MB)、-fileSize 2d (指定最長 2 天時間的資料)。

-fileCount (number)

當 -loggerType 為 FILE 且 -fileLoggerMode 為 CIRCULAR 時，這是必要項目。要建立的日誌檔數上限。當資料量超出此數目的檔案可以儲存的數量上限時，就會刪除最舊的檔案，以便日誌檔數永不超出此參數中指定的值。

-loggerQMGrHost

執行日誌程式佇列管理程式所處機器的主機名稱或 IP 位址。

預設值為 None。

如果未指定 -loggerQMGrHost 參數，則會以連結模式建立日誌程式。

-loggerQMGrPort

日誌程式佇列管理程式正在其中進行接聽的埠號。

預設值是 1414。

-loggerQMGrChannel

用於連接至日誌程式佇列管理程式的通道名稱。

預設值是 SYSTEM.DEF.SVRCONN。

Windows -s (service_name)

選用項目（僅限 Windows 系統）。指出日誌程式是以 Windows 服務方式執行。如果未指定 service_name，則此服務會命名為 mqmftLoggerLOGGERQMGR，其中 LOGGER 是日誌程式名稱，而 QMGR 是日誌程式佇列管理程式名稱。

服務的顯示名稱（顯示在 Windows 服務視窗中的名稱直欄內）一律為 **Managed File Transfer Logger
LOGGER@QMGR**。

Windows **-su (user_name)**

選用項目（僅限 Windows）。日誌程式以 Windows 服務方式執行時，此參數會指定服務執行時所使用的帳戶名稱。若要使用 Windows 網域使用者帳戶執行此日誌程式，請以 `DomainName\UserName` 格式指定此值。若要使用本端內建網域中的帳戶執行服務，請以 `UserName` 格式指定值。

您使用 **-su** 參數指定的 Windows 使用者帳戶必須具有 **Log on as a service** 權限。如需如何授與此權限的相關資訊，請參閱 [以 Windows 服務方式執行 MFT 代理程式或日誌程式的指引](#)。

指定 **-s** 時需要。

Windows **-sp (password)**

選用項目（僅限 Windows）。**-su** 參數所設定使用者帳戶的密碼。

此參數僅在指定 **-s** 時有效。如果在指定 **-s** 參數時不指定此參數，會產生一則警告訊息。此訊息警告您，您必須先使用「Windows 服務」工具設定密碼，然後服務才能順利啟動。

Windows **-sj (options)**

選用項目（僅限 Windows）。以 Windows 服務方式啟動日誌程式時，會以 **-D** 或 **-X** 的格式定義傳遞至 JVM 的選項清單。這些選項會使用 **#** 記號或分號 (**;**) 字元來區隔。如果您必須內含任何 **#** 或分號 (**;**) 字元，請將這些字元放在單引號內。

此參數僅在指定 **-s** 時有效。

Windows **-sl (options)**

選用項目（僅限 Windows）。設定 Windows 服務日誌層次。有效的選項是：`error`、`info`、`warn`、`debug`。預設值是 `info`。如果您在使用 Windows 服務時遇到問題，則此選項非常有用。將它設為 `debug`，可以在服務日誌檔中提供更詳細資訊。

此參數僅在指定 **-s** 時有效。

-p (configuration options)

選用項目。指定用於建立日誌程式的配置選項集。依照慣例，此值是協調佇列管理程式的名稱。如果未指定此參數，則會使用一組預設的配置選項。

-f

選用項目。強制指令改寫現有的配置。

(logger_name)

必要項目。要建立的日誌程式名稱。這會納入 Managed File Transfer 佇列名稱中，因此只能包含字母、數字及句點 (**.**) 和底線字元 (**_**)。它也限制為長度上限為 28 個字元。

-credentialsFile (filePath)

選用項目。現有或新認證檔的完整檔案路徑，IBM MQ 鑑別詳細資料會新增到該認證檔中。

此指令支援將 IBM MQ 鑑別詳細資料的集合新增至名為 Managed File Transfer 的認證檔。已啟用 IBM MQ 連線鑑別時，請使用此指令。如果更新現有的詳細資料，則必須使用 **-f** 強制參數。

-userid (username)

選用項目。用來與認證詳細資料相關聯的使用者 ID。如果未指定使用者 ID，則認證詳細資料將套用至所有使用者。您還必須指定 **-credentialsFile** 參數。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，會建立名為 `filelogger1` 的循環檔案日誌程式。檔案日誌程式會建立最多 10 個檔案，每一個檔案大小為 10MB，最多使用 100MB 的總磁碟空間。

```
fteCreateLogger -loggerType FILE -fileLoggerMode CIRCULAR -fileSize 10MB -fileCount 10 filelogger1
```

在此範例中，會建立稱為 `dblogger1` 的資料庫日誌程式。資料庫日誌程式連接至稱為 `FTEDB` 的 `Db2` 資料庫：

```
fteCreateLogger -loggerType DATABASE -dbName FTEDB -dbType DB2
                -dbDriver "C:\Program Files (x86)\IBM\SQLLIB\java\db2jcc4.jar" dblogger1
```

在此範例中，會建立稱為 `dblogger1` 的資料庫日誌程式。資料庫日誌程式連接至稱為 `FTEDB` 的 `Oracle` 資料庫：

```
fteCreateLogger -loggerType DATABASE -dbName FTEDB -dbType oracle
                -dbDriver "C:\app\oracle\product\12.1.0\dbhome_2\jdbc\lib\ojdbc7.jar" dblogger1
```

在此範例中，會使用主機名稱及預設埠與通道來建立用戶端模式檔案日誌程式：

```
fteCreateLogger -loggerType FILE -loggerQMgr CORDQM -loggerQMgrHost cordqm.ibm.com
                -fileLoggerMode CIRCULAR -fileSize 10MB -fileCount 10 FL1
```

在此範例中，會使用主機名稱、埠及通道來建立用戶端模式檔案日誌程式：

```
fteCreateLogger -loggerType FILE -loggerQMgr CORDQM -loggerQMgrHost cordqm.ibm.com
                -loggerQMgrPort 4444 -loggerQMgrChannel LOGGER_CHANNEL -fileLoggerMode CIRCULAR -fileSize 10MB
                -fileCount 10 FL1
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

fteCreateMonitor: 建立 MFT 資源監視器

fteCreateMonitor 指令可從指令行建立並啟動新的資源監視器。您可以使用 `Managed File Transfer` 來監視資源（例如目錄的內容），以在符合觸發條件時，啟動指定的作業（例如檔案傳送）。

用途

使用 **fteCreateMonitor** 指令，可透過 `Managed File Transfer` 代理程式建立並啟動新的資源監視器。例如，您可以按下列方式使用資源監視器：外部應用程式會將一個以上檔案放在已知目錄中，並在程序完成後，外部應用程式會將觸發檔案放在監視目錄中。然後，系統會偵測到觸發檔案，而且啟動定義的檔案傳送並將檔案從已知目錄複製到目的地代理程式。

您可以使用 **-ox** 及 **-ix** 參數，將資源監視器配置匯出到 XML 檔以及匯入 XML 檔。使用 **fteCreateMonitor** 指令匯入此檔案，可透過與 **fteCreateMonitor** 指令中提供的資源監視器相同的參數建立新的資源監視器，來匯出至 XML 檔。此外，您也可以使用 **-f** 及 **-c** 參數來動態地改寫監視器配置。

註：可以在代理程式上建立且全部以相同優先順序執行的資源監視器數目沒有限制。請考慮重疊的受監視資源、衝突的觸發條件及資源輪詢頻率的含意。如需相關資訊，請參閱 [MFT 資源監視概念](#)。

通訊協定橋接器代理程式不支援 **fteCreateMonitor** 指令。

提示：您也可以使用 **fteListMonitors** 指令，將資源監視器配置匯出至 XML 檔：

- 搭配使用 **fteListMonitors** 指令與 **-ox** 會匯出單一資源監視器的定義。
- **V 9.0.5** 從 IBM MQ 9.0.5 開始，搭配使用 **fteListMonitor** 指令與 **-od** 可將多個資源監視器定義匯出至指定的目錄。您也可以使用 **-od** 選項，將單一資源監視器定義匯出至指定的目錄。

如需 **fteListMonitors** 指令的相關資訊，請參閱第 401 頁的『[fteListMonitors: 列出 MFT 資源監視器](#)』。

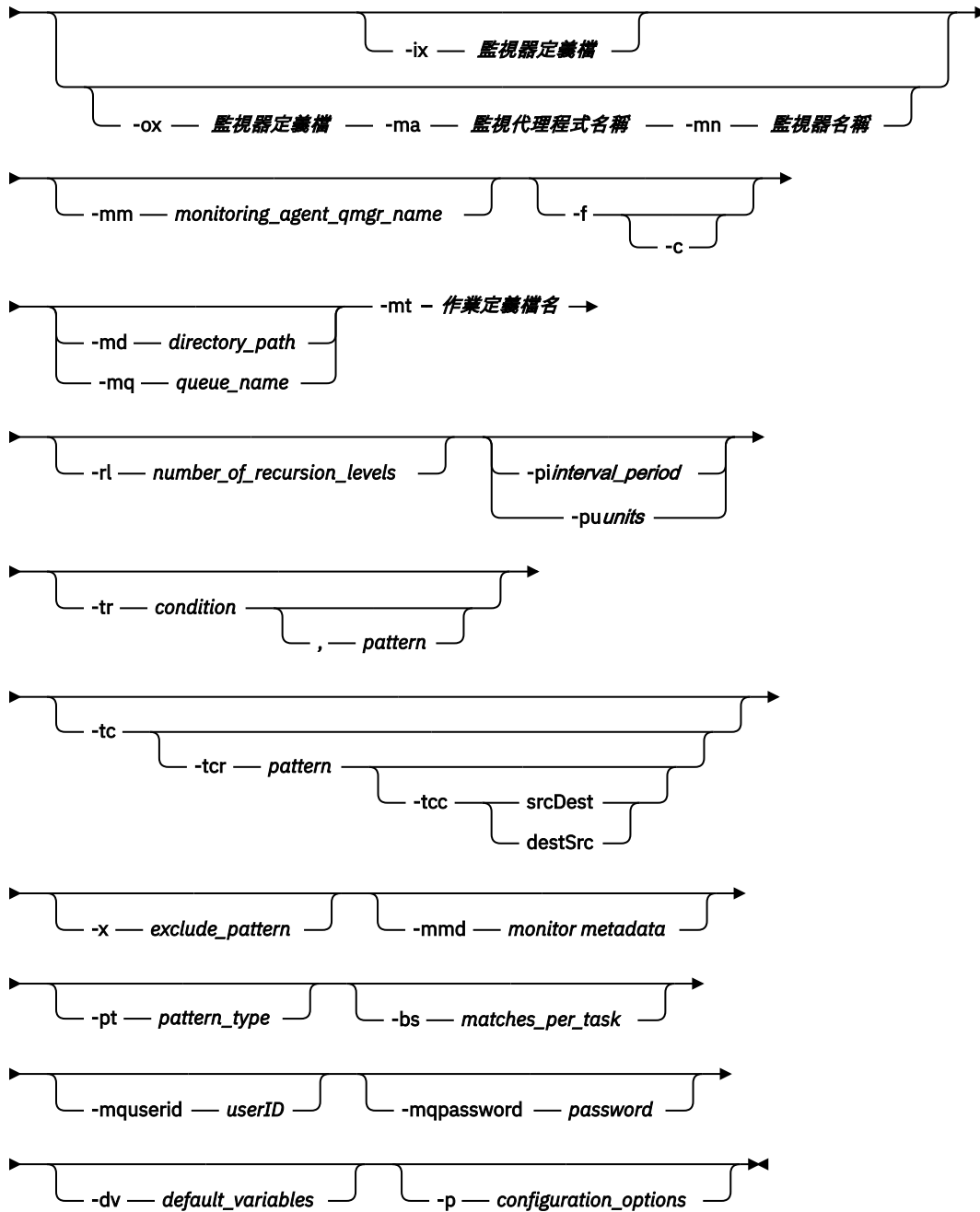
特殊字元

如果使用的參數值包含特殊字元，請格外小心，以免指令 Shell 解譯字元的方式不符合預期。例如，如果完整檔案路徑和名稱包含諸如空格、引號（單引號或雙引號）、正斜線或反斜線字元等字元，則可能會被指令 Shell 解譯，而不是直接傳遞至指令本身。若要避免字元被指令 Shell 解譯，請用雙/單引號將整個參數括住，或者使用指令 Shell 的 ESC 序列來跳出特殊字元。

語法

fteCreateMonitor

► fteCreateMonitor ►



參數

-ix (xml_filename)

選用項目。從 XML 檔匯入資源監視器配置。

-ox (xml_filename)

選用項目。此參數必須與 **-ma** 及 **-mn** 參數一起指定。將資源監視器配置匯出至 XML 檔。

-mn (monitor_name)

必要項目。指派給此監視器的名稱。監視器名稱對於監視代理程式必須是唯一的。但是，您可以刪除監視器，然後以相同名稱建立監視器。

資源監視器名稱的長度上限為 256 個字元。資源監視器名稱不區分大小寫。以小寫或大小寫混合格式輸入的資源監視器名稱，都會轉換成大寫。資源監視器名稱不得包含星號 (*)、百分比 (%) 或問號 (?) 字元。

-ma (monitoring_agent_name)

必要項目。要執行資源監視的代理程式名稱。此監視代理程式還必須是您要觸發的監視作業的來源代理程式。

-mm (monitoring_agent_qmgr_name)

監視代理程式所連接的佇列管理程式的名稱。因為監視代理程式與來源代理程式必須相同，所以此佇列管理程式也是您的來源代理程式的佇列管理程式。

註: `fteCreateMonitor` 指令會連接至 Managed File Transfer 拓撲的指令佇列管理程式。如果此指令佇列管理程式也是監視代理程式的代理程式佇列管理程式，則此參數是選用項目。否則，此參數是必要項目。

-f

選用項目。使用此參數來改寫資源監視器配置。例如，資源監視器代理程式上已有您選擇的資源監視器名稱，而您只想要更新，而不要刪除並重建同名的監視器。使用此參數會使代理程式重新啟動監視器處理程序。

-c

選用項目。此參數會清除所更新資源監視器的歷程，從而導致資源監視器再次檢查觸發條件。您只能將此參數與 **-f** 參數搭配使用。

-md (directory_path)

選用項目。您想要監視的目錄路徑的絕對名稱。除非您使用 **-ix** 或 **-ox** 參數，否則您必須指定 **-md** 或 **-mq** 參數之一。

-mq (queue_name)

選用項目。要監視的佇列的名稱。此佇列必須位於監視代理程式的佇列管理程式上。除非您使用 **-ix** 或 **-ox** 參數，否則您必須指定 **-md** 或 **-mq** 參數之一。

-mt (task_definition_file_name)

必要項目。包含作業定義的 XML 文件的名稱，在滿足觸發條件時您要執行作此業定義。如需相關資訊，請參閱 [第 133 頁的『使用傳送定義檔』](#)。傳送定義 XML 文件的路徑必須在您從中執行 `fteCreateMonitor` 指令的本端檔案系統上。如果您沒有指定檔案的路徑，則此指令會在現行工作目錄中尋找該檔案。除非您使用 **-ix** 或 **-ox** 參數，否則 **-mt** 是必要的參數。

您可以在 `fteCreateTransfer` 指令上使用 **-gt** 參數，以產生包含檔案傳送要求的範本 XML 文件。監視器使用傳送範本作為其作業定義。

V 9.0.1 執行 `fteCreateMonitor` 指令時，您也可以使用傳送回復逾時 **-rt** 參數來搭配 **-gt** 參數。您可以設定時間量（以秒為單位），在此期間內，來源代理程式會一直重試回復已停止的傳送。然後，回復逾時參數將包含在具有監視器所使用之傳送定義的 XML 文件中。如需如何設定此參數的相關資訊，請參閱 [fteCreateTransfer](#) 指令。

z/OS 在 z/OS 上，您必須將作業定義文件儲存在「z/OS UNIX 系統服務」的 UNIX 檔案中。您不能將作業定義文件儲存在 z/OS 循序檔或 PDS 成員中。

IBM i 在 IBM i 上，您必須將作業定義文件儲存在整合檔案系統中。

-rl (number_of_recursion_levels)

選用項目。根監視目錄的監視遞迴層次，也就是將進入的子目錄層次數。例如，在諸如下列範例的目錄結構中，C:\wmqfte\monitor 設為根監視目錄：

```
C:\wmqfte\monitor
C:\wmqfte\monitor\reports
C:\wmqfte\monitor\reports\2009
C:\wmqfte\monitor\reports\2009\April
```

如果您指定 **-rl 2**，則 Managed File Transfer 只會一直向下搜尋

C:\wmqfte\monitor\reports\2009 目錄及其同層級目錄。

C:\wmqfte\monitor\reports\2009\April 目錄會被忽略。依預設，遞迴設定為無。

-pi (interval_period)

選用項目。目錄的各個監視器之間的時間間隔。輪詢間隔必須是正整數值。**-pi** 的預設值為 1。

-pu (units)

選用項目。監視器輪詢間隔的時間單位。如果您指定 **-pu** 參數，您還必須指定 **-pi** 參數。**-pu** 的預設值為 minutes。請指定下列其中一個選項：

秒

分鐘

時

日

-tr

選用項目。指定進行定義的作業必須滿足的觸發條件。如果不滿足條件，根據來源代理程式，不會啟動監視作業（例如，檔案傳送）。觸發條件由兩個選用部分（條件及型樣）組成，以逗點區隔。請指定下列其中一種格式：

- `condition,pattern`

其中 *condition* 是下列其中一個值：

match

對於每一個滿足的觸發程式，會執行定義的作業。match 是預設值。

比方說，例如相符項為 *.go，且呈現檔案 LONDON.go 及 MANCHESTER.go，則會對 LONDON.go 執行此作業，而對 MANCHESTER.go 執行其他作業。

如果呈現前一個輪詢的相同觸發程式檔案（亦即，檔案尚未修改），則此檔案尚未滿足觸發條件。亦即，相符觸發檔案必須是新的，且必須在執行定義的作業之前自前次輪詢以來已修改。

noMatch

監視目錄中沒有任何檔案與型樣相符。亦即，如果監視目錄中的任何檔案不存在，則滿足此條件。如果在建立監視器時沒有任何檔案與觸發條件相符，監視器會立即啟動，但在找到相符檔案及移除之前，不會再次啟動。

noSizeChange=n

目錄中至少一個檔案與型樣相符，且 *n* 輪詢間隔的檔案大小未變更。*n* 的值是正整數。

fileSize>=size

目錄中至少一個檔案與型樣相符，且檔案大小下限大於或等於 *size*。值 *size* 是整合與 B、KB、MB 或 GB 的選用大小單位的組合。例如，fileSize">="10KB。如果您沒有指定大小單位，則使用的預設大小為位元組。在所有作業系統上，當您在指令行指定 fileSize 選項時，必須以雙引號括住大於符號 (>)，如這個範例所示。

型樣是以萬用字元或 Java 正規表示式格式表示的檔案型樣比對序列。型樣的預設值為 *，即與任何檔案相符，預設格式是萬用字元格式。使用 **-pt** 指定型樣的格式。

例如，在字尾為 `.go` 的檔案存在於監視目錄中時，則滿足下列觸發條件。

```
-tr match,*.go
```

在監視目錄中沒有任何字尾為 `.stop` 的檔案時，滿足下列觸發條件。

```
-tr noMatch,*.stop
```

只有在同時指定 `-md` 參數時，才能指定 `condition,pattern` 。

• `condition`

其中 `condition` 是下列其中一個值：

queueNotEmpty

監視佇列不是空的。亦即，如果受監視佇列上存在任何 IBM MQ 訊息，則符合此條件。針對佇列上的所有訊息執行單一作業。

completeGroups

監視佇列上具有完整群組。亦即，如果受監視佇列上的任何 IBM MQ 訊息群組是完整的，則符合此條件。針對佇列上的每一個完整群組執行個別作業。

如果不在群組中的單一訊息放在佇列上，它會視為如同它是完整群組，並針對單一訊息執行作業。

只有在同時指定 `-mq` 參數時，才能指定 `condition` 。

對於您建立的每一個監視器，您只能指定 `-tr` 參數一次。

-tc

選用項目。指出觸發的檔案包含一個以上檔案路徑，以產生傳送要求。觸發檔案內容的預設格式，是一行顯示一個檔案項目。請將檔案路徑指定為 `source file path` 或 `source file path,destination file path`。此參數僅適用於觸發 `match` 及 `noSizeChange` 的目錄監視器。

-tcr (pattern)

選用項目。指定取代正規表示式，以剖析觸發檔案。如果您指定 `-tcr` 參數，您還必須指定 `-tc` 參數。設計可透過一或兩個擷取群組完整剖析每一個行項目的型樣。群組一定義來源檔案路徑，選用的群組二定義目的地檔案路徑。此為預設行為，您可以使用 `-tcc` 參數予以變更。

如需相關資訊與範例，請參閱第 157 頁的『使用觸發檔』。

-tcc

選用項目。定義正規表示式擷取群組順序。

srcDest

此為預設值：群組一是來源檔案路徑，群組二是目的地檔案路徑。

destSrc

`srcDest` 的反向設定。群組一是目的地檔案路徑，群組二是來源檔案路徑。請確定 `destSrc` 的正規表示式有兩個擷取群組。

如果您指定 `-tcc` 參數，您還必須指定 `-tcr` 參數。

-x (exclude_pattern)

選用項目。指定從觸發程式型樣相符排除的檔案。觸發程式型樣由 `-tr` 參數指定。

型樣是以萬用字元或 Java 正規表示式格式表示的檔案型樣比對序列。預設格式是萬用字元格式。使用 `-pt` 參數指定型樣的格式。

-mmd (monitor metadata)

選用項目。指定傳至監視器結束點的使用者定義 meta 資料。此參數可以採用一個以上以逗點區隔的名稱配對。每一個名稱配對各包含一個 `name=value`。您可以在一個指令中多次使用 `-mmd` 參數。

-pt (pattern_type)

選用項目。`-tr` 及 `-x` 參數使用的型樣類型。有效值為：

wildcard

型樣如同萬用字元型樣一樣求值。星號 (*) 會符合零個以上字元，而問號 (?) 則明確符合一個字元。這是預設值。

正規表示式

型樣的求值方式與 Java 正規表示式相同。如需相關資訊，請參閱第 554 頁的『MFT 所使用的正規表示式』。

-bs (*matches_per_task*)

選用項目。單一作業中包含的觸發程式相符數上限。比方說，例如為 *matches_per_task* 指定值 5，且單一輪詢間隔中出現九個觸發程式相符項，則會執行兩項作業。第一項作業對應於觸發程式 1-5（內含），第二項作業對應於觸發程式 6-9。*matches_per_task* 的預設值為 1。

只有當您提供給 **-mt** 參數的作業定義 XML 是 managedTransfer 時，才支援 **-bs** 參數。**-bs** 參數不支援 managedCall。

-mquserid (*userID*)

選用項目。指定用來向指令佇列管理程式進行鑑別的使用者 ID。

-mqpassword (*password*)

選用項目。指定用來向指令佇列管理程式進行鑑別的密碼。您還必須指定 **-mquserid** 參數。如果指定了 **-mquserid**，卻未指定 **-mqpassword**，則系統會提示您提供相關聯的密碼。不會顯示密碼。

-dv (*default_variables*)

選用項目。在監視佇列時，可以在變數替代中使用的預設變數清單（以逗點區隔）。值的格式是鍵值組。例如：

```
-dv size=medium,color=blue
```

如需變數替代的相關資訊，請參閱第 149 頁的『使用變數替代來自訂 MFT 資源監視器作業』。如果您還指定了 **-mq** 參數，則只能指定 **-dv** 參數。

-? 或 -h

選用項目。顯示指令語法。

-p (*configuration_options*)

選用項目。此參數會決定用於取消傳送的配置選項集。依照慣例，請使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，此指令會使用與此非預設協調佇列管理程式相關聯的內容檔集。

如果未指定此參數，則會使用一組以預設協調佇列管理程式為基礎的配置選項。

範例

在此範例中，使用監視代理程式 MYAGENT 建立稱為 MYMONITOR 的新資源監視器。如果觸發條件是大於 5MB 的檔案呈現在目錄 C:\wmqfte\monitors 中，則會啟動檔案 C:\templates\transfer_reports.xml 中定義的檔案傳送。MYAGENT 也是 C:\templates\transfer_reports.xml 中所定義檔案傳送的來源代理程式：

```
fteCreateMonitor -ma MYAGENT -md C:\wmqfte\monitors -mn MYMONITOR -mt C:\templates\transfer_reports.xml -tr fileSize">"=5MB,*go
```

在此範例中，會建立使用代理程式 AGENT1 且稱為 MONITOR1 的資源監視器，以傳送大於 5 MB 的檔案，並匯出至 XML 檔案 monitor.xml。

```
fteCreateMonitor -ox monitor.xml -ma AGENT1 -mn MONITOR1 -mt task.xml -tr "fileSize>=5MB,*zip"
```

然後，XML 檔案會匯入，並變更為排除大於 10MB 的任何檔案。

```
fteCreateMonitor -ix monitor.xml -x "fileSize>=10MB,*zip" -f
```

在此範例中，會建立使用代理程式 MYAGENT 且稱為 MYMONITOR 的新資源監視器。

```
fteCreateMonitor -ma MYAGENT -md c:\wmqfte -mn MYMONITOR -mt c:\templates\transfer_reports.xml -tr "fileSize>=5MB,*go"
```

但是，觸發程式起初未正確設定為監視 `c:\wmqfte`，而非 `c:\wmqfte\monitors`。

fteCreateMonitor 要求會立即重新發出，並更正監視器目錄，以及用來更新監視器的 **-f** (改寫) 和 **-c** (清除歷程) 參數。

```
fteCreateMonitor -ma MYAGENT -md c:\wmqfte\monitors -mn MYMONITOR -mt c:\templates\transfer_reports.xml  
-tr "fileSize>=5MB,*.go" -f -c
```

回覆碼

回覆碼	說明
0	指令已順利完成。
1	指令未順利結束。

fteCreateTemplate: 建立新的檔案傳送範本

fteCreateTemplate 指令可建立檔案傳送範本，以備將來使用。唯一必要的參數是 **-tn** (*template_name*) 參數。其他所有參數是選用的，雖然如果指定來源檔案規格，您還必須提供目的地檔案。同樣地，如果指定目的地檔案，您還必須提供來源檔案規格。

用途

使用 **fteCreateTemplate** 指令可建立檔案傳送範本來儲存您的傳送詳細資料，以備日後使用。使用傳送範本可儲存重複或複式傳送的一般檔案傳送設定。在建立傳送範本之後，請使用「IBM MQ Explorer」提交範本。您無法從指令行提交傳送範本。

您使用 **fteCreateTemplate** 指令建立的傳送範本與您在 **fteCreateTransfer** 指令上使用 **-gt** 參數建立的 XML 訊息不同。您不能交替地使用兩種不同類型的範本。

您可以從可以連接至 IBM MQ 網路的任何系統執行 **fteCreateTemplate** 指令，然後遞送至協調佇列管理程式。具體而言，為了讓指令能夠執行，您必須在此系統上安裝 Managed File Transfer，且必須已在此系統上配置 Managed File Transfer 元件，才能與 IBM MQ 網路進行通訊。

此指令使用 `command.properties` 檔案來連接至 Managed File Transfer 拓撲的指令佇列管理程式。如果 `command.properties` 檔案包含 **connectionQMGrHost** 內容，則指令會使用 CLIENT 傳輸連接至指令佇列管理程式。否則，指令會使用 BINDINGS 傳輸來連接至指令佇列管理程式。如果 `command.properties` 檔案不存在，指令會失敗並產生下列錯誤：


```
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"
```

如需相關資訊，請參閱 `command.properties` 檔。


您可以為檔案傳送指定多個來源檔案，但只能指定一個目的地代理程式；不支援將一個檔案傳送至多個目的地代理程式。但是，您可以在單一目的地代理程式上，將多個來源檔案傳送至多個目的地檔案。

如需如何傳送檔案的相關指引，請參閱第 530 頁的『傳送檔案的準則』。

特殊字元

請謹慎使用包含特殊字元的參數，避免指令 Shell 以超乎預期的方式來解譯字元。  例如，包含單引號的完整資料集名稱及包含星號字元的來源規格可能會由指令 Shell 解譯，而不是透過傳送要求傳遞。若要避免由指令 Shell 解譯字元，請以雙引號括住整個參數，如第 367 頁的『範例』中的最後兩個範例所示，或使用指令 Shell 的 ESC 序列來跳出特殊字元。

相對路徑

fteCreateTemplate 指令支援使用相對檔案路徑。在分散式系統  及「z/OS UNIX 系統服務」上，依預設，會將路徑視為相對於執行代理程式的使用者的起始目錄。若要變更路徑名稱相對的目錄，

請在 `agent.properties` 檔案中設定 `transferRoot` 內容。此檔案位於 `MQ_DATA_PATH/mqft/config/coordination_qmgr/agents/agent_name` 目錄中。將下面這一行加到檔案中：

```
transferRoot=directory_name
```

您必須跳出 Windows 路徑，或以 UNIX 格式撰寫這些路徑。例如，將 `C:\TransferRoot` 指定為 `C:\TransferRoot` 或 `C:/TransferRoot`。

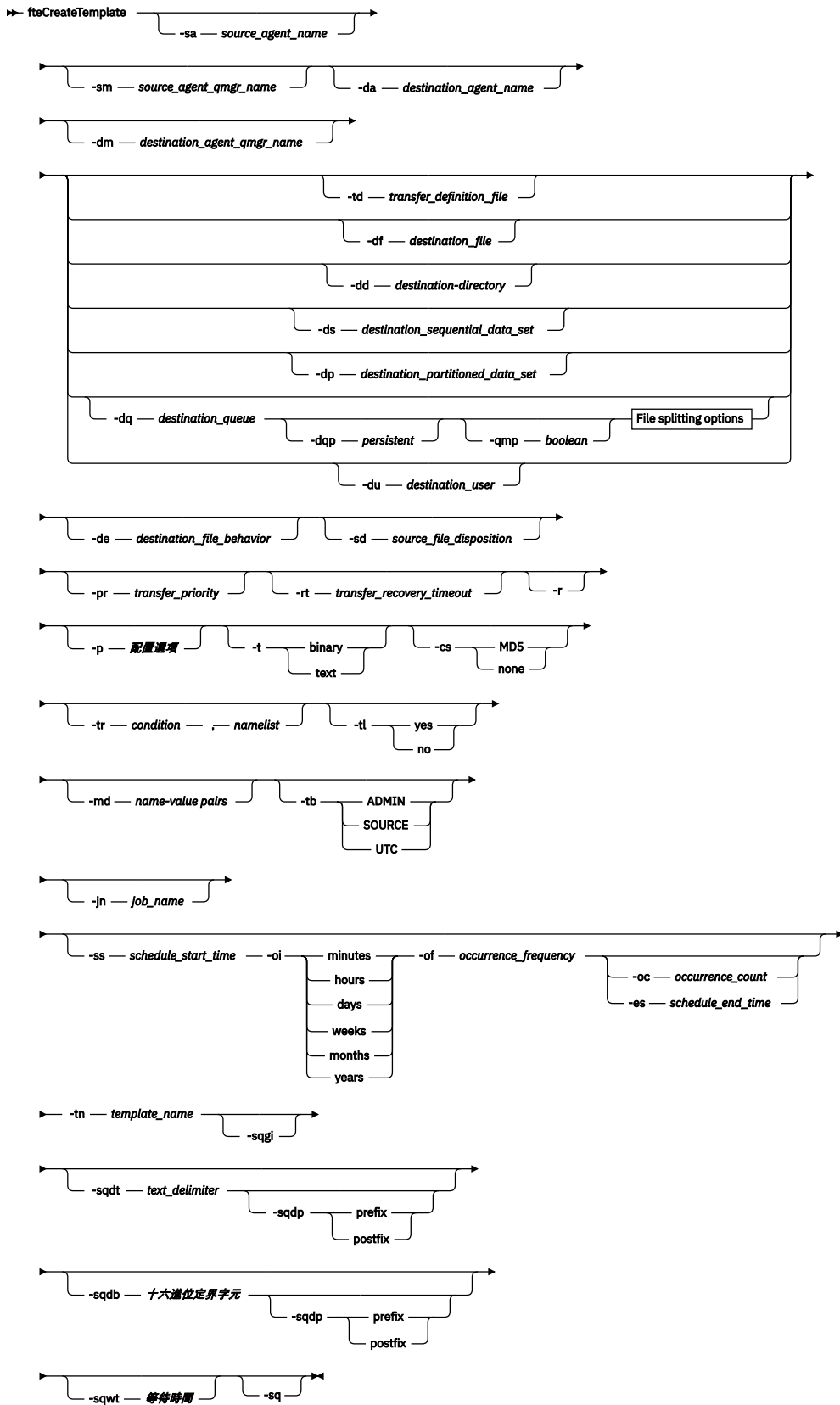
z/OS 在 z/OS 上，依預設，目前正在執行代理程式的使用者名稱會作為高階限定元字首新增至不完整的資料集規格。例如：`//ABC.DEF`。若要變更加在資料集名稱上作為字首的值，請在 `agent.properties` 檔案中設定 `transferRootHLQ` 內容。此檔案位於 `MQ_DATA_PATH/mqft/config/coordination_qmgr/agents/agent_name` 目錄中。將下面這一行加到檔案中：

```
transferRootHLQ=prepend_value
```

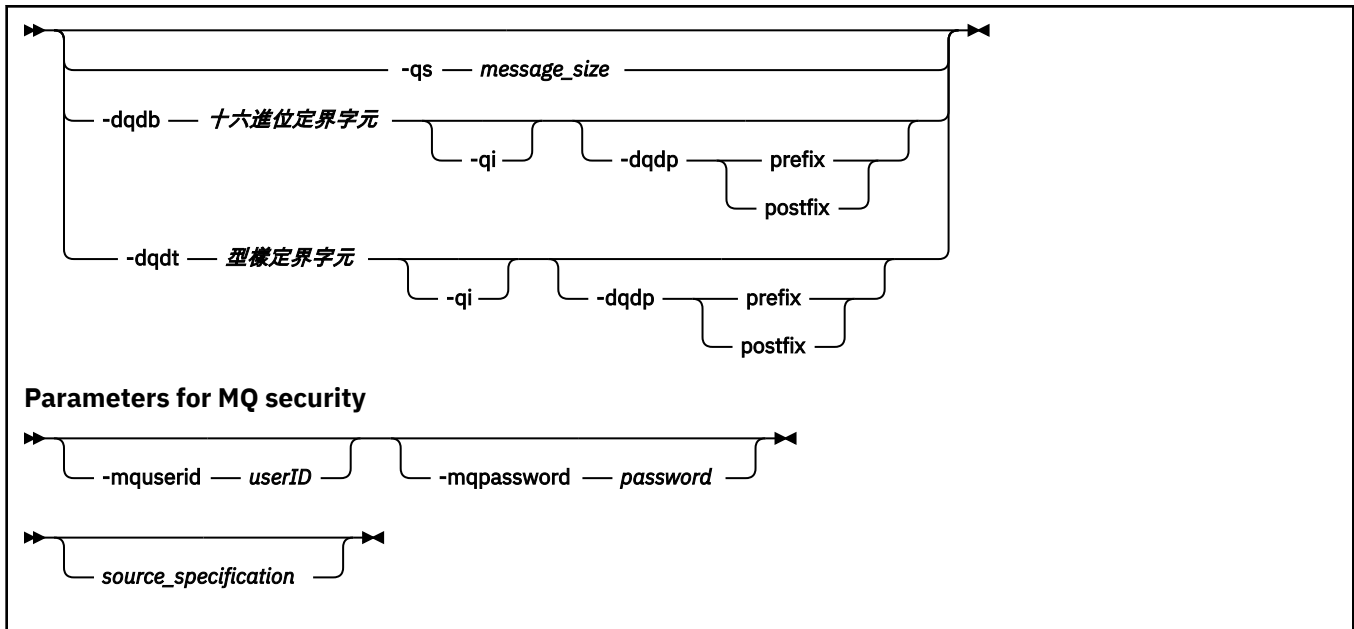
z/OS 不過，如果傳送涉及 z/OS 系統上的 `Connect:Direct` 節點，則資料集規格將解譯成完整名稱。資料集名稱不會加上高階限定元。

語法

fteCreateTemplate



File splitting options



參數

-sa source_agent_name

選用項目。從中傳送來源檔案的代理程式名稱。如果您在建立範本時不指定此代理程式名稱，則必須在使用範本時指定來源代理程式名稱。

-sm source_agent_qmgr_name

選用項目。來源代理程式所連接的佇列管理程式的名稱。

如果未指定 **-sm** 參數，則根據來源代理程式名稱，由使用中的配置選項集來決定使用的佇列管理程式。如果使用這些選項無法決定佇列管理程式名稱，傳送範本建立會失敗。例如，如果找不到來源代理程式的 `agent.properties` 檔案，則範本建立會失敗。

-da destination_agent_name

選用項目。檔案傳送至其中的代理程式名稱。如果您在建立範本時不指定目的地代理程式名稱，則必須在使用範本時指定目的地代理程式名稱。

-dm destination_agent_qmgr_name

選用項目。目的地代理程式所連接的佇列管理程式名稱。

如果未指定 **-dm** 參數，則根據目的地代理程式名稱，由使用中的配置選項集來決定使用的佇列管理程式。如果使用這些選項無法決定佇列管理程式名稱，傳送範本建立會失敗。例如，如果找不到目的地代理程式的 `agent.properties` 檔案，則範本建立會失敗。

-td transfer_definition_file

選用項目。XML 文件的名稱，定義傳送的一個以上來源及目的地檔案規格。

需要 **-td**、**-df**、**-dd**、**-ds**、**-dq**、**-du** 及 **-dp** 參數之一。如果您指定 **-td** 參數，則無法指定來源檔案，或者無法指定 **-df**、**-dd**、**-ds**、**-dp**、**-dq**、**-du**、**-sd**、**-r**、**-de**、**-t** 或 **-cs** 參數。

fteCreateTemplate 指令可尋找與您的現行目錄相關的傳送定義檔。如果您無法使用相對路徑表示法來指定傳送定義檔的位置，請改用傳送定義檔的完整路徑及檔名。


z/OS 在 z/OS 上，您必須將傳送定義檔儲存在「z/OS UNIX 系統服務」上的 UNIX 檔案中。您不能將傳送定義檔儲存在 z/OS 循序檔或 PDS 成員中。

IBM i 在 IBM i 上，您必須將傳送定義檔儲存在整合檔案系統中。

如需相關資訊，請參閱使用傳送定義檔。

-df destination_file

選用項目。目的地檔案的名稱。指定目的地代理程式執行所在的系統上有效的檔名。

如果目的地代理程式是 Connect:Direct 橋接器代理程式，則會以 `connect_direct_node_name:file_path` 格式來指定目的地檔案。Connect:Direct 橋接器代理程式僅接受以此格式指定的檔案路徑。  如果目的地代理程式是 Connect:Direct 橋接器代理程式，且目的地是 PDS 成員，則您必須同時指定 **-de** 參數，值為 `overwrite`。

需要 **-td**、**-df**、**-dd**、**-ds**、**-dq**、**-du** 及 **-dp** 參數之一。如果您指定 **-df** 參數，則無法指定 **-td**、**-dd**、**-dp**、**-dq**、**-du** 或 **-ds** 參數，因為這些參數是互斥的。

-dd destination_directory

選用項目。傳送檔案的目的地目錄名稱。指定目的地代理程式執行所在的系統上有效的目錄名稱。

如果目的地代理程式是 Connect:Direct 橋接器代理程式，則會以 `connect_direct_node_name:directory_path` 格式指定目的地目錄。如果目的地代理程式是 Connect:Direct 橋接器代理程式，且目的地是 PDS，則您必須同時指定 **-de** 參數，值為 `overwrite`。

需要 **-td**、**-df**、**-dd**、**-ds**、**-dq**、**-du** 及 **-dp** 參數之一。如果您指定 **-dd** 參數，則無法指定 **-td**、**-df**、**-dp**、**-dq**、**-du** 或 **-ds** 參數，因為這些參數是互斥的。

-ds destination_sequential_data_set

僅限 z/OS。選用項目。檔案傳送至其中的循序資料集名稱或 PDS 成員。指定循序資料集名稱或分割的資料集成員。

需要 **-td**、**-df**、**-dd**、**-ds**、**-dq**、**-du** 及 **-dp** 參數之一。如果您指定 **-ds** 參數，則無法指定 **-td**、**-dd**、**-df**、**-dq**、**-du** 或 **-dp** 參數，因為這些參數是互斥的。

資料集名稱的語法如下所示：

```
//data_set_name{;attribute;...;attribute}
```

or

```
//pds_data_set_name(member_name){;attribute;...;attribute}
```


亦即，字首為 // 的資料集名稱指定元，選擇性地後接許多屬性（以分號區隔）。

如果資料集位於 Connect:Direct 節點，則資料集名稱字首必須是節點名稱。例如：

```
CD_NODE1:// 'OBJECT.LIB';RECFM(F,B);BLKSIZE(800);LRECL(80)
```

如果目的地代理程式是 Connect:Direct 橋接器代理程式，且目的地是 PDS 成員，則您必須同時指定 **-de** 參數，值為 `overwrite`。如需往返於 Connect:Direct 節點的資料集傳送的相關資訊，請參閱第 534 頁的『從 Connect:Direct 節點來回傳送資料集』。

若為僅涉及 Managed File Transfer 代理程式的傳送，如果以單引號字元括住資料集名稱部分，則指定完整的資料集名稱。如果未以單引號字元括住資料集名稱，系統會加上目的地代理程式的預設高階限定元（`transferRootHLQ` 代理程式內容的值，或者，如果未設定 `transferRootHLQ`，則是執行代理程式的使用者 ID）。

註：  不過，如果傳送涉及 z/OS 系統上的 Connect:Direct 節點，則資料集規格將解譯成完整名稱。資料集名稱不會加上高階限定元。即使以單引號字元括住資料集名稱，作法還是一樣。

資料集屬性用於建立資料集，或確保現有的資料集是相容的。資料集屬性的規格採用適合 BPXWDYN 的格式（如需相關資訊，請參閱要求動態配置）。在代理程式將建立目的地資料集時，系統會自動指定下列 BPXWDYN 屬性：`DSN(data_set_name) NEW CATALOG MSG(numeric_file_descriptor)`，其中 `numeric_file_descriptor` 是 Managed File Transfer 產生的檔案描述子。對於資料集至資料集傳送，則為新目的地資料集從來源選取 RECFM、LRECL 及 BLKSIZE 的屬性。請注意，Managed File Transfer 不會設定新的目的地資料集的 SPACE 設定，且使用系統預設值。因此，建立新的資料集時，建議您指定 SPACE 屬性。您可以在 `agent.properties` 檔中使用 `bpxwdynAllocAdditionalProperties` 內

容，來設定套用至所有傳送的 BPXWDYN 選項。如需相關資訊，請參閱第 463 頁的『MFT agent.properties 檔案』。

使用 **fteCreateTemplate** 指令、**fteCreateTransfer** 指令或 agent.properties 檔中的 **bpxwdynAllocAdditionalOptions** 內容時，不得指定某些 BPXWDYN 選項。如需這些內容的清單，請參閱第 540 頁的『不得與 MFT 一起使用的 BPXWDYN 內容』。

當您將檔案或資料集傳送至磁帶時，任何已在磁帶上的現有資料集都會被取代。新資料集的屬性將根據傳送定義中所傳遞的屬性進行設定。如果您沒有指定任何屬性，會將屬性設定為與來源資料集相同，或者，在來源為檔案時設定為預設值。現有磁帶資料集的屬性會被忽略。

當目的地代理程式是通訊協定橋接器代理程式時，不支援 **-ds** 參數。

z/OS **-dp destination_partitioned_data_set**

僅限 z/OS。選用項目。檔案傳送至其中的目的地 PDS 名稱。指定分割的資料集名稱。如果因為傳送而建立 PDS，此 PDS 依預設會建立為 PDSE。您可以指定 DSNTYPE=PDS 來置換預設值。

需要 **-td**、**-df**、**-dd**、**-ds**、**-dq**、**-du** 及 **-dp** 參數之一。如果您指定 **-dp** 參數，則無法指定 **-td**、**-dd**、**-df**、**-dq**、**-du** 或 **-ds** 參數，因為這些參數是互斥的。

PDS 資料集名稱的語法如下所示：

```
//pds_data_set_name{;attribute;..;attribute}
```

資料集名稱的語法如同 **-ds** (*destination_sequential_data_set*) 參數所述。指定位於 Connect:Direct 節點上的資料集的所有語法詳細資料，也適用於 **-dp** 參數。如果目的地代理程式是 Connect:Direct 橋接器代理程式，則您必須同時指定 **-de** 參數，值為 *overwrite*。

當目的地代理程式是通訊協定橋接器代理程式時，不支援 **-dp** 參數。

-du destination_user

選用項目。將檔案傳送至其中的目的地檔案空間的使用者名稱。

需要 **-td**、**-df**、**-dd**、**-ds**、**-dp**、**-du** 及 **-dq** 參數之一。如果您指定 **-du** 參數，則無法指定 **-td**、**-dd**、**-df**、**-dp**、**-dq** 或 **-ds** 參數，因為這些參數是互斥的。

當目的地代理程式是通訊協定橋接器代理程式或 Connect:Direct 橋接器代理程式時，不支援 **-du** 參數。

-dq destination_queue

選用項目。檔案傳送至其中的目的地佇列名稱。您可以選擇性地使用 QUEUE@QUEUEMANAGER 格式，在此規格中包含佇列管理程式名稱。如果未指定佇列管理程式名稱，而且未將 `enableClusterQueueInputOutput` 代理程式內容設定為 `true`，則會使用目的地代理程式佇列管理程式名稱。如果已將 `enableClusterQueueInputOutput` 代理程式內容設定為 `true`，則目的地代理程式將使用標準 IBM MQ 解析程序來判定放置佇列的位置。您必須指定佇列管理程式上存在的有效佇列名稱。

需要 **-td**、**-df**、**-dd**、**-ds**、**-dp**、**-du** 及 **-dq** 參數之一。如果您指定 **-dq** 參數，則無法指定 **-td**、**-dd**、**-df**、**-dp**、**-du** 或 **-ds** 參數，因為這些參數是互斥的。

當目的地代理程式是通訊協定橋接器代理程式或 Connect:Direct 橋接器代理程式時，或當來源規格是佇列時，不支援 **-dq** 參數。

-dqp persistent

選用項目。指定寫入目的地佇列的訊息是否持續保存。有效選項如下所示：

真

將持續性訊息寫入目的地佇列。這是預設值。

false

將非持續性訊息寫入目的地佇列。

qdef

持續性值取自目的地佇列的 `DefPersistence` 屬性。

如果您還指定了 **-dq** 參數，則只能指定 **-dqp** 參數。

-qmp boolean

選用項目。指定由傳送寫入目的地佇列的第一則訊息是否已設定 IBM MQ 訊息內容。有效選項如下所示：

真

在傳送所建立的第一個訊息上設定訊息內容。

false

不在由傳送建立的第一則訊息上設定訊息內容。這是預設值。

如果您還指定了 **-dq** 參數，則只能指定 **-qmp** 參數。如需相關資訊，請參閱第 569 頁的『寫入目的地佇列的訊息上由 MFT 設定的 MQ 訊息內容』。

-qs message_size

選用項目。指定是否要將檔案分割為多則固定長度的訊息。所有訊息都具有相同的 IBM MQ 群組 ID；群組中的最後一則訊息已設定 IBM MQ LAST_MSG_IN_GROUP 旗標。訊息的大小由 *message_size* 的值指定。*message_size* 的格式為 *lengthunits*，其中 *length* 為正整數值，*units* 為下列其中一個值：

B

位元組。容許的最小值為目的地訊息字碼頁的每個字元佔用的位元組數最大值的兩倍。

K

這相等於 1024 個位元組。

M

這相等於 1048576 個位元組。

如果您為 **-t** 參數指定值 *text*，且檔案採用雙位元組字集或多位元組字集，則會依距指定訊息大小最近的字元界限將檔案分割為訊息。

如果您還指定了 **-dq** 參數，則只能指定 **-qs** 參數。您只能指定 **-qs**、**-dqdb** 及 **-dqdt** 參數之一。

-dqdb hexadecimal_delimiter

選用項目。指定將一個二進位檔分割為多則訊息時要使用的十六進位定界字元。所有訊息都具有相同的 IBM MQ 群組 ID；群組中的最後一則訊息已設定 IBM MQ LAST_MSG_IN_GROUP 旗標。指定十六進位位元組作為定界字元的格式是 *xNN*，其中 *N* 是範圍介於 0-9 或 a-f 間的字元。透過指定以逗點區隔的十六進位位元組清單（例如：x3e,x20,x20,xbf），您可以將十六進位位元組的序列指定為定界字元。

如果您還指定了 **-dq** 參數，且傳送採用二進位模式，則只能指定 **-dqdb** 參數。您只能指定 **-qs**、**-dqdb** 及 **-dqdt** 參數之一。

-dqdt pattern

選用項目。指定將一個文字檔分割為多則訊息時要使用的正規表示式。所有訊息都具有相同的 IBM MQ 群組 ID；群組中的最後一則訊息已設定 IBM MQ LAST_MSG_IN_GROUP 旗標。將正規表示式指定為定界字元所採用的格式為：以括弧括住正規表示式 (*regular_expression*)。此參數值的求值方式與 Java 正規表示式相同。如需相關資訊，請參閱第 554 頁的『MFT 所使用的正規表示式』。

依預設，目的地代理程式限制正規表示式可以比對的字串長度為五個字元。您可以使用 **maxDelimiterMatchLength** 代理程式內容變更此行為。如需相關資訊，請參閱第 464 頁的『進階代理程式內容』。

如果您還指定了 **-dq** 參數，及為 **-t** 參數指定值 *text*，則只能指定 **-dqdt** 參數。您只能指定 **-qs**、**-dqdb** 及 **-dqdt** 參數的其中一個。

-dqdp

選用項目。指定分割檔案時預期的目的地文字及二進位定界字元的位置。如果您還指定了 **-dqdt** 及 **-dqdb** 參數之一，則只能指定 **-dqdp** 參數。

請指定下列其中一個選項：

字首

定界字元應該在每一行開頭。

postfix

定界字元應該在每一行結尾。這是預設選項。

-qi

選用項目。指定是否要在訊息中包含將檔案分割為多則訊息所使用的定界字元。如果已指定 **-qi** 屬性，則會在訊息（包含定界字元之前的檔案資料）的結尾包含定界字元。依預設，訊息不包括定界字元。

如果您還指定了 **-dqdt** 及 **-dqdb** 參數之一，則只能指定 **-qi** 參數。

-de destination_file_behavior

選用項目。指定如果目的地系統上存在目的地檔案時要採取的動作。有效選項如下所示：

error

報告錯誤，且不傳送檔案。這是預設值。

overwrite

改寫現有目的地檔案。

如果您指定 **-de** 參數，則不能指定 **-td** 參數，因為這些參數是互斥的。

-sd source_file_disposition

選用項目。指定來源檔案已順利地傳送至其目的地時，對來源檔案採取的動作。有效選項如下所示：

leave

來源檔案維持不變。這是預設值。

delete

在順利傳送來源檔案之後，從來源系統刪除來源檔案。

z/OS 在 z/OS 上，如果來源是磁帶資料集，且您指定 **delete** 選項，則會重新裝載磁帶來刪除此資料集。此行為是因為系統環境行為而造成。

如果來源是佇列，且您指定 **leave** 選項，則指令會傳回錯誤，不會要求傳送。

如果來源代理程式是 Connect:Direct 橋接器代理程式，且您指定 **delete** 選項，則行為不同於平常的來源處置行為。發生下列其中一種情況：

- 如果 Connect:Direct 使用 Managed File Transfer 產生的程序，從來源中移動檔案或資料集，則指定 **delete** 選項會造成傳送失敗。若要指定刪除來源檔案，您必須提交使用者定義的 Connect:Direct 程序。如需相關資訊，請參閱第 200 頁的『[透過檔案傳送要求提交使用者定義的 Connect:Direct 程序](#)』。
- 如果 Connect:Direct 利用使用者定義的程序從來源中移出檔案或資料集，則會透過 **%FTEFDISP** 內建符號變數將此參數傳遞給程序。使用者定義的程序決定是否刪除來源。傳送所傳回的結果，取決於使用者定義的程序所傳回的結果。

如果您指定 **-sd** 參數，則不能指定 **-td** 參數，因為這些參數是互斥的。但是，您可以在傳送定義檔中指定來源處置行為。

-pr transfer_priority

選用項目。指定傳送的優先順序層次。優先順序是範圍 0-9 中的值，其中 0 是最低優先順序。預設優先順序層次是 0，依預設，傳送時會使用來源代理程式的優先順序層次。

此值符合 IBM MQ 所使用的訊息優先順序值。如需相關資訊，請參閱從佇列取得訊息：[優先順序](#)。檔案傳送資料的訊息傳輸預設為優先順序層次 0，這可讓 IBM MQ 訊息傳輸取得優先權。

V 9.0.1

-rt 傳送回復逾時

選用項目。設定時間量（以秒為單位），在此期間內，來源代理程式會一直嘗試回復已停止的檔案傳送。請指定下列其中一個選項：

-1
代理程式繼續嘗試回復已停止的傳送，直至傳送完成為止。使用此選項相當於代理程式在未設定此內容時的預設行為。

0
一旦進入回復，代理程式即停止檔案傳送。

>0
代理程式繼續嘗試回復已停止的傳送，直至達到已指定的正整數值所設定的時間量（以秒為單位）為止。例如：

```
-rt 21600
```

指出代理程式從進入回復的 6 小時內會一直嘗試回復傳送。此參數的最大值為 999999999。

透過這種方式指定傳送回復逾時值是以每個傳送為基礎來進行設定。若要為 Managed File Transfer 網路中的所有傳送設定廣域值，您可以在 `agent.properties` 檔中新增一個內容。

-p configuration_options

選用項目。此參數會決定用於建立傳送範本的配置選項集。請使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，指令即會使用與此非預設協調佇列管理程式相關聯的一組內容檔。

如果未指定此參數，則會使用一組以預設協調佇列管理程式為基礎的配置選項。

-r

選用項目。當 `source_specification` 包含萬用字元時，遞迴地傳送子目錄中的檔案。在隨 Managed File Transfer 一起提供萬用字元作為 `source_specification` 時，僅在指定 **-r** 參數時傳送與萬用字元相符的任何目錄。在 `source_specification` 與子目錄相符時，該目錄及其子目錄中的所有檔案（包括隱藏式檔案）都一律會傳送。

如需 Managed File Transfer 如何處理萬用字元的相關資訊，請參閱[使用萬用字元](#)

如果您指定 **-r** 參數，則不能指定 **-td** 參數，因為這些參數是互斥的。不過，您可以在傳送定義檔中指定遞迴行為。

-t

選用項目。指定檔案傳送的類型：二進位模式或文字模式。

二進位

傳送檔案中的資料時不經過任何轉換。這是預設值。

text

轉換檔案的字碼頁及行尾字元。執行的確切轉換視來源代理程式及目的地代理程式的作業系統而定。

► **z/OS** 例如，從 Windows 傳送至 z/OS 的檔案，其字碼頁會從 ASCII 轉換為 EBCDIC。檔案從 ASCII 轉換為 EBCDIC 時，行尾字元會從 ASCII 歸位 (CR) 及換行 (LF) 字元配對轉換為 EBCDIC 新行 (NL) 字元。

► **z/OS** 如需如何傳送 z/OS 資料集的相關資訊，請參閱在 [z/OS 與分散式系統之間傳送檔案及資料集及在資料集之間傳送](#)。

如果您指定 **-t** 參數，則不能指定 **-td** 參數，因為這些參數是互斥的。但是，您可以在傳送定義檔中指定傳送模式行為。

-cs

選用項目。指定是否對檔案傳送資料執行總和檢查演算法，以檢查所傳送檔案的完整性。請指定下列其中一個選項：

MD5

計算資料的 MD5 總和檢查。來源及目的地檔案的總和檢查結果會寫入傳送日誌中作為驗證用途。依預設，Managed File Transfer 會計算所有檔案傳送的 MD5 總和檢查。

無

不計算檔案傳送資料的 MD5 總和檢查。傳送日誌會記錄總和檢查設為 `none`，總和檢查的值為空白。例如：

```
<checksum method="none"></checksum>
```

如果使用 `none` 選項，視環境而定，可能會改善檔案傳送效能。不過，選取此選項表示不驗證來源或目的地檔案。

如果您指定 `-cs` 參數，則不能指定 `-td` 參數，因為這些參數是互斥的。但是，您可以在傳送定義檔中指定總和檢查行為。

-tr

選用項目。指定要進行檔案傳送時必須成立的條件。如果條件不是 `true`，根據來源代理程式，檔案傳送會被捨棄，無法進行傳送。請指定下列格式：

```
condition,namelist
```

其中 `condition` 是下列其中一個值：

file=exist

名稱清單中至少一個檔案存在。亦即，如果名稱清單中的任何檔案存在，則條件成立。

file!=exist

`namelist` 中至少其中一個檔案不存在。亦即，如果名稱清單中的任何檔案不存在，則條件成立。

filesize>=size

`namelist` 中至少其中一個檔案存在，且以 `size` 指定大小下限。`size` 的值是整數，選用大小單位為 KB、MB 或 GB。例如，`filesize">"=10KB`。如果未指定大小單位，則假設以位元組為大小單位。在所有作業系統上，當您在指令行指定 `filesize` 選項時，必須以雙引號括住大於符號 (>)，如這個範例所示。

其中，`namelist` 是位於來源系統上的檔案名稱清單（以逗點區隔）。視作業系統而定，如果您想要在名稱清單中使用包含空格的路徑名稱或檔案名稱，可能必須以雙引號括住路徑名稱和檔案名稱。

多次使用 `-tr` 參數可以指定多個觸發條件。不過，在此情況下，每一個觸發條件都必須成立，才會進行檔案傳送。

註：若要繼續監視觸發條件為 `true` 的資源，建立您使用資源監視。您可以使用 `fteCreateMonitor` 指令建立資源監視器。

在下列範例中，如果檔案 `A.txt` 及/或檔案 `B.txt` 存在於 `AGENT1` 上，以及檔案 `A.txt` 及/或檔案 `B.txt` 等於或大於 1 GB，則檔案 `file1.doc` 會從 `AGENT1` 傳送至 `AGENT2`：

```
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
```

您可以將觸發參數與排程參數相結合。如果指定這兩種參數，觸發條件會套用至排程參數所建立的檔案傳送。

-tl

選用項目。指定是否記載觸發程式失敗。請指定下列其中一個選項：

是


為失敗的觸發傳送建立日誌項目。即使未指定 `-tl` 參數，這也是預設行為。

否

不為失敗的觸發傳送建立日誌項目。

-md

選用項目。指定傳遞至代理程式的結束點的使用者定義的 meta 資料。 **-md** 參數可以採用一個以上名稱/值配對（以逗點區隔）。每一個名稱配對由 *name=value* 組成。您可以在指令中多次使用 **-md** 參數。

 在 z/OS 上，空格代表定界字元，因此您必須使用底線來區隔值。例如，使用 `kw=text1_text2_text3` 取代 `kw="text1 text2 text3"`

-tb

選用項目。指定您要用於排定檔案傳送的時間基準。亦即，使用系統時間或世界標準時間 (UTC)。您僅必須將此參數與 **-ss** 參數搭配使用。請指定下列其中一個選項：

admin

用於排定的傳送的開始及結束時間基於管理者使用的系統時間及日期。這是預設值。

source

用於排定的傳送的開始及結束時間基於來源代理程式所在系統的時間及日期。

UTC (世界標準時間)

用於排定的傳送的開始及結束時間基於「世界標準時間 (UTC)」。

-jn job_name

選用項目。在啟動傳送時新增至日誌訊息的使用者定義的工作名稱 ID。

-ss schedule_start_time

選用項目。指定要執行排定傳送的時間和日期。使用下列其中一種格式來指定時間和日期。使用 24 小時制指定時間：

```
yyyy-MM-ddThh:mm  
hh:mm
```

如果沒有可能會影響傳送的問題，排定的檔案傳送會在排程開始時間的一分鐘內啟動。例如，網路或代理程式可能發生問題，造成排定的傳送無法開始。

-oi

選用項目。指定排定的傳送發生的間隔。您僅必須將此參數與 **-ss** 參數搭配使用。請指定下列其中一個選項：

分鐘

時

日

週

月

年

-of occurrence_frequency

選用項目。指定排定的傳送發生的頻率。例如，每隔 **5** 週或每隔 **2** 個月。此參數只能搭配 **-oi** 及 **-ss** 參數一起指定。如果未指定此參數，則會使用預設值 **1**。

-oc occurrence_count

選用項目。指定您想要此排定傳送發生的次數。在符合發生次數之後，排定的傳送會刪除。

此參數只能搭配 **-oi** 及 **-ss** 參數一起指定。

如果您指定 **-oc** 參數，則不能指定 **-es** 參數，因為這些參數是互斥的。

您可以同時省略 **-oc** 及 **-es** 參數，以建立無限期重複的傳送。

-es schedule_end_time

選用項目。重複的排定傳送結束的時間及日期。

此參數只能搭配 **-oi** 及 **-ss** 參數一起指定。

如果您指定 **-es** 參數，則不能指定 **-oc** 參數，因為這些參數是互斥的。

您可以同時省略 **-es** 及 **-oc** 參數，以建立無限期重複的傳送。

請使用下列其中一種格式來指定結束時間及日期。使用 24 小時制指定時間：

```
yyyy-MM-ddThh:mm
```

```
hh:mm
```

-tn *template_name*

必要項目。要建立的範本的名稱。使用敘述性字串，以讓您稍後選取用於傳送的正確範本。對此字串的長度沒有特定限制，但請注意，過長的名稱可能無法正確顯示在部分使用者介面中。

請勿建立多個具有相同名稱的範本。

-sqgi

選用項目。指定訊息依 IBM MQ 群組 ID 進行分組。第一個完整群組會寫入目的地檔案。如果未指定此參數，則來源佇列上的所有訊息會寫入目的地檔案。

如果您還指定了 **-sq** 參數，則只能指定 **-sqgi** 參數。

-sqdt *text_delimiter*

選用項目。指定在將多則訊息附加到文字檔時，要作為定界字元插入的文字序列。您可以在定界字元中針對字串文字併入 Java ESC 序列。例如，`-sqdt \u007d\n`。

如果您還指定了 **-sq** 參數，及為 **-t** 參數指定值 `text`，則只能指定 **-sqdt** 參數。

-sqdb *hexadecimal_delimiter*

選用項目。指定在將多則訊息附加到二進位檔時，要作為定界字元插入的一個以上位元組值。每一個值必須指定為兩個十六進位數字，範圍為 00-FF，字首為 x。若有多個位元組，必須以逗點區隔。例如，`-sqdb x08,xA4`。

如果您還指定了 **-sq** 參數，則只能指定 **-sqdb** 參數。如果您還為 **-t** 參數指定值 `text`，則只能指定 **-sqdb** 參數。

-sqdp

選用項目。指定來源文字及二進位定界字元的插入項目的位置。如果您還指定了 **-sqdt** 及 **-sqdb** 參數之一，則只能指定 **-sqdp** 參數。

請指定下列其中一個選項：

字首

在每一個訊息的開頭插入定界字元。

postfix

在每一個訊息的結尾插入定界字元。這是預設選項。

-sqwt *wait_time*

選用項目。指定等待下列其中一個條件成立的時間（以秒為單位）：

- 使新訊息放在佇列上
- 如果已指定 **-sqgi** 參數，則條件為完整群組放在佇列上。

在 `wait_time` 指定的時間內，如果這兩個條件都不成立，來源代理程式會停止讀取佇列，並完成傳送。如果未指定 **-sqwt** 參數，則在以下情況下來源代理程式會立即停止從來源佇列進行讀取：來源佇列為空白或者已指定 **-sqgi** 參數，佇列上沒有完整群組。

如果您還指定了 **-sq** 參數，則只能指定 **-sqwt** 參數。

-sq

選用項目。指定傳送的來源是佇列。

-mquserid (userID)

選用項目。指定用來向協調佇列管理程式進行鑑別的使用者 ID。

-mqpassword (password)

選用項目。指定用來向協調佇列管理程式進行鑑別的密碼。您還必須指定 **-mquserid** 參數。如果指定了 **-mquserid**，卻未指定 **-mqpassword**，則系統會提示您提供相關聯的密碼。將不會顯示密碼。

source_specification

如果您已指定 **-df**、**-dd**、**-dp** 或 **-ds** 參數，則為必要項目。如果指定 **-td** 參數，請不要指定 *source_specification*。

- 如果未指定 **-sq** 參數，則 *source_specification* 是一個以上檔案規格，決定檔案傳送的一個以上來源。檔案規格以空格區隔。檔案規格可接受五種格式，也可包括萬用字元。如需 WMQFTE 中萬用字元的相關資訊，請參閱第 549 頁的『[搭配使用萬用字元與 MFT](#)』。您可以在檔案規格中使用兩個星號字元 (**)，以跳出檔案規格中的星號。

若要傳送其檔名包含空格的檔案，請以雙引號括住包含空格的檔名。例如，若要將檔案 a b.txt 傳送至檔案 c d.txt，請指定下列文字作為 **fteCreateTemplate** 指令的一部分：

```
-df "c d.txt" "a b.txt"
```

每一個檔案規格必須採用下列其中一種格式：

檔案名稱

檔案的名稱，使用來源代理程式執行所在系統的適當表示法表示。在檔名指定為來源檔案規格時，會複製檔案的內容。

目錄

目錄的名稱，使用來源代理程式執行所在系統的適當表示法表示。在目錄指定為來源檔案規格時，會複製目錄的內容。更準確地說，會複製目錄中及其所有子目錄中的所有檔案，包括隱藏式檔案。

例如，若要僅將 DIR1 的內容複製到 DIR2，請指定 DIR1/* DIR2

z/OS 循序資料集

(僅限 z/OS)。循序資料集的名稱或分割的資料集成員。在資料集名稱之前加上兩個正斜線字元 (//) 來表示資料集。

z/OS 分割的資料集

(僅限 z/OS)。分割資料集的名稱。在資料集名稱之前加上兩個正斜線字元 (//) 來表示資料集名稱。

Connect:Direct 節點上的檔案名稱或目錄

(僅限 Connect:Direct 橋接器代理程式)。Connect:Direct 節點的名稱、冒號字元 (:)，以及管理 Connect:Direct 節點的系統上的檔案或目錄路徑。例如，
`connect_direct_node_name:file_path`。

如果來源代理程式是 Connect:Direct 橋接器代理程式，則僅接受此格式的來源規格。

註：如果來源代理程式是 Connect:Direct 橋接器代理程式，檔案路徑中不支援萬用字元。

- 如果已指定 **-sq** 參數，則 *source_specification* 是來源代理程式佇列管理程式上的本端佇列名稱。您只能指定一個來源佇列。以下列格式指定來源佇列：

```
QUEUE_NAME
```

來源佇列規格中不包括佇列管理程式名稱，因為佇列管理程式與來源代理程式佇列管理程式必須相同。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，會建立稱為 payroll accounts monthly report template 的傳送範本。提交後，此範本會將具有副檔名 .xls 的任何檔案從代理程式 PAYROLL1 傳送至所指定目錄中的代理程式 ACCOUNTS:

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

在此範例中，會建立稱為 jupiter_neptune_sched_template 的傳送範本。提交後，範本會將檔案 originalfile.txt 從 QM_JUPITER 所在的系統傳送至 QM_NEPTUNE 所在的系統。檔案傳送排定在 09:00 進行（根據來源代理程式所在的系統的系統時間），並每隔兩小時進行四次：

```
fteCreateTemplate -tn jupiter_neptune_sched_template -sa AGENT1 -sm QM_JUPITER -da AGENT2 -dm QM_NEPTUNE -tb source -ss 09:00 -oi hours -of 2 -oc 4 -df C:\import\transferredfile.txt C:\export\originalfile.txt
```

在此範例中，會建立稱為 jupiter neptune trigger template 的傳送範本。在提交此範本後，如果檔案 A.txt 存在於 AGENT1 上，檔案 originalfile.txt 會從 AGENT1 傳送至 AGENT2:

```
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 在此範例中，會建立稱為 ascii_ebcdic_template 的範本。提交此範本後，檔案 originalfile.txt 會從 AGENT1 所在的系統傳送至 AGENT2 所在的系統上的資料集 '//USERID.TRANS.FILE.TXT'。已選取文字模式以將資料從 ASCII 轉換為 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 在此範例中，會建立稱為 ebcdic_ascii_template 的範本。提交此範本後，AGENT1 所在系統上的完整資料集的成員會傳送至 AGENT2 所在系統上的檔案。已選取文字模式以將檔案從 EBCDIC 轉換為 ASCII。

```
fteCreateTemplate -tn ebcdic_ascii_template -t text -sa AGENT1 -da AGENT2 -df /tmp/IEEUJV.txt "'/SYS1.SAMPLIB(IEEUJV)'"
```

回覆碼

回覆碼	說明
0	指令已順利完成。
1	指令未順利結束。

fteCreateTransfer: 啟動新的檔案傳送

fteCreateTransfer 指令從指令行建立並啟動新的檔案傳送。此指令可立即啟動檔案傳送、將檔案傳送排定在未來的時間和日期、將排定的傳送重複一次以上，以及根據特定條件觸發檔案傳送。

用途

使用 **fteCreateTransfer** 指令，從 Managed File Transfer 代理程式建立並啟動新的檔案傳送。

註: 檔案傳送只能在相同 Managed File Transfer 拓撲內的代理程式之間進行。

如需如何傳送檔案的相關指引，請參閱第 530 頁的『傳送檔案的準則』。若為 z/OS 平台，您可以傳送文字檔、資料集及世代資料群組 (GDG)。

您可以從可以連接至 IBM MQ 網路的任何系統執行 **fteCreateTransfer** 指令，然後遞送至來源代理程式佇列管理程式。具體而言，為了讓指令能夠執行，您必須在此系統上安裝 Managed File Transfer 元件（「服務」或「代理程式」），且必須在此系統上配置 Managed File Transfer 元件，才能與 IBM MQ 網路進行通訊。

此指令使用稱為 `command.properties` 的內容檔來連接至 IBM MQ 網路。如果 `command.properties` 檔不包含內容資訊，則會對本端系統上的預設佇列管理程式建立連結模式連線。如果 `command.properties` 檔案不存在，將會產生錯誤。如需相關資訊，請參閱 [第 460 頁的『MFT command.properties 檔』](#)。

您可以為一個檔案傳送指定多個來源檔案，但它們必須出自於單一來源代理程式，且終止於單一目的地代理程式。單一傳送內不支援在相同代理程式或多個不同代理程式上將單一來源檔案傳送至多個目的地檔案。Ant Script 可用來將相同來源檔案傳送至一個以上代理程式上的多個目的地。如需相關資訊，請參閱 [第 211 頁的『搭配使用 Apache Ant 與 MFT』](#)。

特殊字元

請謹慎使用包含特殊字元的參數，避免指令 Shell 以超乎預期的方式來解譯字元。例如，包含單引號的完整資料集名稱及包含星號字元的來源規格可能會由指令 Shell 解譯，而不是透過傳送要求來傳遞。為了避免由指令 Shell 解譯字元，請以雙引號括住整個參數，或使用指令 Shell 的 ESC 序列來跳出特殊字元。

相對路徑

fteCreateTransfer 指令支援使用相對檔案路徑。若為下列平台，依預設，會將路徑視為相對於執行該代理程式的使用者的起始目錄：

- ▶ **Multi** 多平台
- ▶ **z/OS** z/OS 上的 UNIX 系統服務

若要變更路徑名稱相對的目錄，請在 `agent.properties` 檔案中設定 `transferRoot` 內容。此檔案位於 `MQ_DATA_PATH/mqft/config/coordination_qmgr/agents/agent_name` 目錄中。將下面這一行加到檔案中：

```
transferRoot=directory_name
```

▶ **Windows** 例如，將 `C:\TransferRoot` 指定為 `C:\\TransferRoot` 或 `C:/TransferRoot`。

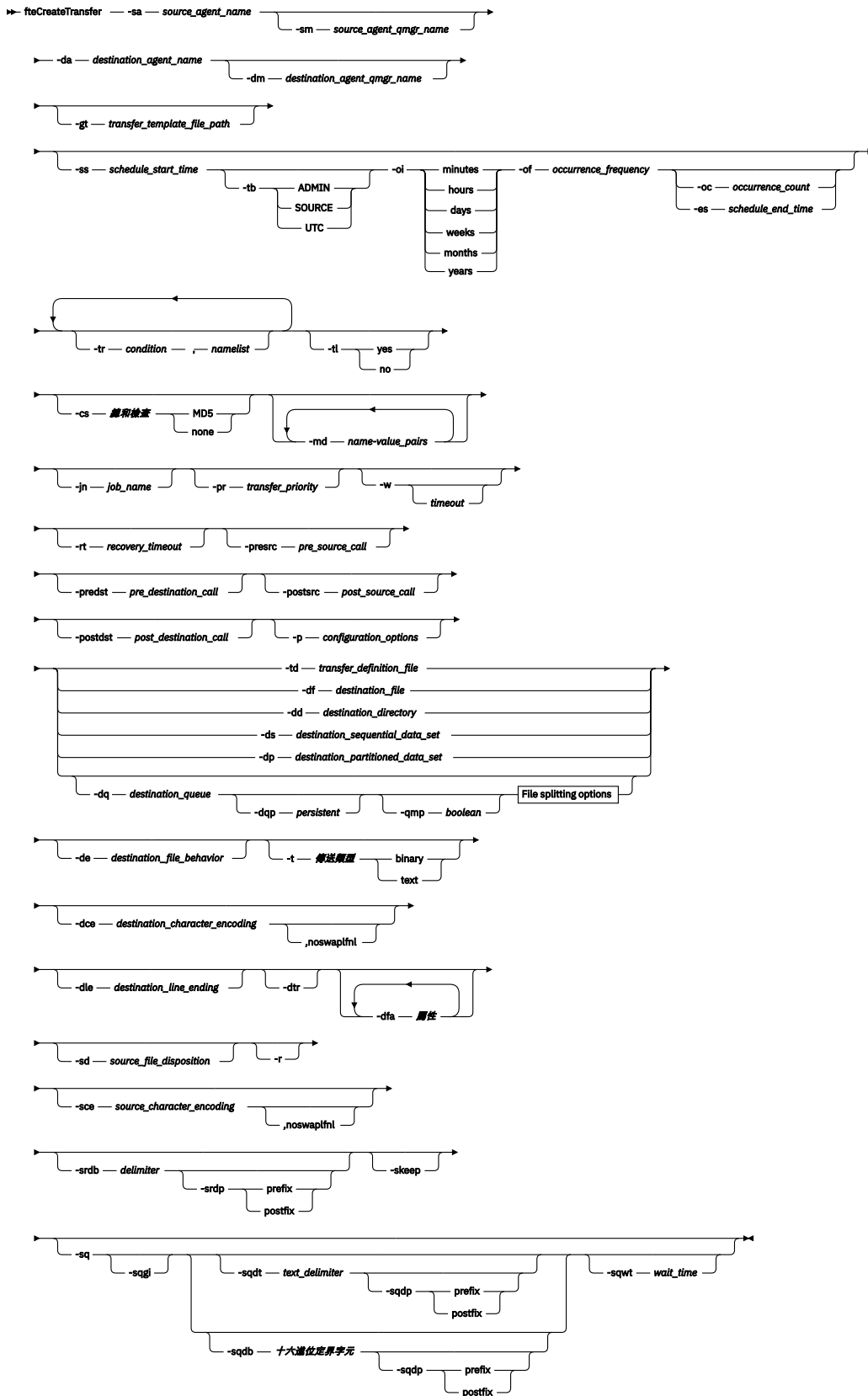
▶ **z/OS** 在 z/OS 上，依預設，執行此代理程式的使用者名稱會作為高階限定元字首新增至不完整的資料集規格。例如：`//ABC.DEF`。若要變更加在資料集名稱上作為字首的值，請在 `agent.properties` 檔案中設定 `transferRootHLQ` 內容。此檔案位於 `MQ_DATA_PATH/mqft/config/coordination_qmgr/agents/agent_name` 目錄中。將下面這一行加到檔案中：

```
transferRootHLQ=prepend_value
```

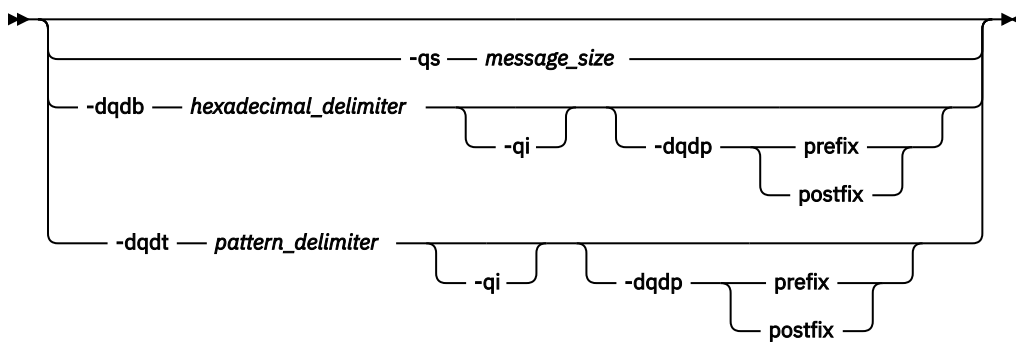
▶ **z/OS** 不過，如果傳送涉及 z/OS 系統上的 `Connect:Direct` 節點，則資料集規格將解譯成完整名稱。資料集名稱不會加上高階限定元。

語法

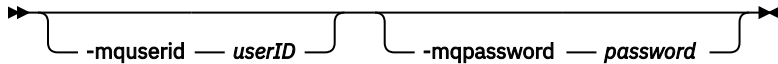
fteCreateTransfer



File splitting options



Parameters for MQ security



► source_specification ◄

代理程式規格的參數

-sa source_agent_name

必要項目。傳送來源檔案的來源代理程式名稱。

z/OS 如果指定通訊協定橋接器代理程式作為來源代理程式，則不能指定資料集作為來源檔案規格。

如果指定 **-td** 參數且傳送定義檔包含要用於傳送的來源代理程式，就不能指定 **-sa** 參數。

-sm source_agent_qmgr_name

選用項目。來源代理程式所連接的佇列管理程式的名稱。

如果未指定 **-sm** 參數，則使用的佇列管理程式由使用中的配置選項集根據來源代理程式名稱決定。如果找不到來源代理程式的 `agent.properties` 檔案，檔案傳送會失敗。

-da destination_agent_name

必要項目。檔案傳送至其中的目的地代理程式名稱。

如果指定 **-td** 參數且傳送定義檔包含要用於傳送之目的地代理程式，就不能指定 **-da** 參數。

-dm destination_agent_qmgr_name

選用項目。目的地代理程式所連接的佇列管理程式名稱。

如果未指定 **-dm** 參數，則使用的佇列管理程式由使用中的配置選項集根據目的地代理程式名稱決定。如果找不到目的地代理程式的 `agent.properties` 檔案，檔案傳送會失敗。

產生傳送範本的參數

-gt transfer_template_file_path

選用項目。產生傳送範本 XML 訊息，並將此訊息寫入檔案。如果指定此參數，則不會將任何傳送要求傳送至 Managed File Transfer。相反地，會將傳送要求訊息的內容寫入指定的 XML 文件。您可以使用此 XML 文件來定義資源監視的作業。如需如何建立資源監視器的相關資訊，請參閱 [fteCreateMonitor 指令](#)。如果未指定此參數，則會採用預設行為，並執行實際的傳送要求。

您必須提供 XML 輸出檔的完整路徑及名稱作為此參數的輸入，例如
C:\templates\transfer_reports.xml

z/OS 在 z/OS 上，您必須將傳送範本文件儲存在「z/OS UNIX 系統服務」上的 UNIX 檔案中。您不能將傳送範本文件儲存在 z/OS 循序檔或 PDS 成員中。

IBM i 在 IBM i 上，您必須將傳送範本文件儲存在整合檔案系統中。

您使用 **-gt** 參數建立的傳送範本 XML 訊息與您使用 **fteCreateTemplate** 指令建立的傳送不同，這表示兩種不同類型的範本不能互換使用。

註: 如果您想要透過執行 **fteCreateTransfer** 指令並指定 **-gt** 參數來產生傳送範本 XML 文件，然後使用 **-td** 參數提供該傳送範本 XML 文件作為 **fteCreateTransfer** 指令的輸入，則必須確保已產生傳送範本 XML 文件，並指定與 **-td** 選項互斥的那些參數。

-td 選項互斥的參數如下：

- **-dd** *destination_directory*
- 來源路徑
- **-df** *destination_file*
- **-cs** 總和檢查
- **-de** *destination_file_behavior*
- **-dq** *destination_queue*
- **-t** 傳送類型
- **-sd** *source_file_disposition*

例如，無法在 **fteCreateTransfer** 指令上同時指定 **-td** 和 **-t** 參數 (指出傳送是二進位或文字傳送)。這表示如果您想要將傳送範本 XML 文件傳入指令，並指定傳送應該是文字傳送，則應該透過指定 **-gt** 及 **-t** 文字參數來建立 XML 文件。

排程傳送的參數

-ss *schedule_start_time*

選用項目。指定要執行排定傳送的時間和日期。使用下列其中一種格式來指定時間和日期。使用 24 小時制指定時間：

```
yyyy-MM-ddThh:mm  
hh:mm
```

如果沒有可能會影響傳送的問題，排定的檔案傳送會在排程開始時間的一分鐘內啟動。例如，網路或代理程式可能發生問題，造成排定的傳送無法開始。

-tb

選用項目。指定您要用於排定檔案傳送的時間基準。亦即，使用系統時間或世界標準時間 (UTC)。您僅必須將此參數與 **-ss** 參數搭配使用。請指定下列其中一個選項：

admin

用於排定的傳送的開始及結束時間基於管理者使用的系統時間及日期。這是預設值。

source

用於排定的傳送的開始及結束時間基於來源代理程式所在系統的時間及日期。

UTC (世界標準時間)

用於排定的傳送的開始及結束時間基於「世界標準時間 (UTC)」。

-oi

選用項目。指定排定的傳送發生的間隔。您僅必須將此參數與 **-ss** 參數搭配使用。請指定下列其中一個選項：

分鐘

時

日

週

月
年

-of occurrence_frequency

選用項目。指定排定的傳送發生的頻率。例如，每隔 **5** 週或每隔 **2** 個月。此參數只能搭配 **-oi** 及 **-ss** 參數一起指定。如果未指定此參數，則會使用預設值 **1**。

-oc occurrence_count

選用項目。指定您想要此排定傳送發生的次數。達到發生次數之後，即會刪除排定傳送。

此參數只能搭配 **-oi** 及 **-ss** 參數一起指定。

如果您指定 **-oc** 參數，則不能指定 **-es** 參數，因為這些參數是互斥的。

您可以同時省略 **-oc** 及 **-es** 參數，以建立無限期重複的傳送。

-es schedule_end_time

選用項目。重複的排定傳送結束的時間及日期。

此參數只能搭配 **-oi** 及 **-ss** 參數一起指定。

如果您指定 **-es** 參數，則不能指定 **-oc** 參數，因為這些參數是互斥的。

您可以同時省略 **-es** 及 **-oc** 參數，以建立無限期重複的傳送。

請使用下列其中一種格式來指定結束時間及日期。使用 24 小時制指定時間：

```
yyyy-MM-ddThh:mm  
hh:mm
```

觸發傳送的參數

-tr

選用項目。指定要進行檔案傳送時必須成立的條件。如果條件不是 **true**，根據來源代理程式，檔案傳送會被捨棄，無法進行傳送。請指定下列格式：

```
condition,namelist
```

其中 *condition* 是下列其中一個值：

file=exist

名稱清單中至少一個檔案存在。亦即，如果名稱清單中的任何檔案存在，則條件成立。

file!=exist

namelist 中至少其中一個檔案不存在。亦即，如果名稱清單中的任何檔案不存在，則條件成立。

filesize>=size

namelist 中至少其中一個檔案存在，且以 *size* 指定大小下限。*size* 是整數，可加上選用的大小單位 KB、MB 或 GB。例如，**filesize">"=10KB**。如果未指定大小單位，則假設以位元組為大小單位。在所有作業系統上，當您在指令行指定 **filesize** 選項時，必須以雙引號括住大於符號 (>)，如這個範例所示。

其中，*namelist* 是以逗點區隔的檔案名稱清單，這些檔案位於與來源代理程式相同的系統上。視作業系統而定，如果您想要在名稱清單中使用包含空格的路徑名稱或檔案名稱，可能必須以雙引號括住路徑名稱和檔案名稱。

多次使用 **-tr** 參數可以指定多個觸發條件。不過，在此情況下，每一個觸發條件都必須成立，才會進行檔案傳送。

註：若要持續監視資源來等待觸發條件成立，強烈建議您使用資源監視。您可以使用 [fteCreateMonitor](#) 指令來建立資源監視器。

在下列範例中，如果檔案 A.txt 及/或檔案 B.txt 存在於 AGENT1 上，以及 檔案 A.txt 及/或檔案 B.txt 等於或大於 1 GB，則檔案 file1.doc 會從 AGENT1 傳送至 AGENT2：

```
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
```

您可以將觸發參數與排程參數相結合。如果指定這兩種參數，觸發條件會套用至排程參數所建立的檔案傳送。

通訊協定橋接器代理程式不支援 **-tr** 參數。

-tl

選用項目。指定是否將觸發失敗寫入傳送日誌。請指定下列其中一個選項：

是

會針對失敗的觸發傳送建立日誌項目。即使未指定 **-tl** 參數，這也是預設行為。

否

不會針對失敗的觸發傳送建立日誌項目。

指定傳送選項的參數

-jn job_name

選用項目。使用者定義的工作名稱 ID，當傳送開始時會將它新增至傳送日誌訊息。

-md

選用項目。指定使用者定義的 meta 資料，以傳遞給代理程式執行的結束點。 **-md** 參數可接受一個以上以逗點區隔的名稱/值配對。每一個名稱配對由 *name=value* 組成。您可以在指令中多次使用 **-md** 參數。

當代理程式內容 `enableUserMetadataOptions` 設定為 `true` 值時，使用者定義的特定 meta 資料索引鍵即會為傳送提供其他選項。如需目前支援之使用者定義的 meta 資料索引鍵相關資訊，請參閱第 480 頁的『[enableUserMetadataOptions: 支援的 MFT 使用者定義 meta 資料索引鍵](#)』。當 `enableUserMetadataOptions` 內容設為 `true` 時，不支援使用開頭為 `com.ibm.wmqfte.` 的索引鍵名稱作為使用者定義索引鍵。

-cs checksum

選用項目。指定是否對檔案傳送資料執行總和檢查演算法，以檢查所傳送檔案的完整性。請指定下列其中一個選項：

MD5

計算資料的 MD5 總和檢查。來源及目的地檔案的總和檢查結果會寫入傳送日誌中作為驗證用途。依預設，Managed File Transfer 會計算所有檔案傳送的 MD5 總和檢查。

無

不計算檔案傳送資料的 MD5 總和檢查。傳送日誌會記錄總和檢查設為 `none`，總和檢查的值為空白。例如：

```
<checksum method="none"></checksum>
```

如果使用 `none` 選項，視環境而定，可能會改善檔案傳送效能。不過，選取此選項表示不驗證來源或目的地檔案。

如果您指定 **-cs** 參數，則不能指定 **-td** 參數，因為這些參數是互斥的。但是，您可以在傳送定義檔中指定總和檢查行為。

-pr transfer_priority

選用項目。指定傳送的優先順序層次。優先順序是範圍 0-9 中的值，其中 0 是最低優先順序。預設優先順序層次是來源代理程式的優先順序層次。

此值符合 IBM MQ 的訊息優先順序值。如需相關資訊，請參閱從佇列取得訊息：優先順序。檔案傳送資料的訊息傳輸預設為優先順序層次 0，這可讓 IBM MQ 訊息傳輸取得優先權。

-qmp boolean

選用項目。指定由傳送寫入目的地佇列的第一則訊息是否已設定 IBM MQ 訊息內容。有效選項如下所示：

true

在由傳送建立的第一則訊息上設定訊息內容。

false

不在由傳送建立的第一則訊息上設定訊息內容。這是預設值。

只有在同時指定 **-dq** 參數時，才可指定 **-qmp** 參數。如需相關資訊，請參閱第 569 頁的『寫入目的地佇列的訊息上由 MFT 設定的 MQ 訊息內容』。

-qs message_size

選用項目。指定是否要將檔案分割為多則固定長度的訊息。所有訊息都具有相同的 IBM MQ 群組 ID；群組中的最後一則訊息已設定 IBM MQ LAST_MSG_IN_GROUP 旗標。訊息的大小由 *message_size* 的值指定。*message_size* 的格式為 *lengthunits*，其中 *length* 為正整數值，*units* 為下列其中一個值：

B

位元組。容許的最小值為目的地訊息字碼頁的每個字元位元組數最大值的兩倍。

K

這相等於 1024 個位元組。

M

這相等於 1048576 個位元組。

如果檔案以文字模式傳送，且為雙位元組字集或多位元組字集，則會依指定的訊息大小，在最近字元界限上將檔案分割成多個訊息。

只有在同時指定 **-dq** 參數時，才可指定 **-qs** 參數。您只能指定 **-qs**、**-dqdb** 及 **-dqdt** 參數的其中一個。

-qi

選用項目。使用此選項包括用來將檔案分割成訊息中多個訊息的定界字元。在訊息的開頭或結尾處併入定界字元，視 **-dqdp** 參數而定（指定字首或字尾）。依預設，訊息不包括定界字元。

只有在同時指定 **-dqdt** 及 **-dqdb** 參數的其中一個時，才可指定 **-qi** 參數。

-p configuration_options

選用項目。此參數決定一組用來建立檔案傳送的配置選項。請使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，指令即會使用與此非預設協調佇列管理程式相關聯的內容檔集。

如果未指定此參數，則會使用基於預設協調佇列管理程式的配置選項集。

-w timeout

選用項目。指定 **-w** 參數可讓 **fteCreateTransfer** 指令在返回之前等待來自代理程式的回應。如果不指定此參數，則 **fteCreateTransfer** 指令會等待最多 5 秒鐘，以從傳送的來源代理程式接收關於代理程式已收到傳送要求的確認通知。如果在五秒的等待時間內未接收到確認通知，**fteCreateTransfer** 指令會傳回下列警告訊息：

```
BFGCL0253W: No acknowledgment to command from agent within timeout.
```

timeout 是選用引數。如果指定 *timeout*，**fteCreateTransfer** 指令等待代理程式回應的最長時間為 *timeout* 秒。如果代理程式在達到時間限制之前未回應，則指令會產生警告，並以回覆碼 2 結束。如果您未指定 *timeout* 值，或將 *timeout* 值指定為 -1，則指令會等待直到代理程式回應為止。

V 9.0.1

-rt recovery_timeout

選用項目。設定時間量（以秒為單位），在此期間內，來源代理程式會一直嘗試回復已停止的檔案傳送。請指定下列其中一個選項：

-1

代理程式繼續嘗試回復已停止的傳送，直至傳送完成為止。使用此選項相當於代理程式在未設定此內容時的預設行為。

0

一旦進入回復，代理程式即停止檔案傳送。

>0

代理程式繼續嘗試回復已停止的傳送，直至達到已指定的正整數值所設定的時間量（以秒為單位）為止。例如：

```
-rt 21600
```

指出代理程式從進入回復的 6 小時內會一直嘗試回復傳送。此參數的最大值為 999999999。

透過這種方式指定傳送回復逾時值是以每個傳送為基礎來進行設定。若要為 Managed File Transfer 網路中的所有傳送設定廣域值，您可以在 `agent.properties` 檔中新增一個內容。

呼叫程式的參數

如需如何從 Managed File Transfer 啟動程式的相關資訊，請參閱第 209 頁的『指定要使用 MFT 執行的程式』。關於使用此處所述參數來指定要呼叫之程式的範例，請參閱第 728 頁的『使用 `fteCreateTransfer` 啟動程式的範例』。

-presrc pre_source_call


選用項目。指定在傳送開始之前要在來源代理程式上呼叫的程式。 `pre_source_call` 使用下列格式：

```
[type:]commandspec[, [retrycount][, [retrywait][, successsrc]]]
```

在此語法中，變數是：

類型

選用項目。有效值為 **executable**、**antscript** 和 **jcl**。預設值是 **executable**。

 只有在目標為 z/OS 環境中的代理程式時，**jcl** 值才適用。在此情況下，該指令會參照 ZFS 檔案或 QSAM 可讀取的資料集，或者 PDS 成員。內容應為可以提交的 JCL。

commandspec

必要項目。指令規格。請使用下列其中一種格式：

- 類型 **executable**: `command[(arg1,arg2,...)]`

如果引數包含變數替代，例如 `${FilePath}` 或 `${FileName}`，則變數會替換為傳送要求中的第一個項目。

例如，如果傳送要求包含檔案 "reports01.csv, reports02.csv, reports03.csv"，且目的地目錄為 "/output"，則下列傳送要求：

```
fteCreateTransfer -sa 1 -da 2 -presrc "executable:archive(${FileName})"  
-dd TargetDir "${FilePath}" -gt task.xml
```

取代為

```
fteCreateTransfer -sa 1 -da 2 -presrc "executable:archive(reports01.csv)"  
-dd TargetDir "/ouptut" -gt task.xml
```

- 類型 **antscript**: `command[(name1=var1|target1,name2=var2|target2,...)]`
- 類型 **jcl**: `command`

其中：

指令

必要項目。要呼叫的程式名稱。

只有在目標為 z/OS 環境中的代理程式時，**jc1** 值才適用。

方括弧 ([]) 中是選用引數，語法視指令類型而定。指令或參數內的括弧、逗點 (,) 及反斜線字元 (\)，必須以反斜線字元 (\) 跳出。

retrycount

選用項目。在程式未傳回成功回覆碼的情況下，重試呼叫程式的次數。預設值是 0。

retrywait

選用項目。重新嘗試呼叫程式之前等待的時間（以秒為單位）。預設值是 0（重試之間不等待）。

successrc

選用項目。用來判斷程式呼叫何時順利執行的表示式。此表示式可由一個以上表示式組成。使用垂直線字元 (|) 結合這些表示式可表示布林值 OR，或使用 & 字元結合可表示布林值 AND。每一個表示式使用下列格式：

```
[>|<|!]value
```

其中

>
選用項目。value 的大於測試。

<
選用項目。value 的小於測試。

!
選用項目。value 的不等於測試。

value

必要項目。有效的整數。

-predst pre_destination_call

選用項目。指定在傳送開始之前要在目的地代理程式上呼叫的程式。pre_destination_call 與 pre_source_call 的格式相同。

-postsrc post_source_call

選用項目。指定在傳送完成之後要在來源代理程式上呼叫的程式。post_source_call 與 pre_source_call 的格式相同。

-postdst post_destination_call

選用項目。指定在傳送完成之後要在目的地代理程式上呼叫的程式。post_destination_call 與 pre_source_call 的格式相同。


指定目的地的參數

需要 **-td**、**-df**、**-dd**、**-ds**、**-dq** 及 **-dp** 參數其中之一。傳送要求中不可指定多個這些參數；它們彼此互斥。

-td transfer_definition_file

選用項目。XML 文件的名稱，定義傳送的一個以上來源及目的地檔案規格。或者，包含受管理傳送要求的 XML 文件名稱（可能已由 **-gt** 參數產生）。如果在指令行指定 **-td** 參數，同時也指定其他參數，則其他這些參數會置換傳送定義檔中的對應值。

fteCreateTransfer 指令會在現行目錄的相對位置尋找傳送定義檔。如果您無法使用相對路徑表示法來指定傳送定義檔的位置，請改用傳送定義檔的完整路徑及檔名。

 在 z/OS 上，您必須將傳送定義檔儲存在「z/OS UNIX 系統服務」上的 UNIX 檔案中。您不能將傳送定義檔儲存在 z/OS 循序檔或 PDS 成員中。

IBM i 在 IBM i 上，您必須將傳送定義檔儲存在整合檔案系統中。

如需相關資訊，請參閱[使用傳送定義檔](#)。

-df destination_file

選用項目。目的地檔案的名稱。

如果目的地代理程式是 Connect:Direct 橋接器代理程式，則會以 `connect_direct_node_name:file_path` 格式來指定目的地檔案。Connect:Direct 橋接器代理程式僅接受以此格式指定的檔案路徑。**z/OS** 如果目的地代理程式是 Connect:Direct 橋接器代理程式，且目的地是 PDS 成員，則您必須同時指定 **-de** 參數，值為 `overwrite`。

請注意下列資訊：

- 如果目的地代理程式是通訊協定橋接器代理程式，而您想要指定檔案的端點，請使用下列格式：

```
protocol_server:file_path
```

其中 `protocol_server` 是通訊協定伺服器名稱（選用），`file_path` 是通訊協定伺服器系統上的檔案路徑。如果未指定通訊協定伺服器，則會使用預設通訊協定伺服器。

- 如果您想要對目的地代理程式呼叫您已定義的任何 Managed File Transfer 傳送 I/O 使用者結束程式，您可以在傳送中使用 **-df** 參數。
- **z/OS** 當目的地代理程式位於 z/OS 上，如果指定的檔案是以 // 為開頭，則假設它是分割的 z/OS 資料集。

-dd destination_directory

選用項目。傳送檔案的目的地目錄名稱。指定正在執行目的地代理程式的系統上的有效目錄名稱。

如果目的地代理程式是 Connect:Direct 橋接器代理程式，則會以 `connect_direct_node_name:directory_path` 格式指定目的地目錄。如果目的地代理程式是 Connect:Direct 橋接器代理程式，且目的地是 PDS，則您必須同時指定 **-de** 參數，值為 `overwrite`。

請注意下列資訊：

- 如果目的地代理程式是通訊協定橋接器代理程式，而您想要指定特定端點上的目錄，請使用下列格式：

```
protocol_server:directory_path
```

其中 `protocol_server` 是通訊協定伺服器名稱（選用），`directory_path` 是通訊協定伺服器系統上的目錄路徑。如果未指定通訊協定伺服器，則會使用預設通訊協定伺服器。

- 如果您想要對目的地代理程式呼叫您已定義的任何 Managed File Transfer 傳送 I/O 使用者結束程式，您可以在傳送中使用 **-dd** 參數。
- **z/OS** 當目的地代理程式位於 z/OS 上，如果指定的檔案是以 // 為開頭，則假設它是 z/OS 分割的資料集。

z/OS -ds destination_sequential_data_set

僅限 z/OS。選用項目。檔案傳送至其中的循序資料集名稱或 PDS 成員。指定循序資料集名稱或分割的資料集成員。如需傳送資料集的相關資訊，請參閱第 530 頁的『[傳送檔案的準則](#)』。

資料集名稱的語法如下所示：

```
//data_set_name{;attribute(value);..;attribute(value)}
```

or

```
//pds_data_set_name(member_name){;attribute(value);..;attribute(value)}
```

亦即，以 // 為字首的資料集名稱指定元，選擇性地後接許多以分號區隔的屬性。

例如：

```
//'TEST.FILE.NAME';DSNTYPE(PDS);RECFM(F,B);BLKSIZE(800);LRECL(80);CYL;SPACE(2,2)
```

如果資料集位於 Connect:Direct 節點，則資料集名稱字首必須是節點名稱。例如：

```
CD_NODE1://'OBJECT.LIB';RECFM(F,B);BLKSIZE(800);LRECL(80)
```

如果目的地代理程式是 Connect:Direct 橋接器代理程式，且目的地是 PDS 成員，則您必須同時指定 **-de** 參數，值為 `overwrite`。如需往返於 Connect:Direct 節點的資料集傳送的相關資訊，請參閱第 534 頁的『從 Connect:Direct 節點來回傳送資料集』。

若為僅涉及 Managed File Transfer 代理程式的傳送，如果以單引號字元括住資料集名稱部分，則指定完整的資料集名稱。如果未以單引號字元括住資料集名稱，系統會加上目的地代理程式的預設高階限定元（`transferRootHLQ` 代理程式內容的值，或者，如果未設定 `transferRootHLQ`，則是執行代理程式的使用者 ID）。

註： **z/OS** 不過，如果傳送涉及 z/OS 系統上的 Connect:Direct 節點，則資料集規格將解譯成完整名稱。資料集名稱不會加上高階限定元。即使以單引號字元括住資料集名稱，作法還是一樣。

當您將檔案或資料集傳送至磁帶時，任何已在磁帶上的現有資料集都會被取代。新資料集的屬性將根據傳送定義中所傳遞的屬性進行設定。如果您沒有指定任何屬性，會將屬性設定為與來源資料集相同，或者，在來源為檔案時設定為預設值。現有磁帶資料集的屬性會被忽略。

資料集屬性用於建立資料集，或確保現有的資料集是相容的。資料集屬性的規格採用適合 BPXWDYN 的格式（如需相關資訊，請參閱[要求動態配置](#)）。當代理程式準備建立目的地資料集時，會動態指定下列 BPXWDYN 屬性：`DSN(data_set_name) NEW CATALOG MSG(numeric_file_descriptor)`。Managed File Transfer 會產生 `numeric_file_descriptor` 的值。對於資料集至資料集傳送，則為新目的地資料集從來源選取 RECFM、LRECL 及 BLKSIZE 的屬性。新的目的地資料集的 SPACE 設定不是由 Managed File Transfer 設定，而是使用系統預設值。因此，建立新的資料集時，建議您指定 SPACE 屬性。您可以在 `agent.properties` 檔中使用 `bpxwdynAllocAdditionalProperties` 內容，來設定套用至所有傳送的 BPXWDYN 選項。如需相關資訊，請參閱第 463 頁的『MFT agent.properties 檔案』。

z/OS 使用 `fteCreateTemplate` 指令、`fteCreateTransfer` 指令或 `agent.properties` 檔中的 `bpxwdynAllocAdditionalProperties` 內容時，不得指定某些 BPXWDYN 選項。如需這些內容的清單，請參閱第 540 頁的『不得與 MFT 一起使用的 BPXWDYN 內容』。

當目的地代理程式是通訊協定橋接器代理程式時，不支援 **-ds** 參數。

如果您想要對代理程式呼叫您已定義的任何 Managed File Transfer 傳送 I/O 使用者結束程式，請不要在傳送中指定 **-ds** 參數。使用 **-ds** 參數會導致無法對目的地呼叫傳送 I/O 使用者結束程式，並表示改用標準 Managed File Transfer I/O。

z/OS **-dp destination_partitioned_data_set**

僅限 z/OS。選用項目。檔案傳送至其中的目的地 PDS 名稱。指定分割的資料集名稱。如果因為傳送而建立 PDS，此 PDS 依預設會建立為 PDSE。您可以指定 `DSNTYPE=PDS` 來置換預設值。

PDS 資料集名稱的語法如下所示：

```
//pds_data_set_name{;attribute;..;attribute}
```

資料集名稱的語法如同 **-ds** (`destination_sequential_data_set`) 參數所述。指定位於 Connect:Direct 節點上的資料集的所有語法詳細資料，也適用於 **-dp** 參數。如果目的地代理程式是 Connect:Direct 橋接器代理程式，則您必須同時指定 **-de** 參數，值為 `overwrite`。

當目的地代理程式是通訊協定橋接器代理程式時，不支援 **-dp** 參數。

如果您想要對代理程式呼叫您已定義的任何 Managed File Transfer 傳送 I/O 使用者結束程式，請不要在傳送中指定 **-dp** 參數。使用 **-dp** 參數會導致無法對目的地呼叫傳送 I/O 使用者結束程式，並表示改用標準 Managed File Transfer I/O。

-dq destination_queue

選用項目。檔案傳送至其中的目的地佇列名稱。此規格中可以選擇性地併入佇列管理程式名稱，使用的格式是 QUEUE@QUEUEMANAGER。如果您沒有指定佇列管理程式名稱，則會使用目的地代理程式佇列管理程式名稱。您必須指定佇列管理程式上存在的有效佇列名稱。

當目的地代理程式是通訊協定橋接器代理程式或 Connect:Direct 橋接器代理程式時，或當來源規格是佇列時，不支援 **-dq** 參數。

如果您想要對代理程式呼叫您已定義的任何 Managed File Transfer 傳送 I/O 使用者結束程式，請不要在傳送中指定 **-dq** 參數。使用 **-dq** 參數會導致無法對目的地呼叫傳送 I/O 使用者結束程式，並表示改用標準 Managed File Transfer I/O。

-dqp persistent

選用項目。指定寫入目的地佇列的訊息是否持續保存。有效選項如下所示：

true

將持續性訊息寫入目的地佇列。這是預設值。

false

將非持續性訊息寫入目的地佇列。

qdef

持續性值取自目的地佇列的 DefPersistence 屬性。

只有在同時指定 **-dq** 參數時，才可指定 **-dqp** 參數。

-dqdb hexadecimal_delimiter

選用項目。指定將一個二進位檔分割為多則訊息時要使用的十六進位定界字元。所有訊息都具有相同的 IBM MQ 群組 ID；群組中的最後一則訊息已設定 IBM MQ LAST_MSG_IN_GROUP 旗標。指定十六進位元組作為定界字元的格式是 xNN，其中 N 是範圍介於 0-9 或 a-f 間的字元。透過指定以逗點區隔的十六進位元組清單（例如：x3e,x20,x20,xbf），您可以將十六進位元組的序列指定為定界字元。

只有在同時指定 **-dq** 參數且以二進位模式進行傳送時，才可指定 **-dqdb** 參數。您只能指定 **-qs**、**-dqdb** 及 **-dqdt** 參數的其中一個。

-dqdt pattern

選用項目。指定將一個文字檔分割為多則訊息時要使用的 Java 正規表示式。所有訊息都具有相同的 IBM MQ 群組 ID；群組中的最後一則訊息已設定 IBM MQ LAST_MSG_IN_GROUP 旗標。將正規表示式指定為定界字元所採用的格式為：以括弧括住正規表示式，即 (*regular_expression*)，或以雙引號括住正規表示式，即 "*regular_expression*"。如需相關資訊，請參閱 [第 554 頁的『MFT 所使用的正規表示式』](#)。

依預設，目的地代理程式限制正規表示式可以比對的字串長度為五個字元。您可以編輯 **maxDelimiterMatchLength** 代理程式內容來變更此行為。如需相關資訊，請參閱 [第 464 頁的『進階代理程式內容』](#)。

只有在同時指定 **-dq** 參數且為 **-t** 參數指定 text 值時，才可指定 **-dqdt** 參數。您只能指定 **-qs**、**-dqdb** 及 **-dqdt** 參數的其中一個。

-dqdp position

選用項目。指定分割檔案時預期的目的地文字及二進位定界字元的位置。只有在同時指定 **-dqdt** 及 **-dqdb** 參數的其中一個時，才可指定 **-dqdp** 參數。

請指定下列其中一個選項：

字首

定界字元應該在每一行開頭。

postfix

定界字元應該在每一行結尾。這是預設選項。

-de destination_file_behavior

選用項目。指定如果目的地系統上存在目的地檔案時要採取的動作。有效選項如下所示：

error

報告錯誤，且不傳送檔案。這是預設值。

overwrite

改寫現有目的地檔案。

如果您指定 **-de** 參數，則不能指定 **-td** 參數，因為這些參數是互斥的。不過，您可以在傳送定義檔中指定目的地檔案存在行為。

-t 傳送類型

選用項目。指定檔案傳送的類型：二進位模式或文字模式。

二進位

傳送檔案中的資料時不經過任何轉換。這是預設值。

text

轉換檔案的字碼頁及行尾字元。您可以使用 **-sce**、**-dce** 或 **-dle** 參數，來指定用於轉換的字碼頁及行尾結束符號。如果未指定 **-sce**、**-dce** 或 **-dle** 參數，則實際執行的轉換視來源代理程式及目的地代理程式的作業系統而定。

▶ **z/OS** 例如，從 Windows 傳送至 z/OS 的檔案，其字碼頁會從 ASCII 轉換為 EBCDIC。檔案從 ASCII 轉換為 EBCDIC 時，行尾字元會從 ASCII 歸位 (CR) 及換行 (LF) 字元配對轉換為 EBCDIC 新行 (NL) 字元。

▶ **z/OS** 如需如何傳送 z/OS 資料集的相關資訊，請參閱在 [z/OS 與分散式系統之間傳送檔案及資料集及在資料集之間傳送](#)。

如果您指定 **-t** 參數，則不能指定 **-td** 參數，因為這些參數是互斥的。但是，您可以在傳送定義檔中指定傳送模式行為。

-dce destination_character_encoding

選用項目。指定在目的地撰寫檔案時使用的字元編碼。此選項僅適用於文字檔，所以必須同時指定 **-t text**。可用於轉換的字碼頁，視目的地代理程式的平台而定。如需可用字碼頁的清單，請參閱第 575 頁的『MFT 的可用字碼頁』主題。

noswaplfnl

依預設，Managed File Transfer 會將 swaplfnl 與支援的 EBCDIC 字集搭配使用。使用 swaplfnl 會變更與 EBCDIC LF 0x25 字元之間的字集對映行為。但是，這有時可能會導致進行不必要的對映。使用 noswaplfnl 可置換此行為。

-dle destination_line_ending

選用項目。指定在目的地撰寫檔案時使用的行尾字元。此選項僅適用於文字檔，所以必須同時指定 **-t text** 參數。有效的選項為：

LF

換行。這是下列平台的預設值：

- ▶ **UNIX** UNIX 平台
- ▶ **z/OS** z/OS UNIX 系統服務檔案

當您對 EBCDIC 檔使用 Managed File Transfer 隨附的標準 EBCDIC 字碼頁時，行尾字元會對映至 NL 字元 (0x15)，而非 LF 字元 (0x25)。

CRLF

歸位後面跟著換行。▶ **Windows** 這是 Windows 的預設值。

▶ **z/OS** 如果傳送的目的地是 z/OS 資料集，則會忽略此選項。

-dtr

選用項目。指定要將長度超過 LRECL 資料集屬性的目的地記錄截斷。如果未指定此參數，則會將記錄換行。此參數僅適用於目的地為資料集的文字模式傳送。

-dfa attributes

選用項目。傳送至在 4690 上執行的 IBM MQ 8.0 Managed File Transfer 代理程式時，此參數用來指定與傳送中的目的地檔案相關聯的檔案屬性清單 (以分號區隔)。您可以為 **-dfa** 參數指定值或不指定值。不包含值的範例：

```
-dfa ATTRIBUTE1;ATTRIBUTE2
```

包含值的範例：

```
-dfa ATTRIBUTE1(VALUE);ATTRIBUTE2(VALUE)
```

一個屬性包含值而另一個屬性不包含值的範例：

```
-dfa ATTRIBUTE1;ATTRIBUTE2(VALUE)
```

您可以在指令中使用多個 **-dfa** 參數。

如需 4690 上檔案屬性的相關資訊，請參閱 IBM Documentation 中的 [檔案配送屬性](#)。

安全的參數

-mquserid (userID)

選用項目。指定用來向指令佇列管理程式進行鑑別的使用者 ID。

-mqpassword (password)

選用項目。指定用來向指令佇列管理程式進行鑑別的密碼。您還必須指定 **-mquserid** 參數。如果指定了 **-mquserid**，卻未指定 **-mqpassword**，則系統會提示您提供相關聯的密碼。將不會顯示密碼。

指定來源的參數

-sd source_file_disposition

選用項目。指定當來源檔案順利傳送至其目的地時，對「檔案轉為檔案」或「檔案轉為訊息」傳送中的來源檔案所採取的動作。有效選項如下所示：


leave

來源檔案維持不變。這是預設值。

delete

順利傳送來源檔案之後，即從來源系統中刪除來源檔案。

註：若為「訊息轉為檔案」傳送，一旦順利傳送來源佇列上的訊息，一律會刪除這些訊息。這表示如果 **-sd** 參數設為 **leave** 以進行「訊息轉為檔案」傳送，則會忽略該值。

 在 z/OS 上，如果來源是磁帶資料集，且您指定 **delete** 選項，則會重新裝載磁帶來刪除此資料集。此行為是因為系統環境行為而造成。

如果來源是佇列，且您指定 **leave** 選項，則指令會傳回錯誤，不會要求傳送。

如果來源代理程式是 Connect:Direct 橋接器代理程式，且您指定 **delete** 選項，則行為不同於平常的來源處置行為。發生下列其中一種情況：

- 如果 Connect:Direct 使用 Managed File Transfer 產生的程序，從來源中移動檔案或資料集，則指定 **delete** 選項會造成傳送失敗。若要指定刪除來源檔案，您必須提交使用者定義的 Connect:Direct 程序。如需相關資訊，請參閱第 200 頁的『[透過檔案傳送要求提交使用者定義的 Connect:Direct 程序](#)』。
- 如果 Connect:Direct 利用使用者定義的程序從來源中移出檔案或資料集，則會透過 **%FTEFDISP** 內建符號變數將此參數傳遞給程序。使用者定義的程序決定是否刪除來源。傳送所傳回的結果，取決於使用者定義的程序所傳回的結果。

如果您指定 **-sd** 參數，則不能指定 **-td** 參數，因為這些參數是互斥的。但是，您可以在傳送定義檔中指定來源處置行為。

-r

選用項目。當 *source_specification* 包含萬用字元時，遞迴地傳送子目錄中的檔案。如果為 Managed File Transfer 提供萬用字元作為 *source_specification*，則只有在指定 **-r** 參數時，才會傳送任何符合萬用字元的子目錄。在 *source_specification* 與子目錄相符時，該目錄及其子目錄中的所有檔案（包括隱藏式檔案）都一律會傳送。

如需 Managed File Transfer 如何處理萬用字元的相關資訊，請參閱 [使用萬用字元](#)

如果您指定 **-r** 參數，則不能指定 **-td** 參數，因為這些參數是互斥的。不過，您可以在傳送定義檔中指定遞迴行為。

-sce source_character_encoding

選用項目。指定執行字元轉換時用來讀取來源檔案的字元編碼。此選項僅適用於文字檔，所以必須同時指定 **-t text**。可用於轉換的字碼頁，視目的地代理程式的平台而定，因為轉換是在目的地系統上執行。如需可用字碼頁的清單，請參閱第 575 頁的『MFT 的可用字碼頁』主題。

noswaplfnl

依預設，Managed File Transfer 會將 `swaplfnl` 與支援的 EBCDIC 字集搭配使用。使用 `swaplfnl` 會變更與 EBCDIC LF 0x25 字元之間的字集對映行為。但是，這有時可能會導致進行不必要的對映。使用 `noswaplfnl` 可置換此行為。

z/OS

-skeep

選用項目。指定在從固定長度格式的記錄導向檔案（例如，z/OS 資料集）讀取的來源記錄中，將尾端空格保留為文字模式傳送的一部分。如果未指定此參數，則會刪除來源記錄中的尾端空格。

z/OS

-srdb delimiter

選用項目。若為記錄導向的來源檔案（例如，z/OS 資料集），請指定一個以上要插入的位元組值，作為記錄附加至二進位檔時的定界字元。您必須將每一個值指定為兩個十六進位數字，範圍介於 00-FF 之間，並以 `x` 作為字首。以逗點區隔多個位元組。例如：

```
-srdb x0A
```

or

```
-srdb x0D,x0A
```

您必須在二進位模式中配置傳送。

z/OS

-srdp position

選用項目。指定要插入來源記錄定界字元的位置。只有在同時指定 **-srdb** 參數時，才可指定 **-srdp** 參數。

請指定下列其中一個選項：

字首

在每一筆記錄的開頭插入定界字元。

postfix

在每一筆記錄的結尾插入定界字元。這是預設選項。

-sq

選用項目。指定傳送的來源是佇列。

如果您想要對代理程式呼叫您已定義的任何 Managed File Transfer 傳送 I/O 使用者結束程式，請不要在傳送中指定 **-sq** 參數。使用 **-sq** 參數會造成無法對來源呼叫傳送 I/O 使用者結束程式，同時表示會改用標準 Managed File Transfer I/O。

-sqgi

選用項目。指定訊息依 IBM MQ 群組 ID 進行分組。第一個完整群組會寫入目的地檔案。如果未指定此參數，則來源佇列上的所有訊息會寫入目的地檔案。

只有在同時指定 **-sq** 參數時，才可指定 **-sqgi** 參數。

-sqdt text_delimiter

選用項目。指定在將多則訊息附加到文字檔時，要作為定界字元插入的文字序列。您可以在定界字元中針對字串文字併入 Java ESC 序列。例如，`-sqdt \u007d\n`。

文字定界字元會使用傳送的來源編碼而編碼為二進位格式。每則訊息皆會以二進位格式進行讀取。編碼的定界字元會以二進位格式附加到訊息前面或後面(如 **-sqdp** 參數所指定)，結果會以二進位格式傳送至目的地代理程式。如果來源代理程式字碼頁包括移入及移出狀態，則該代理程式會假設每則訊息的訊息結尾都為移出狀態。在目的地代理程式上，會以與檔案相同的方式，將二進位資料轉換為檔案文字傳送。

只有在同時指定 **-sq** 參數且為 **-t** 參數指定 text 值時，才可指定 **-sqdt** 參數。

-sqdb hexadecimal_delimiter

選用項目。指定在將多則訊息附加到二進位檔時，要作為定界字元插入的一個以上位元組值。每一個值必須指定為兩個十六進位數字，範圍為 00-FF，字首為 x。若有多個位元組，必須以逗點區隔。例如，`-sqdb x08,xA4`。

只有在同時指定 **-sq** 參數時，才可指定 **-sqdb** 參數。如果同時為 **-t** 參數指定 text 值，則無法指定 **-sqdb** 參數。

-sqdp position

選用項目。指定來源文字及二進位定界字元的插入項目的位置。只有在同時指定 **-sqdt** 及 **-sqdb** 參數的其中一個時，才可指定 **-sqdp** 參數。

請指定下列其中一個選項：

字首

在每一個訊息的開頭插入定界字元。

postfix

在每一個訊息的結尾插入定界字元。這是預設選項。

-sqwt wait_time

選用項目。指定等待下列其中一個條件成立的時間（以秒為單位）：

- 新訊息出現在佇列上
- 完整群組出現在佇列上（如果已指定 **-sqgi** 參數）

如果在 *wait_time* 指定的時間內，這兩個條件都不符合，來源代理程式即會停止從佇列進行讀取並完成傳送。如果未指定 **-sqwt** 參數，則在以下情況下來源代理程式會立即停止從來源佇列進行讀取：來源佇列為空白或者已指定 **-sqgi** 參數，佇列上沒有完整群組。

如需使用 **-sqwt** 參數的相關資訊，請參閱第 574 頁的『指定「訊息轉為檔案」傳送等待時間的指引』。

只有在同時指定 **-sq** 參數時，才可指定 **-sqwt** 參數。

source_specification

一個以上檔案規格，決定檔案傳送的一個以上來源。

如果您指定其中一個 **-df**、**-dd**、**-dp**、**-dq** 或 **-ds** 參數，則此為必要項目。如果指定 **-td** 參數，請不要指定 *source_specification*。

- 如果未指定 **-sq** 參數，則 *source_specification* 是一個以上檔案規格，決定檔案傳送的一個以上來源。檔案規格可接受五種格式，也可包括萬用字元。如需萬用字元的相關資訊，請參閱第 549 頁的『搭配使用萬用字元與 MFT』。您可以在檔案規格中使用兩個星號字元 (**)，以跳出檔案規格中的星號。

您可以指定以空格字元分隔的多個來源檔規格。不過，如果對 **-df** 或 **-ds** 參數指定多個來源規格，同時指定 **-de overwrite**，則目的地只包含您最後指定的來源檔案的資料。如果未指定 **-de overwrite**，則傳送只會局部成功。如果目的地檔案先前不存在，它將包含您先指定之來源檔案的資料。

若要傳送其檔名中包含空格的檔案（如 a b.txt 至 c d.txt 檔），請以雙引號括住包含空格的檔名。在 **fteCreateTransfer** 指令中，指定下列文字：

```
-df "c d.txt" "a b.txt"
```

每一個檔案規格必須為下列其中一個種類：

檔案名稱

檔案的名稱，以來源代理程式執行所在系統的適當表示法表達。在檔名指定為來源檔案規格時，會複製檔案的內容。

目錄

目錄的名稱，以來源代理程式執行所在系統的適當表示法表達。在目錄指定為來源檔案規格時，會複製目錄的內容。更準確地說，會複製目錄中及其所有子目錄中的所有檔案，包括隱藏式檔案。

例如，若要僅將 DIR1 的內容複製到 DIR2，請指定 `fteCreateTransfer ... -dd DIR2 DIR1/*`

z/OS

循序資料集

循序資料集的名稱或分割的資料集成員。在資料集名稱之前加上兩個正斜線字元 (//) 來表示資料集。

如果指定通訊協定橋接器代理程式作為來源代理程式，則不能指定資料集作為來源檔案規格。

z/OS

分割的資料集

分割資料集的名稱。在資料集名稱之前加上兩個正斜線字元 (//) 來表示資料集名稱。

如果指定通訊協定橋接器代理程式作為來源代理程式，則不能指定資料集作為來源檔案規格。

Connect:Direct 節點上的檔案名稱或目錄

（僅限 Connect:Direct 橋接器代理程式）。Connect:Direct 節點的名稱、冒號字元 (:), 以及管理 Connect:Direct 節點的系統上的檔案或目錄路徑。例如，`connect_direct_node_name:file_path`。

如果來源代理程式是 Connect:Direct 橋接器代理程式，則僅接受此格式的來源規格。

註：如果來源代理程式是 Connect:Direct 橋接器代理程式，檔案路徑中不支援萬用字元。

通訊協定檔案伺服器上的檔案名稱或目錄

通訊協定檔案伺服器的名稱、冒號字元 (:), 以及通訊協定伺服器系統上的檔案或目錄路徑。例如，`protocol_server:file_path`。

如果未指定通訊協定伺服器，則會使用預設通訊協定伺服器。

- 如果指定 **-sq** 參數，則 *source_specification* 為來源代理程式佇列管理程式上的本端佇列名稱。您只能指定一個來源佇列。以下列格式指定來源佇列：

```
QUEUE_NAME
```

來源佇列規格中不包括佇列管理程式名稱，因為佇列管理程式與來源代理程式佇列管理程式必須相同。

- **z/OS** 如果來源代理程式位於 z/OS 上，則以 // 開頭的原始檔假設為 z/OS 分割的資料集。

其他參數

-? 或 -h

選用項目。顯示指令語法。

範例

在此基本範例中，originalfile.txt 檔從 AGENT1 傳送至相同系統上的 AGENT2，然後重新命名為 transferredfile.txt

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

在此範例中，originalfile.txt 及 originalfile2.txt 檔案從 AGENT1 傳送至相同系統上的 AGENT2，再傳送至 C:\import 目錄

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -dd C:\import C:\export\originalfile.txt  
C:\export\originalfile2.txt
```

在此範例中，originalfile.txt 檔案從 AGENT1 系統傳送至 AGENT2 系統。根據來源代理程式系統的系統時間，排定在 09:00 進行檔案傳送，且每兩小時進行四次：

```
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
```

在此範例中，originalfile.txt 檔從 AGENT1 傳送至 AGENT2，條件為 A.txt 檔必須存在於 AGENT1 上：

```
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 在此範例中，originalfile.txt 檔從 AGENT1 系統傳送至 AGENT2 系統上的資料集 //'USERID.TRANS.FILE.TXT'。同時還會選取文字模式以將資料從 ASCII 轉換為 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 在此範例中，會將 AGENT1 系統上的完整資料集的成員傳送至 AGENT2 系統上的檔案。同時還會選取文字模式，以將檔案從 EBCDIC 轉換為 AGENT2 系統的預設字碼頁。

```
fteCreateTransfer -t text -sa AGENT1 -da AGENT2 -df /tmp/IEEUJV.txt "'SYS1.SAMPLIB(IEEUJV)'"
```

在此範例中，將會使用目的地代理程式 BRIDGE1，將代理程式 AGENT1 上稱為 file.bin 的檔案，傳送至通訊協定檔案伺服器 accountshost.ibm.com 上稱為 file.bin 的目的地檔案。

```
fteCreateTransfer -sa AGENT1 -da BRIDGE1 -df accountshost.ibm.com:/tmp/file.bin /tmp/file.bin
```

在此範例中，會使用未加上引號的萬用字元。AGENT1 的現行工作目錄中所有以 .txt 結尾的檔案，都會傳送到 AGENT2 上的 C:\import 目錄。檔名會維持不變。

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -dd C:\import *.txt
```

在此範例中，會使用加上雙引號的萬用字元。AGENT1 的傳送根目錄中所有以 .txt 結尾的檔案，都會傳送到 AGENT2 上的 C:\import 目錄。檔名會維持不變。

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -dd C:\import "*.txt"
```

回覆碼

回覆碼	說明
0	指令已順利完成。
1	指令未順利結束。
2	指令結束時逾時。指令已傳送訊息至代理程式，但代理程式未在指定的時間內回應。
20	指令局部順利完成，部分檔案已傳送。
21	在判斷傳送結果之前， fteCreateTransfer 指令所連接的佇列管理程式已停止。
40	失敗。未傳送任何指定的檔案。
41	傳送已取消。
42	未執行傳送，因為傳送為條件式，且必要條件不符。
43	傳送要求訊息形態異常。
44	來源代理程式沒有足夠的容量可執行傳送。
45	目的地代理程式沒有足夠的容量可執行傳送。
46	正在傳送的檔案數超出來源代理程式的限制。
47	已傳送的檔案數超出目的地代理程式的限制。

fteDefine (產生配置 Script)

使用 **fteDefine** 指令來產生必要的配置 Script，以定義指定的「代理程式佇列管理程式」物件。

用途

當您必須在相對於配置資料所在系統的遠端系統上執行某些配置步驟時，您會需要使用 **fteDefine** 指令。例如，在佇列管理程式上為代理程式配置佇列，以便可以透過用戶端連線來存取。

語法

<pre>fteDefine ► fteDefine -t type -d outputDir name ◄</pre>
--

參數

-t type

必要項目。要定義的物件類型。類型的選項為 **agent**。

-d outputDir

選用項目。用於寫入 Script 的目錄路徑。如果未提供，便會將 Script 寫入標準輸出串流。

名

必要項目。要定義之物件的一或多個名稱。若要為多個物件指定名稱，請以空格加以區隔。例如：
name1 name2...

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，**fteDefine** 指令與 **-t agent** 參數及單一代理程式名稱一起指定。輸出會寫入檔案。

```
fteDefine -t agent EXAMPLE.AGENT >EXAMPLE.AGENT_create.mqsc
```

此指令所產生的輸出，是要對代理程式佇列管理程式執行，以建立所需代理程式佇列的 MQSC 指令 Script：

```
$ 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.
```

在此範例中，**fteDefine** 指令與 **-d outputDir** 參數及數個代理程式名稱一起指定。

```
fteDefine -t agent -d /tmp EXAMPLE.AGENT.1 EXAMPLE.AGENT.2 EXAMPLE.AGENT.3
```

此指令所產生的輸出是 MQSC 指令 Script 位置的絕對檔案路徑：

```
$ 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'.
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

相關參考

第 388 頁的『[fteDelete](#) (產生可以移除配置的 Script)』

使用 **fteDelete** 指令來產生必要的配置 Script，以移除指定的「代理程式佇列管理程式」物件。

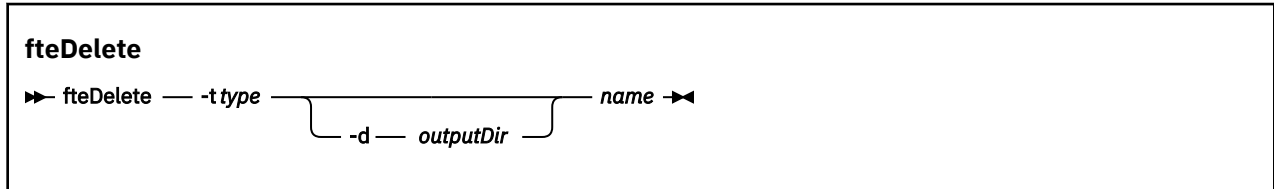
fteDelete (產生可以移除配置的 Script)

使用 **fteDelete** 指令來產生必要的配置 Script，以移除指定的「代理程式佇列管理程式」物件。

用途

當您必須在相對於配置資料所在系統的遠端系統上執行某些配置步驟時，您會需要使用 **fteDelete** 指令。例如，在本端佇列管理程式上移除遠端用戶端代理程式的佇列。

語法



參數

-t type

必要項目。要刪除的物件類型。類型的選項為 **agent**。

-d outputDir

選用項目。用於寫入 Script 的目錄路徑。如果未提供，便會將 Script 寫入標準輸出串流。

名

必要項目。要刪除之物件的一或多個名稱。若要為多個物件指定名稱，請以空格加以區隔。例如：
name1 name2...

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，**fteDelete** 指令與 **-t agent** 參數及單一代理程式名稱一起指定。輸出會寫入檔案。

```
fteDelete -t agent EXAMPLE.AGENT >EXAMPLE.AGENT_delete.mqsc
```

此指令所產生的輸出，是要對代理程式佇列管理程式執行，以刪除代理程式佇列的 MQSC 指令 Script：

```
$ 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.
```

在此範例中，**fteDelete** 指令與 **-d outputDir** 參數及數個代理程式名稱一起指定。

```
fteDelete -t agent -d /tmp EXAMPLE.AGENT.1 EXAMPLE.AGENT.2 EXAMPLE.AGENT.3
```

此指令所產生的輸出是 MQSC 指令 Script 位置的絕對檔案路徑：

```
$ 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'.
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

相關參考

第 386 頁的『[fteDefine \(產生配置 Script\)](#)』

使用 **fteDefine** 指令來產生必要的配置 Script，以定義指定的「代理程式佇列管理程式」物件。

fteDeleteAgent

fteDeleteAgent 指令會刪除 Managed File Transfer Agent 及其配置。如果代理程式為通訊協定橋接器代理程式，則使用者認證檔會留在檔案系統中。

用途

在執行 **fteDeleteAgent** 指令之前，請使用 [fteStopAgent](#) 指令停止代理程式。

如果已將代理程式配置為以 Windows 服務方式執行，則執行 **fteDeleteAgent** 指令會刪除服務定義。

V 9.0.4

從 IBM MQ 9.0.4 開始，刪除代理程式時將會移除所有資源監視器及排定的傳送。

只有 IBM MQ 管理者（及 mqm 群組的成員）身分的使用者才能執行此指令。如果您嘗試以非 IBM MQ 管理者的使用者身分執行此指令，則會收到錯誤訊息，指令也不會執行。

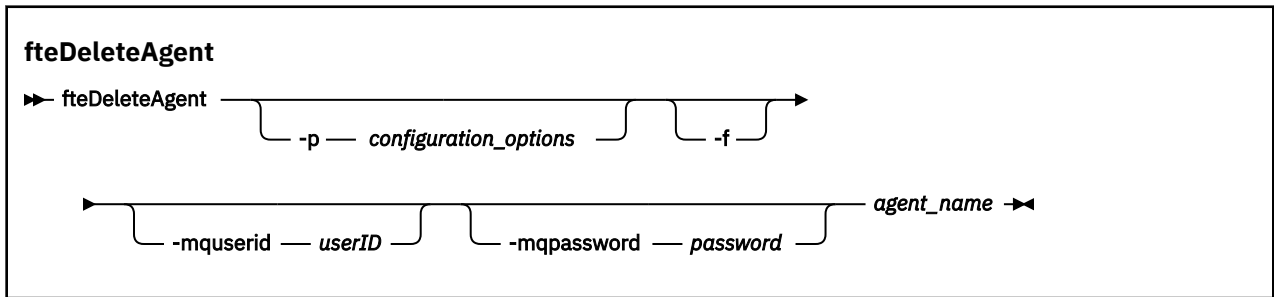
fteDeleteAgent 指令會提供一些 MQSC 指令，必須針對代理程式的佇列管理程式執行這些指令，才能清除及刪除代理程式的系統佇列。這些佇列如下：

- 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
- 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 指令也會在以下位置的檔案中提供這些指令：

```
MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name/agent_name_delete.mqsc
```

語法



參數

-p (*configuration_options*)

選用項目。如果具有多個協調佇列管理程式，則使用此參數可明確指定要刪除的代理程式配置。請使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，指令即會使用與此非預設協調佇列管理程式相關聯的配置選項。

只有在您想要使用不同於預設值的配置選項時，才需指定選用的 **-p** 參數。如果您沒有指定 **-p**，則會使用 `installation.properties` 檔案中定義的配置選項。如需相關資訊，請參閱第 49 頁的『多平台上的 MFT 配置選項』。

-f

選用項目。強制指令從協調佇列管理程式取消登錄代理程式，即使找不到代理程式的配置檔亦是如此。因為代理程式的佇列管理程式的相關資訊在這種狀況下無法使用，所以此指令會直接連接至協調佇列管理程式，而非像通常一樣使用代理程式佇列管理程式。

-mquserid (*userID*)

選用項目。除非有 **-f** 強制參數，否則指定用來向代理程式佇列管理程式進行鑑別的使用者 ID。如果有 **-f** 參數，它會指定用來向協調佇列管理程式進行鑑別的使用者 ID。

-mqpassword (*password*)

選用項目。除非有 **-f** 強制參數，否則指定用來向代理程式佇列管理程式進行鑑別的密碼。如果有 **-f** 參數，它會指定用來向協調佇列管理程式進行鑑別的密碼。您還必須指定 **-mquserid** 參數。如果指定了 **-mquserid**，卻未指定 **-mqpassword**，則系統會提示您提供相關聯的密碼。將不會顯示密碼。

agent_name

必要項目。要刪除的代理程式的名稱。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，會刪除 AGENT3 及其在協調佇列管理程式 QM_COORD1 上的配置：

```
fteDeleteAgent -p QM_COORD1 AGENT3
```

此範例指令會輸出下列 MQSC 指令來刪除代理程式的三個佇列：

```
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)
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

相關參考

第 449 頁的『[fteStopAgent](#)』

使用 **fteStopAgent** 指令，以受管制的方式停止 Managed File Transfer 代理程式，或在必要時使用 **-i** 參數立即停止代理程式。

第 322 頁的『[fteCleanAgent: 清除 MFT 代理程式](#)』

使用 **fteCleanAgent** 指令，可透過從代理程式所使用的持續性及非持續性佇列中刪除訊息，來清除 Managed File Transfer Agent 使用的佇列。如果因資訊保留在代理程式使用的佇列中，而可能導致您在啟動代理程式時遇到問題，請使用 **fteCleanAgent** 指令。

第 326 頁的『[fteCreateAgent \(建立 MFT 代理程式\)](#)』


fteCreateAgent 指令會建立 Managed File Transfer Agent 及其相關聯的配置。

第 445 頁的『[fteStartAgent](#)』



fteStartAgent 指令會從指令行啟動 Managed File Transfer 代理程式。

fteDeleteLogger

使用 **fteDeleteLogger** 指令可刪除 Managed File Transfer 日誌程式及其配置。與日誌程式相關聯的現有日誌檔可以保留，也可以一併刪除。

重要:  在 IBM MQ for UNIX, Linux, and Windows 上，只有 IBM MQ 管理者（及 mqm 群組的成員）身分的使用者才能執行此指令。如果您嘗試以非 IBM MQ 管理者的使用者身分執行此指令，則會收到錯誤訊息 BFGCL0502E: 您未獲授權執行所要求的作業。，且指令將不會執行。

 在 z/OS 系統上，使用者必須滿足（至少）下列其中一項條件才能執行移轉指令：

- 是 mqm 群組的成員（如果 mqm 群組已定義在系統上）。
-  是 BFG_GROUP_NAME 環境變數所指名之群組的成員（如果已指名群組的話）。
-  指令執行時，BFG_GROUP_NAME 環境變數中未設定任何值。

用途

在執行 **fteDeleteLogger** 指令之前，使用 **fteStopLogger** 指令停止日誌程式。

如果已將日誌程式配置為以 Windows 服務方式執行，則執行 **fteDeleteLogger** 指令會刪除服務定義。

日誌程式配置目錄中包含可用於刪除日誌程式佇列及訂閱的 MQSC Script。這些佇列如下：

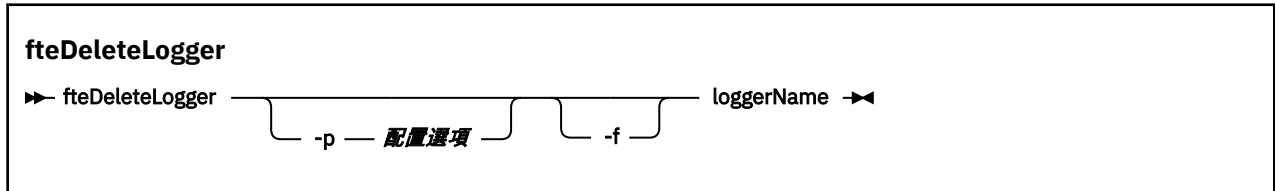
- SYSTEM.FTE.LOG.CMD.*logger_name*
- SYSTEM.FTE.LOG.RJCT.*logger_name*

訂閱名稱如下：

- SYSTEM.FTE.AUTO.SUB.*logger_name*

MQSC Script 位於
MQ_DATA_PATH\mqft\config\coordination_qmgr\loggers\logger_name\logger_name_delete.mqsc 中

語法



參數

-p (configuration_options)

選用項目。決定一組用來啟動獨立式資料庫日誌程式的配置選項。使用一組配置選項的名稱作為 **-p** 參數的值。依照慣例，此值是協調佇列管理程式的名稱。如果未指定此參數，則會使用一組預設的配置選項。

-f

選用項目。強制移除由此日誌程式建立的任何日誌檔。如果省略此參數，則會保留該日誌程式建立的任何日誌檔；如果不需要這些日誌檔時，必須手動移除。

logger_name

必要項目。要刪除的日誌程式的名稱。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，會刪除名為 logger1 的日誌程式。已指定 **-f** 參數，這會導致移除日誌程式的日誌檔，以及日誌程式的配置檔。

```
fteDeleteLogger -f logger1
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

相關參考

第 343 頁的『[fteCreate 日誌程式 \(建立 MFT 檔案或資料庫日誌程式\)](#)』

使用 **fteCreateLogger** 指令來建立 Managed File Transfer 檔案或資料庫日誌程式。

第 447 頁的『[fteStartLogger](#)』

fteStartLogger 指令會啟動 Managed File Transfer 記載應用程式。

第 450 頁的『[fteStopLogger](#)』

fteStopLogger 指令會停止 Managed File Transfer 日誌程式。

第 415 頁的『[fteModify 日誌程式 \(以 Windows 服務方式執行 MFT 日誌程式\)](#)』

使用 **fteModifyLogger** 指令來修改 Managed File Transfer 日誌程式，以便它可以 Windows 服務方式執行。您只能在 Windows 平台上使用此指令，必須由 IBM MQ 管理者及 mqm 群組成員的使用者執行，且必須先使用 **fteStopLogger** 指令停止日誌程式。

fteDeleteMonitor: 刪除 MFT 資源監視器

利用指令行，使用 **fteDeleteMonitor** 指令來停止並刪除現有 Managed File Transfer 資源監視器。針對資源監視代理程式發出此指令。

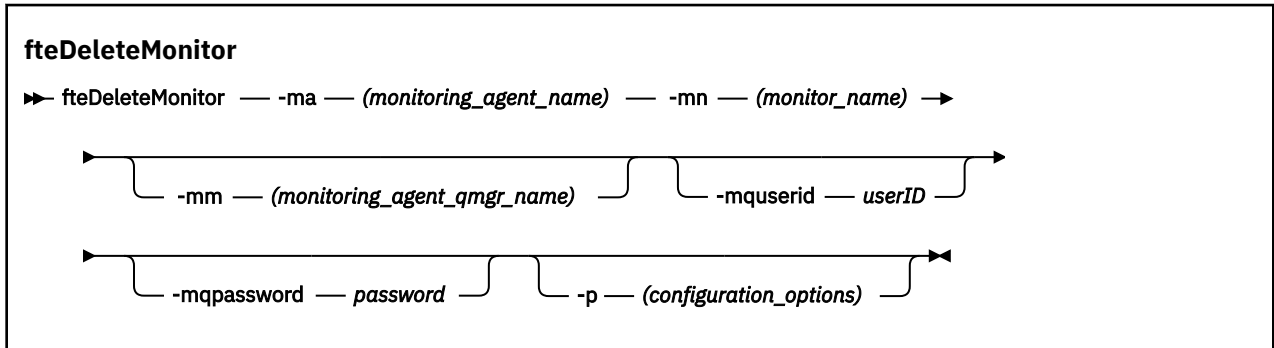
用途

使用 **fteDeleteMonitor** 指令可停止資源監視，並從監視代理程式移除監視定義。執行此指令時，不會再輪詢資源，也不會啟動後續作業。

您可以從任何可以連接至 IBM MQ 網路的系統執行 **fteDeleteMonitor** 指令，然後遞送至代理程式的佇列管理程式。具體而言，為了讓指令能夠執行，您必須在此系統上安裝 Managed File Transfer 元件（「服務」或「代理程式」），且必須已配置此系統的 Managed File Transfer，才能與 IBM MQ 網路進行通訊。如果沒有可用的連線功能詳細資料，則只要有可用的代理程式佇列管理程式詳細資料，即會改用這些詳細資料進行連線。

僅在要使用一組不同於您的預設集的配置選項時，才為此指令指定選用的 **-p** 參數。如需相關資訊，請參閱 [配置選項](#)。

語法



參數

-ma (monitoring_agent_name)

必要項目。執行資源監視的代理程式的名稱。此監視代理程式還必須是您要觸發的檔案傳送的來源代理程式。

-mn (monitor_name)

必要項目。指派給此資源監視器的名稱。您可以刪除資源監視器，然後以相同名稱建立新監視器。

-mm (monitoring_agent_qmgr_name)

選用項目。監視代理程式的佇列管理程式的名稱。因為監視代理程式與監視器所觸發的傳送來源代理程式必須相同，所以此佇列管理程式也是來源代理程式的佇列管理程式。

-mquserid (userID)

選用項目。指定用來向指令佇列管理程式進行鑑別的使用者 ID。

-mqpassword (password)

選用項目。指定用來向指令佇列管理程式進行鑑別的密碼。您還必須指定 **-mquserid** 參數。如果指定了 **-mquserid**，卻未指定 **-mqpassword**，則系統會提示您提供相關聯的密碼。將不會顯示密碼。

-p (configuration_options)

選用項目。此參數會決定用於取消傳送的配置選項集。依照慣例，會使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，指令即會使用與此非預設協調佇列管理程式相關聯的一組內容檔。

如果未指定此參數，則會使用一組以預設協調佇列管理程式為基礎的配置選項。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，會刪除監視（及檔案傳送來源代理程式）AGENT1 的資源監視器 MONITOR1：

```
fteDeleteMonitor -ma AGENT1 -mm QM_JUPITER -mn MONITOR1
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

相關工作

第 141 頁的『監視 MFT 資源』

您可以監視 Managed File Transfer 資源；例如，佇列或目錄。滿足此資源的條件時，資源監視器會啟動作業，例如檔案傳送。您可以在 IBM MQ Explorer 的 Managed File Transfer 外掛程式中使用 **fteCreateMonitor** 指令或「監視器」視圖來建立資源監視器。

相關參考

第 348 頁的『fteCreateMonitor: 建立 MFT 資源監視器』

fteCreateMonitor 指令可從指令行建立並啟動新的資源監視器。您可以使用 Managed File Transfer 來監視資源（例如目錄的內容），以在符合觸發條件時，啟動指定的作業（例如檔案傳送）。

第 401 頁的『fteListMonitors: 列出 MFT 資源監視器』

使用 **fteListMonitors** 指令，可讓您透過指令行列出 Managed File Transfer 網路中的所有現有資源監視器。

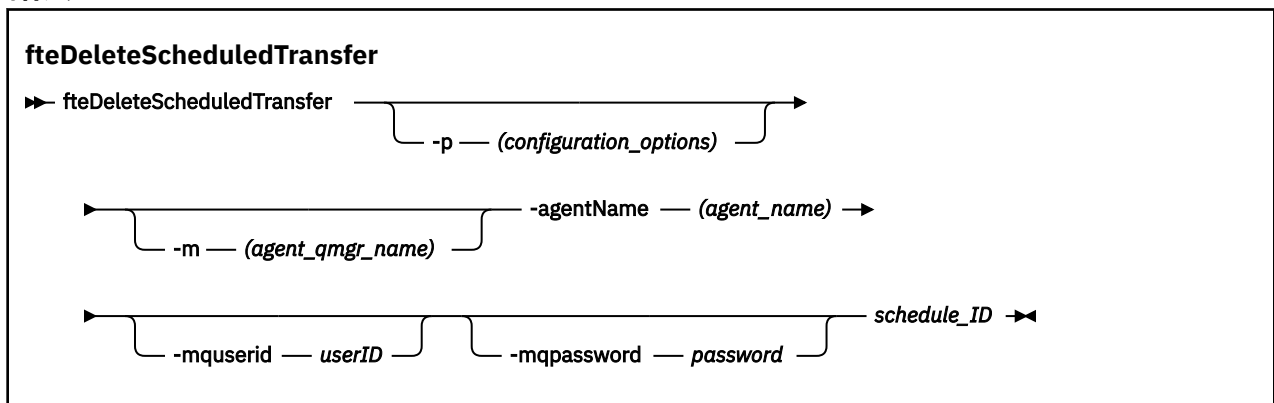
fteDeleteScheduledTransfer

用途

使用 **fteDeleteScheduledTransfer** 指令來刪除您先前使用指令行或 IBM MQ Explorer 建立的 Managed File Transfer 排程傳送。

只有在您想要使用不同於預設值的配置選項時，才為此指令指定選用的 **-p** 參數。如果您沒有指定 **-p**，則會使用 `installation.properties` 中定義的配置選項。如需相關資訊，請參閱第 49 頁的『多平台上的 MFT 配置選項』。

語法



參數

-p (configuration_options)

選用項目。如果具有多個協調佇列管理程式，則使用此參數可明確指定要刪除的排定傳送。請使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，指令即會使用與此非預設協調佇列管理程式相關聯的配置選項。

如果您未指定此參數，則會使用以預設協調佇列管理程式為基礎的配置選項。

-m (agent_qmgr_name)

選用項目。來源代理程式所連接的佇列管理程式的名稱。如果未指定此參數，將透過正在使用的配置選項確定代理程式的佇列管理程式。

-agentName (agent_name)

必要項目。要從中刪除排定傳送的來源代理程式的名稱。

-mquserid (userID)

選用項目。指定用來向指令佇列管理程式進行鑑別的使用者 ID。

-mqpassword (password)

選用項目。指定用來向指令佇列管理程式進行鑑別的密碼。您還必須指定 **-mquserid** 參數。如果指定了 **-mquserid**，卻未指定 **-mqpassword**，則系統會提示您提供相關聯的密碼。將不會顯示密碼。

schedule_ID

必要項目。要刪除的排定傳送的 ID。

可以透過針對來源代理程式名稱執行 [fteListScheduledTransfers](#) 指令，來尋找排程 ID。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，會刪除來源代理程式 AGENT2 上 ID 為 27 的排定傳送：

```
fteDeleteScheduledTransfer -agentName AGENT2 27
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

相關工作

第 135 頁的『[建立排定的檔案傳送](#)』

您可以從「IBM MQ Explorer 探險家」或從指令行排定新的檔案傳送。排定的傳送可以包含群組中的單一檔案或多個檔案。您可以執行排定的檔案傳送一次，或多次重複傳送。

相關參考

第 403 頁的『[fteListScheduledTransfers](#)』

使用 **fteListScheduledTransfers** 指令來列出您先前使用指令行或 IBM MQ Explorer 建立的所有 Managed File Transfer 傳送。

fteDeleteTemplates

使用 **fteDeleteTemplates** 指令可從協調佇列管理程式刪除現有 Managed File Transfer 範本。

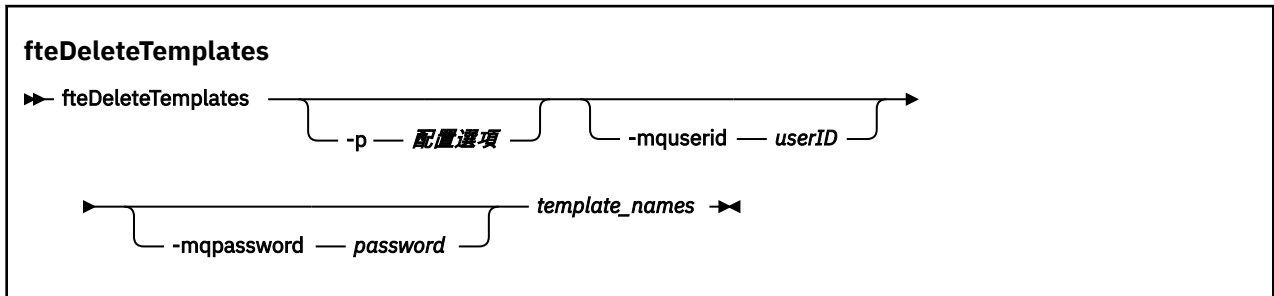
用途

fteDeleteTemplates 指令用於從協調佇列管理程式移除一個以上檔案傳送範本。執行此指令時，會向 IBM MQ 系統傳遞從協調佇列管理程式移除範本的要求，使範本不再可供「IBM MQ 探險家」或指令行使用。在指令完成後的一剎那，可以繼續存取所刪除的範本，直至 IBM MQ 系統處理該要求為止。

您可以從可以連接至 IBM MQ 網路的任何系統執行 **fteDeleteTemplates** 指令，然後遞送至協調佇列管理程式。具體而言，為了讓指令能夠執行，您必須在此系統上安裝 Managed File Transfer，且必須已配置此系統的 Managed File Transfer，才能與 IBM MQ 網路進行通訊。如果沒有可用的連線功能詳細資料，則只要有可用的代理程式佇列管理程式詳細資料，即會改用這些詳細資料進行連線。

僅在要使用一組不同於您的預設集的配置選項時，才為此指令指定選用的 **-p** 參數。如需相關資訊，請參閱配置選項。

語法



參數

-p (configuration_options)

選用項目。此參數決定一組用來刪除範本的配置選項。依照慣例，會使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，指令即會使用與此非預設協調佇列管理程式相關聯的一組內容檔。

如果未指定此參數，則會使用一組以預設協調佇列管理程式為基礎的配置選項。

-mquserid (userID)

選用項目。指定用來向協調佇列管理程式進行鑑別的使用者 ID。

-mqpassword (password)

選用項目。指定用來向協調佇列管理程式進行鑑別的密碼。您還必須指定 **-mquserid** 參數。如果指定了 **-mquserid**，卻未指定 **-mqpassword**，則系統會提示您提供相關聯的密碼。將不會顯示密碼。

(template_names)

必要項目。指定要刪除的一個以上範本名稱。根據 **fteListTemplates** 指令的顯示結果指定名稱。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，會刪除範本 STANDBY：

```
fteDeleteTemplates STANDBY
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

相關工作

第 162 頁的『使用檔案傳送範本』

您可以使用檔案傳送範本來儲存一般檔案傳送設定，以進行重複或複式傳送。請使用

fteCreateTemplate 指令，從指令行建立傳送範本，或者使用 IBM MQ Explorer，利用為受管理檔案傳送建立新範本精靈建立傳送範本，或選取將傳送設定儲存為範本勾選框，在建立檔案傳送時儲存範本。傳送範本視窗會顯示您在 Managed File Transfer 網路中建立的所有傳送範本。

第 163 頁的『使用 IBM MQ Explorer 建立檔案傳送範本』

您可以從「IBM MQ Explorer」或從指令行建立檔案傳送範本。然後，您可以使用該範本，利用範本詳細資料建立新的檔案傳送，或提交範本以啟動檔案傳送。

相關參考

第 354 頁的『fteCreateTemplate: 建立新的檔案傳送範本』

fteCreateTemplate 指令可建立檔案傳送範本，以備將來使用。唯一必要的參數是 **-tn** (*template_name*) 參數。其他所有參數是選用的，雖然如果指定來源檔案規格，您還必須提供目的地檔案。同樣地，如果指定目的地檔案，您還必須提供來源檔案規格。

第 405 頁的『fteListTemplates』

使用 **fteListTemplates** 指令，可列出協調佇列管理程式上的可用 Managed File Transfer 傳送範本。

fteDisplayVersion

使用 **fteDisplayVersion** 指令可顯示已安裝 Managed File Transfer 的版本。

用途

IBM 客戶服務代表可能會要求您執行 **fteDisplayVersion** 指令，以協助進行問題判斷。

語法



參數

-v

選用項目。顯示產品版本的大量相關資訊。

當指定 **-v** 參數時，所顯示的精確的詳細資料會隨產品版次而異。不建議您依賴於 **fteDisplayVersion -v** 指令輸出中所提供的特定資訊。

 在 z/OS 上，如果指定了產品 ID，則 **-v** 會顯示 **productId** 內容的值。

-? 或 -h

選用項目。顯示指令語法。

未指定參數的範例

在此範例中，**fteDisplayVersion** 指令指定為不使用任何參數。

```
fteDisplayVersion
```

此指令的輸出為產品版本層次。例如，這是 IBM MQ 9.0 的輸出：

```
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 -v 參數的範例

在此範例中，**fteDisplayVersion** 指令指定為使用 **-v** 參數。

```
fteDisplayVersion -v
```

此指令的輸出包括產品版本的更詳細資訊。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:   9.0.0.0
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
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

fteListAgents：列出協調佇列管理程式的 MFT 代理程式

使用 **fteListAgents** 指令，以列出向特定協調佇列管理程式登錄的所有 Managed File Transfer 代理程式。

用途

您可以從任何可連接至協調佇列管理程式的系統執行 **fteListAgents** 指令。每一個代理程式的下列詳細資料都會導向至標準輸出裝置 (STDOUT)：

- 代理程式名稱
- 代理程式佇列管理程式
- 如果代理程式是通訊協定橋接器代理程式，則代理程式名稱會附加 **bridge** 。
- 如果代理程式是 Connect:Direct 橋接器代理程式，則代理程式名稱會附加 (Connect:Direct bridge)
- 代理程式狀態

此指令會使用 `coordination.properties` 檔連接至協調佇列管理程式。如需相關資訊，請參閱 [第 456 頁的『MFT coordination.properties 檔』](#)。

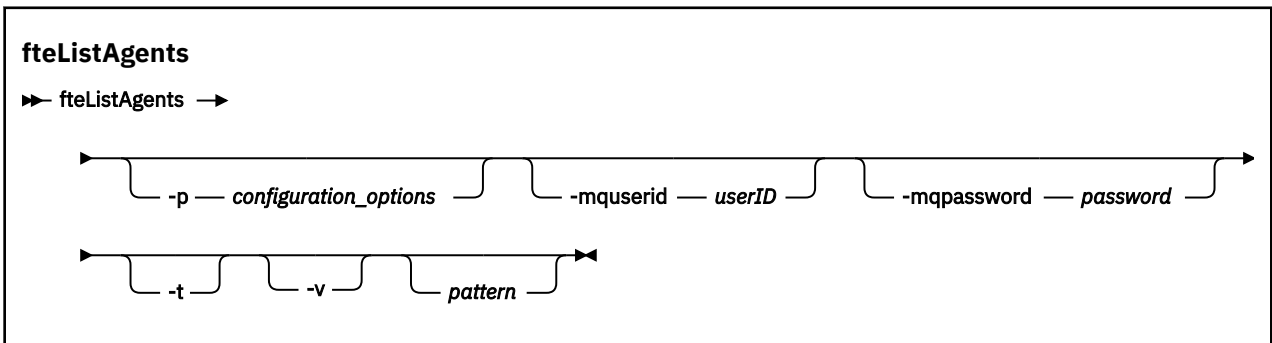
僅在要使用一組不同於您的預設集的配置選項時，才為此指令指定選用的 **-p** 參數。如需相關資訊，請參閱 [第 49 頁的『多平台上的 MFT 配置選項』](#)。

如果 **fteListAgents** 指令未列出代理程式，請使用下列主題中的診斷流程圖來尋找並修正問題：[如果 fteListAgents 指令未列出您的代理程式](#)。

代理程式狀態資訊

此指令產生的代理程式狀態資訊，是從代理程式發佈至 SYSTEM.FTE 主題的狀態訊息中產生的。這些訊息在主題 [第 625 頁的『MFT 代理程式狀態訊息格式』](#) 中有說明。**fteListAgents** 指令所產生的狀態資訊，指出前次發佈狀態訊息時的代理程式狀態。這些狀態訊息的頻率視 `agentStatusPublishRateLimit` 內容值而定。如需此內容的詳細資料，請參閱主題 [第 463 頁的『MFT agent.properties 檔案』](#)。

語法



參數

-p (configuration_options)

選用項目。此參數決定一組用來發出列出代理程式要求的配置選項。請使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，指令即會使用與此非預設協調佇列管理程式相關聯的一組內容檔。

如果未指定此參數，則會使用一組以預設協調佇列管理程式為基礎的配置選項。

-mquserid (userID)

選用項目。指定用來向協調佇列管理程式進行鑑別的使用者 ID。

-mqpassword (password)

選用項目。指定用來向協調佇列管理程式進行鑑別的密碼。您還必須指定 **-mquserid** 參數。如果指定了 **-mquserid**，卻未指定 **-mqpassword**，則系統會提示您提供相關聯的密碼。將不會顯示密碼。

-v

選用項目。指定詳細模式。詳細模式會為每一個代理程式產生其他輸入，包括格式為 Source/Destination 的現行傳送數目，其中 Source 是現行來源傳送數目，Destination 是現行目的地傳送數目。

現行傳送資訊取自代理程式狀態發佈，下列主題提供相關說明：[第 625 頁的『MFT 代理程式狀態訊息格式』](#)。因此，只有在 `agentStatusPublishRateLimit` 代理程式內容值（預設為 30 秒）的設定範圍內，這項傳送資訊才是正確的。

-t

選用項目。指定簡要模式。在 IBM MQ 9.0.2, 依預設, 輸出包括 **Status Age** 直欄。如果您不想看到 **Status Age** 資訊, 您可以發出含有 **-t** 參數的指令來隱藏直欄。如需相關資訊, 請參閱 [如果 fteListAgents 指令顯示代理程式狀態為 UNKNOWN, 怎麼辦。](#)

型樣

選用項目。用來過濾 Managed File Transfer 代理程式清單的型樣。此型樣會比對代理程式名稱。星號 (*) 字元會解譯為符合任何值 (包括零字元) 的萬用字元。

在 UNIX 及 Linux 系統上, 如果您要將星號 (*) 及 # 記號之類的特殊字元當作文字來處理, 就必須以單引號 (') 或雙引號 (" ") 來跳出這些特殊字元。如果未跳出這些字元, 則會根據這些字元在特定 UNIX 或 Linux 系統上的意義進行解譯。

如果未指定此參數, 則會列出所有向協調佇列管理程式登錄的代理程式。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中, 會列出所有已在佇列管理程式上登錄、且在配置選項中名稱以 B 開頭的代理程式:

```
fteListAgents "B*"
```

在此範例中, 會以詳細模式列出已向協調佇列管理程式 QM_EUROPE (非預設協調佇列管理程式) 登錄的代理程式:

```
fteListAgents -p QM_EUROPE -v
```

此指令的輸出如下所示:

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

如需可能的代理程式狀態值及其意義的清單, 請參閱第 525 頁的『MFT 代理程式狀態值』主題。

在此範例中, 會以詳細模式列出所有已向協調佇列管理程式登錄、且名稱以 BRIDGE 開頭的代理程式:

```
fteListAgents -v "BRIDGE*"
```

此指令的輸出如下所示:

```
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
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

相關工作

第 176 頁的『列出 MFT 代理程式』

您可以使用指令行或 IBM MQ Explorer 來列出向特定佇列管理程式登錄的 Managed File Transfer 代理程式。

相關參考

第 525 頁的『MFT 代理程式狀態值』

fteListAgents 及 **fteShowAgentDetails** 指令可產生代理程式狀態資訊。此狀態有數個可能的值。

第 436 頁的『fteShowAgentDetails』

使用 **fteShowAgentDetails** 指令顯示特定 Managed File Transfer Agent 的詳細資料。這些是代理程式的 Managed File Transfer 協調佇列管理程式儲存的詳細資料。

fteListMonitors: 列出 MFT 資源監視器

使用 **fteListMonitors** 指令，可讓您透過指令行列出 Managed File Transfer 網路中的所有現有資源監視器。

用途

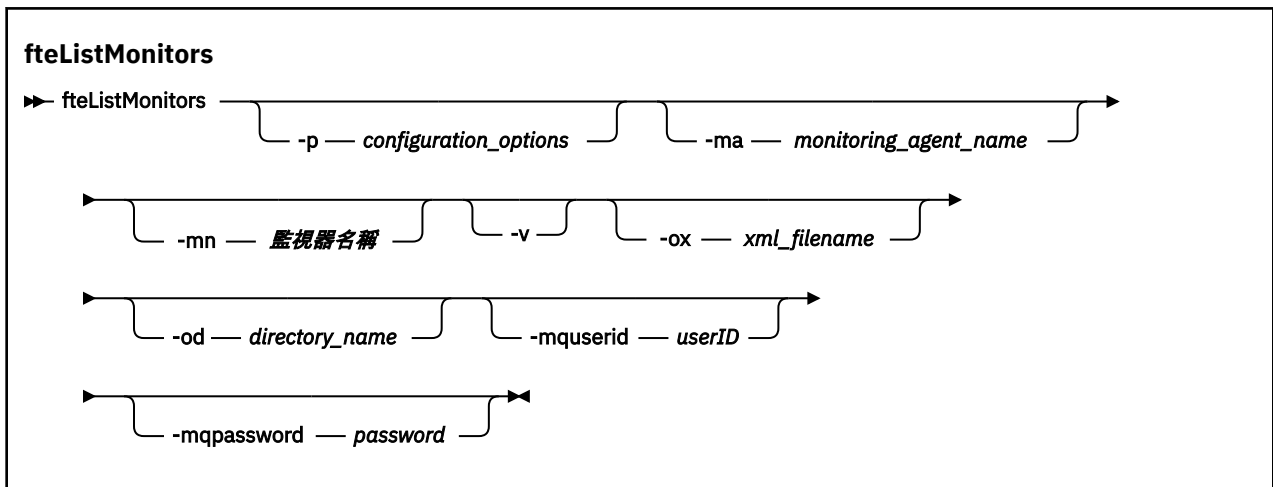
fteListMonitors 指令可列出現有資源監視器。您可以指定代理程式名稱及資源監視器名稱，以過濾指令輸出。

此指令會使用 `coordination.properties` 檔連接至協調佇列管理程式。如需相關資訊，請參閱第 456 頁的『MFT `coordination.properties` 檔』。

您可以使用 **-ox** 參數將資源監視器匯出至 XML 檔。如需如何使用此 XML 檔的相關資訊，請參閱第 348 頁的『fteCreateMonitor: 建立 MFT 資源監視器』。

僅在要使用一組不同於您的預設集的配置選項時，才為此指令指定選用的 **-p** 參數。如需相關資訊，請參閱配置選項。

語法



參數

-p (configuration_options)

選用項目。此參數會決定用於取消傳送的配置選項集。依照慣例，會使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，指令即會使用與此非預設協調佇列管理程式相關聯的內容檔集。

如果未指定此參數，則會使用一組以預設協調佇列管理程式為基礎的配置選項。

-ma (monitoring_agent_name)

選用項目。使用您提供作為輸入的型樣，依代理程式名稱過濾資源監視器。星號(*)字元會被解譯為符合零個以上字元的萬用字元。如果未指定 **-ma** 參數，依預設會列出所有與預設協調佇列管理程式的所有代理程式相關聯的資源監視器。

-mn (monitor_name)

選用項目。使用您提供作為輸入的型樣，依監視器名稱過濾資源監視器。星號 (*) 字元會被解譯為符合零個以上字元的萬用字元。如果未指定 **-mn** 參數，依預設會列出所有與預設協調佇列管理程式的所有代理程式相關聯的資源監視器。

-mquserid (userID)

選用項目。指定用來向協調佇列管理程式進行鑑別的使用者 ID。

-mqpassword (password)

選用項目。指定用來向協調佇列管理程式進行鑑別的密碼。您還必須指定 **-mquserid** 參數。如果指定了 **-mquserid**，卻未指定 **-mqpassword**，則系統會提示您提供相關聯的密碼。不會顯示密碼。

-v

選用項目。產生詳細輸出，內含監視器狀態的其他相關資訊，包括監視器是否已啟動或停止、所監視的目錄資源路徑及觸發條件。

-ox (xml_filename)

選用項目。此參數必須與 **-ma** 及 **-mn** 參數一起指定。將資源監視器匯出至 XML 檔，即可供 **fteCreateMonitor** 指令及 **-ix** 參數使用。

V 9.0.5 **-ox** 參數不得與 **-od** 參數結合使用。

V 9.0.5 -od (directory_name)

選用項目。將多個資源監視器定義匯出至指定目錄。每個資源監視器定義皆會儲存至名稱格式為 *agent name.monitor name.xml* 的個別 XML 檔。您必須為 XML 檔指定有效的目標目錄，否則會顯示錯誤訊息。此參數不得與 **-ox** 參數結合使用。

-? 或 -h

選用項目。顯示指令語法。

範例：列出資源監視器

在此範例中，列出所有與監視代理程式 AGENT1 相關聯的資源監視器（以及與監視器相關聯的檔案傳送來源代理程式）：

```
fteListMonitors -ma AGENT1
```

範例：將一個資源監視器匯出至 XML 檔

在此範例中，透過使用 **-ox** 參數來指定 XML 檔名，以將 AGENT1 上的單一資源監視器 MONITOR1 匯出至 XML 檔 filename1.xml：

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

範例：將一個資源監視器匯出至指定目錄**V 9.0.5**

在此範例中，會將 AGENT1 上的單一資源監視器 MONITOR1 匯出至 **-od** 參數所指定的目錄。除了 XML 檔名稱格式不同以外，此範例類似於使用 **-ox** 參數。

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

範例：將一批資源監視器匯出至指定目錄中的 XML 檔**V 9.0.5**

在下列所有範例中，皆會將資源監視器匯出至 **-od** 參數所指定的目錄。每一個資源監視器定義都會儲存至個別 XML 檔案，並具有格式為 *agent name.monitor name.xml* 的名稱。

在此範例中，會將所有資源監視器匯出至指定目錄：

```
fteListMonitors -od /usr/mft/resmonbackup
```

在此範例中，會將 AGENT1 上的所有資源監視器匯出至指定目錄：

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

在指定符合代理程式名稱及/或監視器名稱的型樣時使用星號字元 (*)，即可使用萬用字元比對來定義要匯出的資源監視器。

在此範例中，會將 AGENT1 上名稱符合型樣 MON* 的所有資源監視器匯出至指定目錄：

```
fteListMonitors -ma AGENT1 -mn MON* -od /usr/mft/resmonbackup
```

在此範例中，會將名稱符合型樣 AGEN* 的代理程式上的所有資源監視器匯出至指定目錄：

```
fteListMonitors -ma AGEN* -od /usr/mft/resmonbackup
```

在此範例中，會將名稱符合型樣 AGENT* 的代理程式上名稱符合型樣 MON* 的所有資源監視器匯出至指定目錄：

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

回覆碼

0

指令已順利完成。

1

指令未順利結束。

相關工作

[第 141 頁的『監視 MFT 資源』](#)

您可以監視 Managed File Transfer 資源；例如，佇列或目錄。滿足此資源的條件時，資源監視器會啟動作業，例如檔案傳送。您可以在 IBM MQ Explorer 的 Managed File Transfer 外掛程式中使用 **fteCreateMonitor** 指令或「監視器」視圖來建立資源監視器。

[第 160 頁的『備份及還原 MFT 資源監視器』](#)

您可以透過下列方式來備份要供未來使用的資源監視器：將其定義匯出至 XML 檔，隨後即可匯入該檔案以從備份建立新的資源監視器。

相關參考

[第 348 頁的『fteCreateMonitor: 建立 MFT 資源監視器』](#)

fteCreateMonitor 指令可從指令行建立並啟動新的資源監視器。您可以使用 Managed File Transfer 來監視資源（例如目錄的內容），以在符合觸發條件時，啟動指定的作業（例如檔案傳送）。

[第 393 頁的『fteDeleteMonitor: 刪除 MFT 資源監視器』](#)

利用指令行，使用 **fteDeleteMonitor** 指令來停止並刪除現有 Managed File Transfer 資源監視器。針對資源監視代理程式發出此指令。

fteListScheduledTransfers

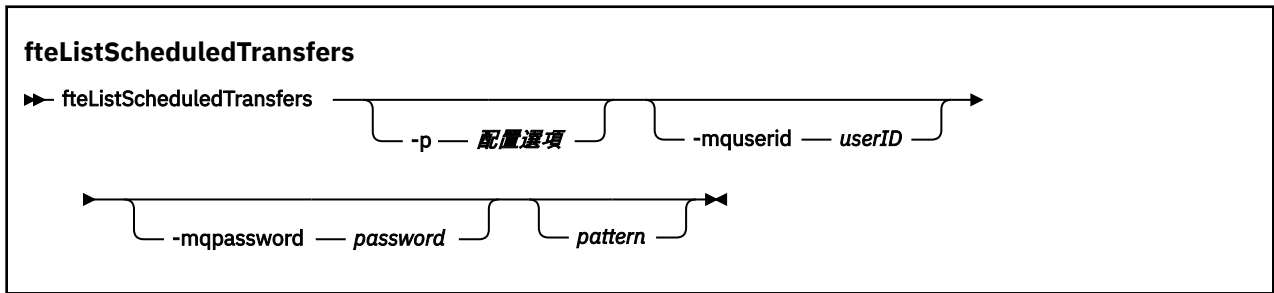
使用 **fteListScheduledTransfers** 指令來列出您先前使用指令行或 IBM MQ Explorer 建立的所有 Managed File Transfer 傳送。

用途

您可以根據來源代理程式名稱或協調佇列管理程式來列出所有排定的傳送。

只有在您想要使用不同於預設值的配置選項時，才為此指令指定選用的 **-p** 參數。如果您沒有指定 **-p**，則會使用 `installation.properties` 中定義的配置選項。如需相關資訊，請參閱 [第 49 頁的『多平台上的 MFT 配置選項』](#)。

語法



參數

`-p` (*configuration_options*)

選用項目。如果您有多個協調佇列管理程式，請使用此參數明確指定您想要為哪些代理程式列出排定的傳送。請使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，指令即會使用與此非預設協調佇列管理程式相關聯的配置選項。

如果您未指定此參數，則會使用以預設協調佇列管理程式為基礎的配置選項。

`-mquserid` (*userID*)

選用項目。指定用來向協調佇列管理程式進行鑑別的使用者 ID。

`-mqpassword` (*password*)

選用項目。指定用來向協調佇列管理程式進行鑑別的密碼。您還必須指定 `-mquserid` 參數。如果指定了 `-mquserid`，卻未指定 `-mqpassword`，則系統會提示您提供相關聯的密碼。將不會顯示密碼。

型樣

選用項目。用來過濾 Managed File Transfer 排定傳送清單的型樣。此型樣會比對來源代理程式名稱。星號 (*) 字元會被解譯為符合零個以上字元的萬用字元。

如果未指定此參數，依預設會列出所有向協調佇列管理程式登錄的排定傳送。

`-?` 或 `-h`

選用項目。顯示指令語法。

範例

在此範例中，會列出來源代理程式符合型樣 *2 的所有排定傳送：

```
fteListScheduledTransfers "*2"
```

此範例指令會產生下列輸出。排程開始時間與下次傳送時間會以世界標準時間 (UTC) 顯示：

```
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
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

相關工作

第 135 頁的『建立排定的檔案傳送』

您可以從「IBM MQ Explorer 探險家」或從指令行排定新的檔案傳送。排定的傳送可以包含群組中的單一檔案或多個檔案。您可以執行排定的檔案傳送一次，或多次重複傳送。

相關參考

第 394 頁的『fteDeleteScheduledTransfer』

fteListTemplates

使用 **fteListTemplates** 指令，可列出協調佇列管理程式上的可用 Managed File Transfer 傳送範本。

用途

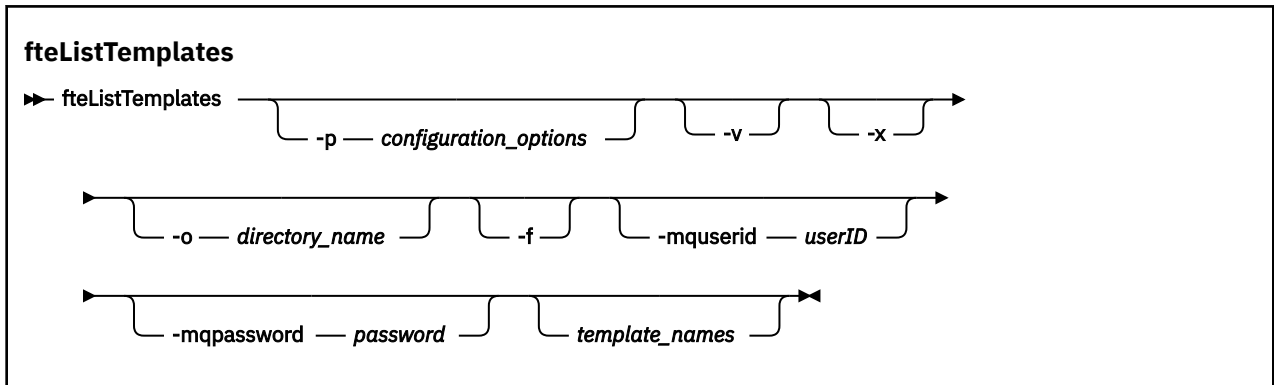
此指令可列出所有範本名稱或過濾選取的範本名稱。清單的輸出格式可以是下列任一種：

- 僅範本名稱（預設行為）
- 含有範本摘要的範本名稱（詳細模式）
- 說明範本的完整 XML 訊息（**-x** 及 **-o** 參數）

此指令會使用 `coordination.properties` 檔連接至協調佇列管理程式。如需相關資訊，請參閱 [第 456 頁的『MFT coordination.properties 檔』](#)。

僅在要使用一組不同於您的預設集的配置選項時，才為此指令指定選用的 **-p** 參數。如需相關資訊，請參閱 [配置選項](#)。

語法



參數

-p

選用項目。此參數決定一組用來刪除範本的配置選項。依照慣例，會使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，指令即會使用與此非預設協調佇列管理程式相關聯的一組內容檔。

如果未指定此參數，則會使用一組以預設協調佇列管理程式為基礎的配置選項。

-v

選用項目。指定詳細模式，並提供每一個相符範本的簡短摘要。如果同時指定了 **-x** 參數，則會忽略此參數。

-v 參數會併入每一個範本的摘要。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** 參數，依預設輸出模式會列出相符範本的名稱。

-x

選用項目。提供每一個相符範本的 XML 格式訊息。如果未同時指定 **-o** 參數，則會忽略此參數。

-o (directory_name)

選用項目。將 XML 格式訊息傳送至指定目錄中的檔案。每一個範本各會建立一個檔案，而每一個檔案都會使用與範本相同的名稱，並加上 .xml 字尾。如果未同時指定 **-x** 參數，則會忽略此參數。

-f

選用項目。強制改寫任何現有的輸出檔。如果未同時指定 **-o** 參數，則會忽略此參數。如果未指定 **-f**，但指定了現有輸出檔的名稱，預設行為是報告錯誤並繼續作業。

-mquserid (userID)

選用項目。指定用來向協調佇列管理程式進行鑑別的使用者 ID。

-mqpassword (password)

選用項目。指定用來向協調佇列管理程式進行鑑別的密碼。您還必須指定 **-mquserid** 參數。如果指定了 **-mquserid**，卻未指定 **-mqpassword**，則系統會提示您提供相關聯的密碼。將不會顯示密碼。

(template_names)

選用項目。要列出的一個以上範本名稱清單。範本名稱可以併入星號作為萬用字元，表示符合零個以上字元。視作業系統而定，您可能需要為任何包括萬用字元的範本名稱加上引號 (" ") 或單引號 (')，以避免 Shell 展開。Shell 展開可能造成非預期的行為。

如果未指定任何 *template_names* 值，依預設將會列出所有範本。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，會列出名稱以 ST 開頭的所有範本：

```
fteListTemplates "ST*"
```

此範例會在現行目錄的 STANDBY.xml 檔中，建立 XML 格式訊息形式的範本 STANDBY：

```
fteListTemplates -x -o . STANDBY
```

此指令會在 STANDBY.xml 中建立下列輸出：

```
<?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>

```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

fteMigrate 代理程式: 將 FTE 7.0 代理程式移轉至 MQ 7.5 或更新版本

如果要將現有代理程式及其相關聯配置從 IBM WebSphere MQ File Transfer Edition 7.0 移轉至 IBM WebSphere MQ 7.5 或更新版本，請使用 **fteMigrateAgent** 指令進行移轉。此指令可用於移轉標準代理程式、Connect:Direct 代理程式或通訊協定橋接器代理程式。您也可以使用此指令在單一要求中移轉多個代理程式。

V 9.0.0 從 IBM MQ 9.0 開始，Managed File Transfer 不支援 Web 代理程式。如果您嘗試使用 **fteMigrateAgent** 指令，將 Web 代理程式從舊版次移轉至 IBM MQ 9.0，則會顯示錯誤訊息，說明不支援移轉 Web 代理程式。

註: 如果您要從 IBM WebSphere MQ File Transfer Edition 7.0 版或更新版本進行移轉，且要繼續使用 FTE_CONFIG 環境變數，則不需要變更 FTE_CONFIG 值即可完成。您可以執行標準移轉，但不得設定 BFG_DATA，且 FTE_CONFIG 的設定必須與 IBM WebSphere MQ 7.0 中使用的相同。

重要: **ULW** 在 IBM MQ for UNIX, Linux, and Windows 上，只有 IBM MQ 管理者（及 mqm 群組的成員）身分的使用者才能執行此指令。如果您嘗試以非 IBM MQ 管理者的使用者身分執行此指令，則會收到錯誤訊息 BFGCL0502E：您未獲授權執行所要求的作業。，且指令將不會執行。

z/OS 在 z/OS 系統上，使用者必須滿足（至少）下列其中一項條件才能執行移轉指令：

- 是 mqm 群組的成員（如果 mqm 群組已定義在系統上）。
- **V 9.0.0.1** 是 BFG_GROUP_NAME 環境變數所指名之群組的成員（如果已指名群組的話）。
- **V 9.0.0.1** 指令執行時，BFG_GROUP_NAME 環境變數中未設定任何值。

如果代理程式配置為以 Windows 服務方式執行，請使用 **fteModifyAgent** 指令來重新配置代理程式，使其不再是 Windows 服務。移轉完成之後，請再次使用 **fteModifyAgent** 指令，將新的代理程式配置為 Windows 服務。此外，如有加入 **-f** 參數，指令會完成，但會產生警告。

您必須先使用 **fteStopAgent** 指令停止要移轉的代理程式，才能執行 **fteMigrateAgent** 指令。

若您搭配 **-f** 參數執行此指令，則只會重新整理代理程式的相關資訊。如果遺漏必要檔案，指令會失敗。

具體而言，此指令會移轉下列與代理程式相關聯的內容檔、XML 檔案及目錄：

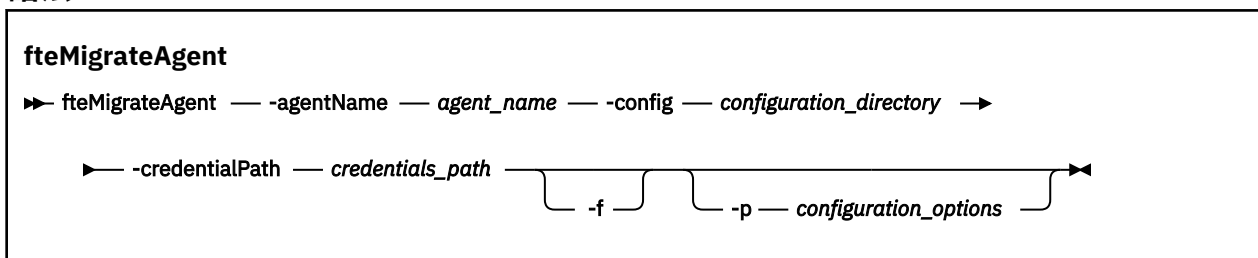
表 37: fteMigrateAgent 指令所移轉的代理程式檔案	
fteMigrateAgent 指令針對每一個代理程式移轉的檔案名稱	資訊
wmqfte.properties	在 IBM WebSphere MQ 7.5 或更新版本中，wmqfte.properties 檔案會重新命名為 installation.properties。
command.properties	
coordination.properties	

表 37: `fteMigrateAgent` 指令所移轉的代理程式檔案 (繼續)

fteMigrateAgent 指令針對每一個代理程式移轉的檔案名稱	資訊
<code>coordination_queue_manager.mqsc</code>	
<code>agent_name_create.mqsc</code>	
<code>agent_name_delete.mqsc</code>	
exits 目錄	指令會複製 exits 目錄中的所有檔案。
僅適用於標準代理程式:	
<code>UserSandboxes.xml</code>	
僅適用於 Connect:Direct 橋接器代理程式:	
<code>ConnectDirectCredentials.xml</code>	
<code>ConnectDirectNodeProperties.xml</code>	
<code>ConnectDirectProcessDefinitions.xml</code>	
僅適用於通訊協定橋接器代理程式:	
<code>ProtocolBridgeCredentials.xml</code>	
<code>ProtocolBridgeProperties.xml</code>	此檔案僅存在於 IBM WebSphere MQ File Transfer Edition 7.0.4 版或更新版本。

fteMigrateAgent 指令會移轉用於安裝、協調和指令佇列管理程式的檔案，並將這些檔案複製到 IBM WebSphere MQ 7.5 版或更新版本，前提是 IBM WebSphere MQ 7.5 版或更新版本中還沒有這些檔案。如果這些檔案已經存在，在執行此指令時就不會進行複製。

語法



參數

-agentName `agent_name`

必要項目。要移轉至 IBM WebSphere MQ 7.5 版或更新版本的代理程式的名稱。

-config `configuration_directory`

必要項目。您要從中移轉代理程式之安裝的配置目錄路徑。例如: `C:\Documents and Settings\All Users\Application Data\IBM\WMQFTE\config`

-credentialPath `credentials_path`

必要項目。定義將認證資訊移轉至其中的位置。此參數可以是現有認證檔案所在的目錄路徑，或是用來接收新認證檔案的新位置。對於 z/OS 平台，這可以是預先存在的分割資料集延伸 (PDSE)，其中含有要更新的現有成員，或是沒有現有成員以併入這些認證的新成員。

註: 如果使用 PDSE，它必須是分區塊變數。

-f

選用項目。即使有部分以一般方式移轉的配置檔與現有配置發生衝突，仍強制代理程式進行移轉。例如，如果 Managed File Transfer 上的內容檔與 IBM WebSphere MQ 7.5 上的內容檔或更新版本之間存在不符，則指定 **-f** 參數表示忽略此不符。

-p configuration_options

選用項目。此參數決定一組用來找出要移轉之配置的配置選項。使用一組配置選項的名稱，作為 **-p** 參數的值。依照慣例，這是協調佇列管理程式的名稱。如果未指定此參數，則會使用一組預設的配置選項。如需相關資訊，請參閱第 49 頁的『多平台上的 MFT 配置選項』。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，AGENT3 及其在 /var/ibm/WMQFTE/config 中的配置將移轉至 IBM WebSphere MQ 7.5 版或更新版本：

```
fteMigrateAgent -agentName AGENT3 -config /var/ibm/WMQFTE/config -credentialPath /home/user1/AGENT3
```

在此範例中，C:\Documents and Settings\All Users\Application Data\IBM\WMQFTE\config 中的所有代理程式及其配置都會移轉至 IBM WebSphere MQ 7.5 或更新版本。Windows 檔案路徑以雙引號 (") 括住。指定 **-f** 參數以強制移轉，並忽略任何不符的內容檔：

```
fteMigrateAgent -agentName "*" -config "C:\Documents and Settings\All Users\Application Data\IBM\WMQFTE\config" -credentialPath "C:\Documents and Settings\user1\AGENT3" -p "configurationOption" -f
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

如需回覆碼的相關資訊，請參閱第 284 頁的『MFT 的回覆碼』。



fteMigrateConfigurationOptions: 將 FTE 7.0 配置移轉至 MQ 7.5 或更新版本

fteMigrateConfigurationOptions 指令會從 IBM WebSphere MQ File Transfer Edition 7.0 移轉一組配置選項，並將其複製到 IBM WebSphere MQ 7.5 或更新版本，但前提是目標版本上還沒有這些檔案。如果這些檔案已存在，則會輸出一則訊息，且不會繼續進行指令。

註：如果您要從 IBM WebSphere MQ File Transfer Edition 7.0 進行移轉，且要繼續使用 FTE_CONFIG 環境變數，則不需要變更 FTE_CONFIG 值即可完成。您可以執行標準移轉，但不得設定 BFG_DATA，且 FTE_CONFIG 的設定必須與 IBM WebSphere MQ File Transfer Edition 7.0 中使用的相同。

重要：  在 IBM MQ for UNIX, Linux, and Windows 上，只有 IBM MQ 管理者（及 mqm 群組的成員）身分的使用者才能執行此指令。如果您嘗試以非 IBM MQ 管理者的使用者身分執行此指令，則會收到錯誤訊息 BFGCL0502E：您未獲授權執行所要求的作業。，且指令將不會執行。

 在 z/OS 系統上，使用者必須滿足（至少）下列其中一項條件才能執行移轉指令：

- 是 mqm 群組的成員（如果 mqm 群組已定義在系統上）。
-  是 BFG_GROUP_NAME 環境變數所指名之群組的成員（如果已指名群組的話）。
-  指令執行時，BFG_GROUP_NAME 環境變數中未設定任何值。

語法

fteMigrateConfigurationOptions

```
► fteMigrateConfigurationOptions — -config — configuration_directory — -credentialPath →  
  
← credentials_path — -configurationOptionsName — configuration_options_name ◄
```

參數

-config (*configuration_directory*)

必要項目。您要從中移轉之安裝的配置目錄路徑。例如: C:\Documents and Settings\All Users\Application Data\IBM\WMQFTE\config

-credentialPath (*credentials_path*)

必要項目。定義將認證資訊移轉至其中的位置。此參數可以是現有認證檔案所在的目錄路徑，或是用來接收新認證檔案的新位置。

此參數用於將 `agent.properties`、`coordination.properties` 及 `command.properties` 檔案中呈現的 SSL/TLS 金鑰儲存庫及信任儲存庫內容的密碼內容從 IBM WebSphere MQ 7.5 之前的產品版本移轉至 7.5 或更新版本。

在 IBM WebSphere MQ 7.5 之前，密碼內容 (例如 `coordinationSslTrustStorePassword`) 曾出現在 `coordination.properties` 檔案中，但已移至 IBM WebSphere MQ 7.5 中的 `MQMFTCredentials.xml` 檔案。若要將密碼內容從 `coordination.properties` 檔案移至 `MQMFTCredentials.xml` 檔案，則會使用 **-credentialsPath**。使用此選項時，**fteMigrateConfigurationOptions** 指令會查看 `coordination.properties` 檔案中是否存在 "coordinationSslTrustStorePassword"，如果存在，則將該內容移轉至 `MQMFTCredentials.xml` 檔案。

z/OS 對於 z/OS 平台，這可以是預先存在的分割資料集延伸 (PDSE)，其中含有要更新的現有成員，或是沒有現有成員以併入這些認證的新成員。

註: 如果使用 PDSE，該 PDSE 必須是可變區塊。

-configurationOptionsName (*configuration_options_name*)

必要項目。您要移轉之一組配置選項的名稱。您可以使用星號字元 (*) 代表零個以上字元，來移轉多組配置選項。您可以使用星號搭配字串。例如，若要移轉名稱以 IBM 開頭的所有配置選項集，請使用此參數，如下所示: `-configurationOptionsName IBM*`。

範例

在此範例中，會移轉 C:\Documents and Settings\All Users\Application Data\IBM\WMQFTE\config 目錄中的所有配置。目錄路徑以雙引號括住:

```
fteMigrateConfigurationOptions -config "C:\Documents and Settings\All Users\Application  
Data\IBM\WMQFTE\config"  
-credentialPath "C:\Documents and Settings\user1\configurationoptions" -configurationOptionsName *
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

相關參考

第 12 頁的『MFT 的移轉考量』

如果您計劃移轉 Managed File Transfer，請檢閱下列資訊。

第 407 頁的『fteMigrate 代理程式: 將 FTE 7.0 代理程式移轉至 MQ 7.5 或更新版本』

如果要將現有代理程式及其相關聯配置從 IBM WebSphere MQ File Transfer Edition 7.0 移轉至 IBM WebSphere MQ 7.5 或更新版本, 請使用 **fteMigrateAgent** 指令進行移轉。此指令可用於移轉標準代理程式、Connect:Direct 代理程式或通訊協定橋接器代理程式。您也可以使用此指令在單一要求中移轉多個代理程式。

第 411 頁的『fteMigrate 日誌程式: 將 FTE 7.0 資料庫日誌程式移轉至 MQ 7.5 或更新版本』

如果要將現有獨立式資料庫日誌程式的配置從 IBM WebSphere MQ File Transfer Edition 7.0.1 或更新版本移轉至 IBM WebSphere MQ 7.5 或更新版本, 請使用 **fteMigrateLogger** 指令。

fteMigrate 日誌程式: 將 FTE 7.0 資料庫日誌程式移轉至 MQ 7.5 或更新版本



如果要將現有獨立式資料庫日誌程式的配置從 IBM WebSphere MQ File Transfer Edition 7.0.1 或更新版本移轉至 IBM WebSphere MQ 7.5 或更新版本, 請使用 **fteMigrateLogger** 指令。

您不能使用此指令來移轉 Java EE 資料庫日誌程式: 請改用 [移轉 Java EE 資料庫日誌程式](#) 中的資訊。

註: 如果您要從 IBM WebSphere MQ 7.0 版或更新版本進行移轉, 且要繼續使用 FTE_CONFIG 環境變數, 則不需要變更 FTE_CONFIG 值即可完成。您可以執行標準移轉, 但不得設定 BFG_DATA, 且 FTE_CONFIG 的設定必須與 IBM WebSphere MQ 7.0 中使用的相同。

重要:  在 IBM MQ for UNIX, Linux, and Windows 上, 只有 IBM MQ 管理者 (及 mqm 群組的成員) 身分的使用者才能執行此指令。如果您嘗試以非 IBM MQ 管理者的使用者身分執行此指令, 則會收到錯誤訊息 BFGCL0502E: 您未獲授權執行所要求的作業。 , 且指令將不會執行。

 在 z/OS 系統上, 使用者必須滿足 (至少) 下列其中一項條件才能執行移轉指令:

- 是 mqm 群組的成員 (如果 mqm 群組已定義在系統上)。
-  是 BFG_GROUP_NAME 環境變數所指名之群組的成員 (如果已指名群組的話)。
-  指令執行時, BFG_GROUP_NAME 環境變數中未設定任何值。

如果您已將獨立式資料庫日誌程式配置成以 Windows 服務方式執行, 即無法使用 **fteMigrateLogger** 指令來移轉該日誌程式的配置。如果您對配置成以 Windows 服務方式執行的日誌程式執行 **fteMigrateLogger** 指令, 該指令會產生錯誤且不會繼續執行。此外, 如有加入 **-f** 參數, 指令會完成, 但會產生警告。

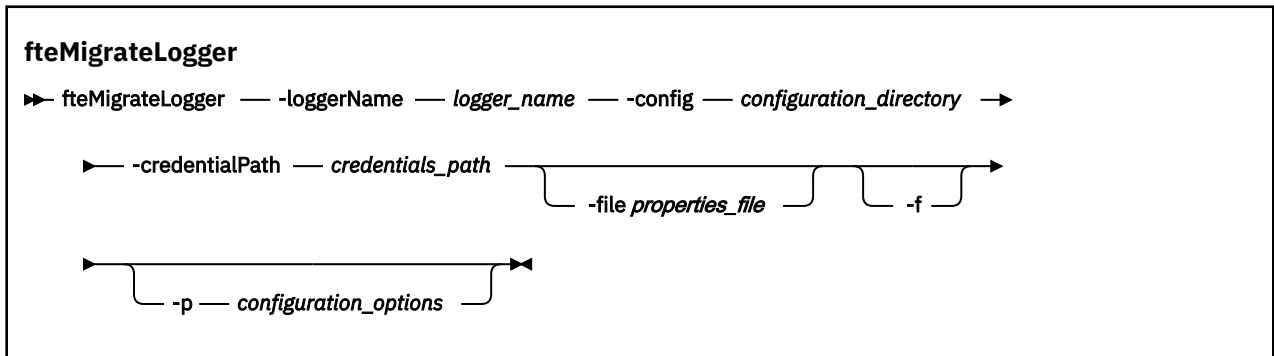
在執行 **fteMigrateLogger** 指令之前, 請停止您要在 IBM WebSphere MQ File Transfer Edition 7.0 上移轉其配置的資料庫日誌程式。

如果您搭配 **-f** 參數執行此指令, 將只會重新整理日誌程式的相關資訊。如果遺漏必要檔案, 指令會失敗。具體而言, 此指令會移轉下列與日誌程式配置相關聯的內容檔及 .mqsc 檔案:

fteMigrateLogger 指令移轉的檔案名稱	資訊
wmqfte.properties	wmqfte.properties 檔案以 IBM WebSphere MQ 7.5 或更新版本中的 installation.properties 為基礎
command.properties	
coordination.properties	
coordination_queue_manager.mqsc	
databaselogger.properties, 或使用 -file 參數指定的其他內容檔	databaselogger.properties 用來在 IBM WebSphere MQ 7.5 或更新版本中建立 logger.properties 檔案。

fteMigrateLogger 指令會移轉用於安裝、協調和指令佇列管理程式的檔案, 並將這些檔案複製到 IBM WebSphere MQ 7.5 或更新版本, 前提是目標版本中還沒有這些檔案。如果這些檔案已經存在, 在執行此指令時就不會進行複製。

語法



參數

-loggerName logger_name

必要項目。您希望為 IBM WebSphere MQ 7.5 或更新版本中的已移轉日誌程式配置指定的名稱。如需 IBM WebSphere MQ 7.5 的新日誌程式名稱的相關資訊，請參閱 [logger_name](#) 參數。

-config configuration_directory

必要項目。您要從中移轉日誌程式配置之安裝的配置目錄路徑。

-credentialPath credentials_path

必要項目。定義將認證資訊移轉至其中的位置。此參數可以是現有認證檔案所在的目錄路徑，或是用來接收新認證檔案的新位置。對於 z/OS 平台，這可以是預先存在的分割資料集延伸 (PDSE)，其中含有要更新的現有成員，或是沒有現有成員以併入這些認證的新成員。

註：如果使用 PDSE，該 PDSE 必須是可變區塊。

-file properties_file

選用項目。指定要移轉的資料庫日誌程式內容檔。只有在內容檔未使用下列預設名稱及路徑時，才需使用此參數：`configuration_directory/coordination_qmgr_name/databaselogger.properties`

-f

選用項目。即使有部分以一般方式移轉的配置檔與現有配置發生衝突，仍強制進行移轉。例如，如果 IBM WebSphere MQ File Transfer Edition 上的資料庫日誌程式內容檔與 IBM WebSphere MQ 7.5 或更新版本上的內容檔不符，則指定 **-f** 參數表示忽略此不符。

-p configuration_options

選用項目。此參數決定一組用來找出要移轉之日誌程式配置的配置選項。使用一組配置選項的名稱，作為 **-p** 參數的值。依照慣例，這是協調佇列管理程式的名稱。如果未指定此參數，則會使用一組預設的配置選項。如需相關資訊，請參閱第 49 頁的『多平台上的 MFT 配置選項』。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，位於 `/var/ibm/WMQFTE/config` 中的獨立式資料庫日誌程式的配置，會移轉至 IBM WebSphere MQ 7.5，並命名為 `FTELOGGER1`：

```
fteMigrateLogger -loggerName FTELOGGER1 -config /var/ibm/WMQFTE/config  
-credentialPath /home/user1/FTELOGGER1
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

如需回覆碼的相關資訊，請參閱第 284 頁的『MFT 的回覆碼』。

執行 fteMigrateLogger 指令之後

若要驗證移轉，請在順利執行 **fteMigrateLogger** 指令之後，在 IBM WebSphere MQ 7.5 版或更新版本上使用第 447 頁的『fteStartLogger』指令，來啟動您已移轉其配置的資料庫日誌程式。

相關參考

第 12 頁的『MFT 的移轉考量』

如果您計劃移轉 Managed File Transfer，請檢閱下列資訊。

第 407 頁的『fteMigrate 代理程式: 將 FTE 7.0 代理程式移轉至 MQ 7.5 或更新版本』

如果要將現有代理程式及其相關聯配置從 IBM WebSphere MQ File Transfer Edition 7.0 移轉至 IBM WebSphere MQ 7.5 或更新版本，請使用 **fteMigrateAgent** 指令進行移轉。此指令可用於移轉標準代理程式、Connect:Direct 代理程式或通訊協定橋接器代理程式。您也可以使用此指令在單一要求中移轉多個代理程式。

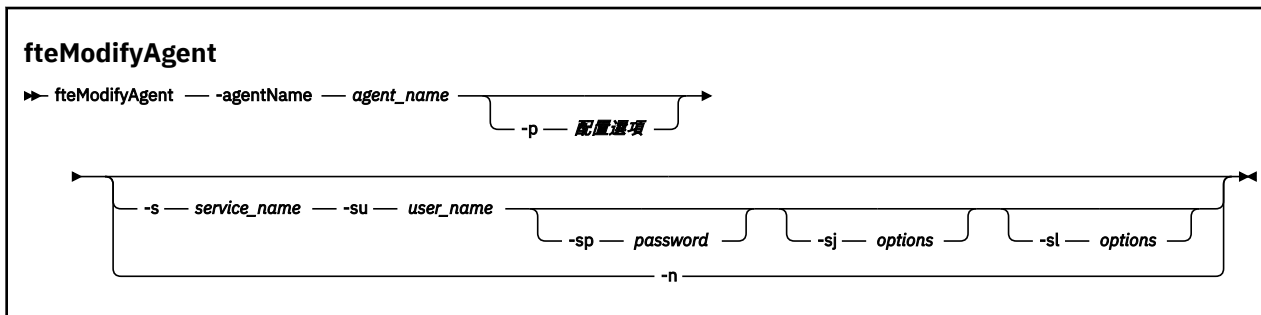
第 409 頁的『fteMigrateConfigurationOptions: 將 FTE 7.0 配置移轉至 MQ 7.5 或更新版本』

fteMigrateConfigurationOptions 指令會從 IBM WebSphere MQ File Transfer Edition 7.0 移轉一組配置選項，並將其複製到 IBM WebSphere MQ 7.5 或更新版本，但前提是目標版本上還沒有這些檔案。如果這些檔案已存在，則會輸出一則訊息，且不會繼續進行指令。

Windows fteModify 代理程式 (以 Windows 服務方式執行 MFT 代理程式)

fteModifyAgent 指令可修改現有代理程式，讓它能夠以 Windows 服務方式執行。此指令僅在 Windows 上可用，且必須由 IBM MQ 管理者及 mqm 群組成員的使用者執行。

語法



參數

-agentName *agent_name*

必要項目。您想要修改的代理程式名稱。

-p *configuration_options*

選用項目。此參數決定一組用來修改代理程式的配置選項。依照慣例，會使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，**fteModifyAgent** 指令即會使用與此非預設協調佇列管理程式相關聯的一組內容檔。

只有在您想要使用不同於預設值的配置選項時，才需指定選用的 **-p** 參數。如果未指定此參數，則會使用一組以預設協調佇列管理程式為基礎的配置選項。

-s *service_name*

選用項目。指出代理程式是以 Windows 服務方式執行。如果未指定 *service_name*，則此服務會命名為 `mqmftAgentAGENTQMGR`，其中 *AGENT* 是代理程式名稱，而 *QMGR* 是代理程式佇列管理程式名稱。

服務的顯示名稱（顯示在 Windows 服務視窗中的名稱直欄內）一律為 **Managed File Transfer Agent AGENT@QMGR**。

註：如果可轉散發的代理程式將以 Windows 服務形式來執行，則需要在系統環境中設定 **BFG_DATA** 環境變數，服務才能運作。

-su user_name

選用項目。代理程式以 Windows 服務方式執行時，此參數會指定服務執行時應使用的帳戶名稱。若使用 Windows 網域使用者帳戶執行此代理程式，請以 `DomainName\UserName` 格式指定此值。若要使用本端內建網域中的帳戶執行服務，請以 `UserName` 格式指定值。

您使用 **-su** 參數指定的 Windows 使用者帳戶必須具有 **Log on as a service** 權限。如需如何授與此權限的相關資訊，請參閱 [以 Windows 服務方式執行 MFT 代理程式或日誌程式的指引](#)。

指定 **-s** 時需要使用此參數。

-sp password

選用項目。此參數僅在指定 **-s** 時有效。

-sj options

選用項目。以 Windows 服務方式啟動代理程式時，此參數會以 **-D** 或 **-X** 的格式定義要傳遞至 Java Virtual Machine (JVM) 的選項清單。這些選項會使用 **#** 記號或分號 (**;**) 字元來區隔。如果必須內含任何 **#** 或 **;** 字元，請以單引號括住這些字元。

此參數僅在指定 **-s** 時有效。

如需 **fteModifyAgent** 指令處理 JVM 選項更新驗證的方式的相關資訊，請參閱 [更新代理程式或日誌程式 JVM 選項的指引](#)。

-sl options

選用項目。設定 Windows 服務日誌層次。有效的選項是：error、info、warn、debug。預設值是 info。如果您在使用 Windows 服務時遇到問題，則此選項非常有用。將它設為 debug，可以在服務日誌檔中提供更詳細資訊。

此參數僅在指定 **-s** 時有效。

-n

選用項目。指示代理程式以正常程序執行。此參數與 **-s** 選項互斥。如果既未指定 **-s** 也未指定 **-n** 選項，則代理程式會配置為一般 Windows 處理程序。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，會修改 AGENT1，讓它以 Windows 服務方式執行：

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

在此範例中，會修改 AGENT1，以移除 Windows 服務：

```
fteModifyAgent -agentName AGENT1
```

在執行 **fteModifyAgent** 指令之前，您必須先使用 [fteStopAgent](#) 指令停止要修改的代理程式。

回覆碼

0

指令已順利完成。

1

指令未順利結束。

相關參考

第 326 頁的『[fteCreateAgent \(建立 MFT 代理程式\)](#)』

fteCreateAgent 指令會建立 Managed File Transfer Agent 及其相關聯的配置。

第 415 頁的『[fteModify 日誌程式 \(以 Windows 服務方式執行 MFT 日誌程式\)](#)』

使用 **fteModifyLogger** 指令來修改 Managed File Transfer 日誌程式，以便它可以 Windows 服務方式執行。您只能在 Windows 平台上使用此指令，必須由 IBM MQ 管理者及 mqm 群組成員的使用者執行，且必須先使用 **fteStopLogger** 指令停止日誌程式。

相關資訊

以 Windows 服務方式啟動 MFT 代理程式

以 Windows 服務方式執行 MFT 代理程式或日誌程式的指引

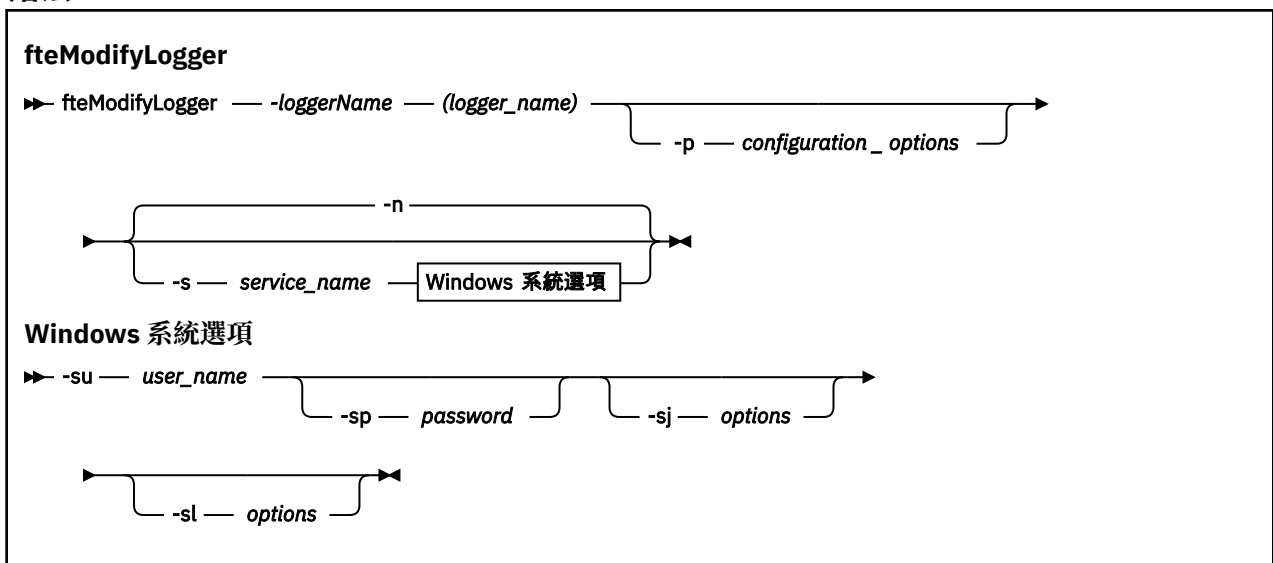
Windows fteModify 日誌程式 (以 Windows 服務方式執行 MFT 日誌程式)

使用 **fteModifyLogger** 指令來修改 Managed File Transfer 日誌程式，以便它可以 Windows 服務方式執行。您只能在 Windows 平台上使用此指令，必須由 IBM MQ 管理者及 mqm 群組成員的使用者執行，且必須先使用 **fteStopLogger** 指令停止日誌程式。

用途

獨立式日誌程式 (不論是針對檔案或資料庫) 在 **服務** 應用程式的 **名稱** 直欄中顯示為 "Managed File Transfer logger for property set *logger_name@logger_qm*"。 *logger_qm* 的值是日誌程式的指令佇列管理程式的名稱。

語法



參數

-loggerName (logger_name)

必要項目。您想要修改的 Managed File Transfer 日誌程式名稱。

-p configuration_options

選用項目。此參數決定一組用來修改日誌程式的配置選項。依照慣例，會使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，**fteModifyLogger** 指令即會使用與此非預設協調佇列管理程式相關聯的一組內容檔。

只有在您想要使用不同於預設值的配置選項時，才需指定選用的 **-p** 參數。如果未指定此參數，則會使用一組以預設協調佇列管理程式為基礎的配置選項。

-s service_name

選用項目。指出日誌程式是以 Windows 服務方式執行。如果未指定 `service_name`，則此服務會命名為 `mqmfLoggerLOGGERQMGR`，其中 `LOGGER` 是日誌程式名稱，而 `QMGR` 是日誌程式佇列管理程式名稱。

服務的顯示名稱（顯示在 Windows 服務視窗中的名稱直欄內）一律為 **Managed File Transfer Logger
LOGGER@QMGR**。

-su user_name

指定 `-s` 時，為必要項目。指定執行 Windows 服務時應使用的帳戶名稱。若要使用 Windows 網域使用者帳戶執行此代理程式，請以 `DomainName\UserName` 格式指定此值。若要使用本端內建網域中的帳戶來執行服務，請指定下列格式的值：`UserName`。

您使用 `-su` 參數指定的 Windows 使用者帳戶必須具有以服務方式登入的許可權。如需如何授與此許可權的相關資訊，請參閱 [以 Windows 服務方式執行 MFT 代理程式或日誌程式的指引](#)。

-sp password

選用項目。此參數僅在指定 `-s` 時有效。 `-su` 參數所設定使用者帳戶的密碼。

當您指定 `-s` 參數時，如果未指定此參數，將會警告您，您必須先使用「Windows 服務」工具設定密碼，然後服務才能順利啟動。

-sj options

選用項目。此參數僅在指定 `-s` 時有效。以 Windows 服務方式啟動日誌程式時，此參數會以 `-D` 或 `-X` 的格式定義要傳遞至 JVM 的選項清單。這些選項會使用 `#` 記號或分號 (`;`) 字元來區隔。如果必須內含任何 `#` 或 `;` 字元，請以單引號 (`'`) 括住這些字元。

如需 `fteModifyLogger` 指令處理 JVM 選項更新驗證的方式的相關資訊，請參閱 [更新代理程式或日誌程式 JVM 選項的指引](#)。

-sl options

選用項目。此參數僅在指定 `-s` 時有效。設定 Windows 服務日誌層次。有效的選項是：`error`、`info`、`warn`、`debug`。預設值是 `info`。如果您在使用 Windows 服務時遇到問題，則此選項非常有用。將它設為 `debug`，可以在服務日誌檔中提供更詳細資訊。

-n

選用項目。指出日誌程式是要以一般程序方式執行。此參數與 `-s` 選項互斥。如果 `-s` 與 `-n` 選項均未指定，則日誌程式會配置為一般 Windows 程序。

-? 或 -h

選用項目。顯示指令語法。

範例

您必須先使用 `fteStopLogger` 指令來停止日誌程式，才能執行 `fteModifyLogger` 指令。

在此範例中，先前已建立一個名為 `logger1` 的日誌程式。此指令顯示如何變更日誌程式，讓它以 Windows 服務方式執行：

```
fteModifyLogger -loggerName logger1 -s -su fteuser -sp ftepassword
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

相關參考

[第 447 頁的『fteStartLogger』](#)

`fteStartLogger` 指令會啟動 Managed File Transfer 記載應用程式。

第 450 頁的『fteStopLogger』

fteStopLogger 指令會停止 Managed File Transfer 日誌程式。

相關資訊

以 Windows 服務方式啟動 MFT 代理程式

以 Windows 服務方式執行 MFT 代理程式或日誌程式的指引

fteObfuscate: 加密機密資料

fteObfuscate 指令可為認證檔案中的機密資料加密。這會讓取得檔案存取權的人員無法讀取認證檔案的內容。

用途

可以模糊化認證檔案中的使用者名稱和密碼內容。這些內容會轉換成新的相關內容，並含有「密碼」字尾。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>
```

語法

fteObfuscate

►► fteObfuscate — **-credentialsFile** — *credentials_file_name* ►►

參數

-credentialsFile

必要項目。要將內容模糊化的認證檔案的名稱。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，會將 MQMFTCredentials.xml 內容模糊化。

```
fte0bfuscate -credentialsFile /home/fteuser/MQMFTCredentials.xml
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

相關參考

第 36 頁的『MFT 及 IBM MQ 連線鑑別』

連線鑑別可讓佇列管理程式配置成使用提供的使用者 ID 和密碼來鑑別應用程式。如果相關聯的佇列管理程式已啟用安全，且需要認證詳細資料(使用者 ID 和密碼)，則必須先啟用連線鑑別功能，才能順利建立與佇列管理程式的連線。連線鑑別可以在相容模式或 MQCSP 鑑別模式中執行。

第 696 頁的『MFT 認證檔格式』

MQMFTCredentials.xml 檔案包含機密的使用者 ID 及密碼資訊。MQMFTCredentials.xml 檔案中的元素必須符合 MQMFTCredentials.xsd 綱目。認證檔的安全由使用者負責維護。

ftePingAgent: 檢查 MFT 代理程式是否處於作用中狀態

ftePingAgent 指令會對 Managed File Transfer 代理程式進行連線測試，以判斷是否可以呼叫到代理程式，如果可以呼叫到，代理程式是否能夠回應簡式查詢。

用途

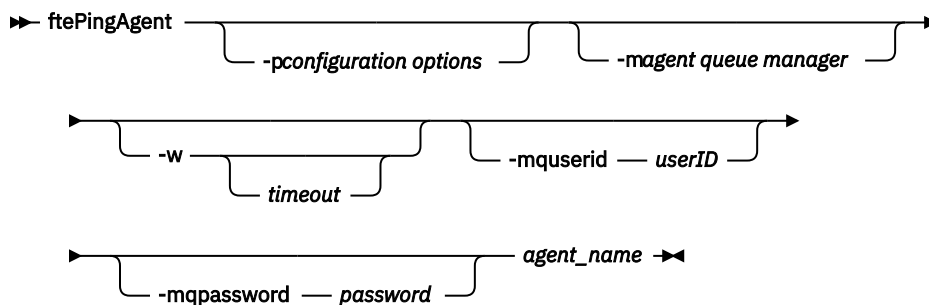
使用 **ftePingAgent** 指令來檢查是否可以呼叫到 Managed File Transfer 代理程式，如果可以，則檢查它是否能夠沿著 are you there? 這幾行來回應簡式查詢。此指令的範例輸出如下所示：

```
C:\> ftePingAgent AGENT86
5724-H72 Copyright IBM Corp. 2008 , 2023. ALL RIGHTS RESERVED
BFGPR0127W: 未指定要連線至 IBM MQ 的認證檔。因此，會假設 IBM MQ 鑑別已停用。
BFGCL0212I: 正在向代理程式 AGENT86 發出連線測試要求
BFGCL0213I: 代理程式 AGENT86 在 0.094 秒內回應了連線測試。
```

僅在要使用一組不同於您的預設集的配置選項時，才為此指令指定選用的 **-p** 參數。如需相關資訊，請參閱配置選項。

語法

ftePingAgent



參數

-p (configuration options)

選用項目。此參數決定一組用來發出要求對代理程式進行連線測試的配置選項。請使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，指令即會使用與此非預設協調佇列管理程式相關聯的一組內容檔。如果未指定此參數，則會使用一組以預設協調佇列管理程式為基礎的配置選項。如需相關資訊，請參閱[配置選項](#)。

-m (queue manager)

選用項目。您想要連線測試的代理程式所連接的佇列管理程式名稱。如果未指定 **-m** 參數，則會從一組使用的配置選項決定所使用的佇列管理程式。

-w (timeout)

選用項目。指定指令等待代理程式回應的最長時間應該為 *timeout* 秒。如果未指定逾時，或指定的逾時值為 **-1**，則指令會無限期等待，直至代理程式回應為止。如果未指定此選項，則等待代理程式回應的預設時間多達 5 秒。

如果已指定 *timeout*，**ftePingAgent** 指令訊息會在兩倍 *timeout* 值之後逾時，而不會進入指定的「無法傳送的郵件」佇列中。如果該指令已設為無限期等待，則指令訊息不會逾時。

-mquserid (user ID)

選用項目。指定用來向指令佇列管理程式進行鑑別的使用者 ID。

-mqpassword (password)

選用項目。指定用來向指令佇列管理程式進行鑑別的密碼。您還必須指定 **-mquserid** 參數。如果指定了 **-mquserid**，卻未指定 **-mqpassword**，則系統會提示您提供相關聯的密碼。將不會顯示密碼。

(agent name)

必要項目。您想要連線測試的 Managed File Transfer 代理程式名稱。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，指令會對連接至 QM_MERCURY 的代理程式 AGENT1 進行連線測試。傳回 AGENT1 回應之前，指令最多等待時間為 40 秒。

```
ftePingAgent -m QM_MERCURY -w 40 AGENT1
```

回覆碼

0

指令已順利完成。代理程式處於作用中狀態，而且能夠處理傳送。

1

指令未順利結束。指令無法傳送訊息至代理程式。

2

指令結束時逾時。指令已傳送訊息至代理程式，但代理程式未在時間範圍內回應。

相關參考

第 398 頁的『[fteListAgents: 列出協調佇列管理程式的 MFT 代理程式](#)』

使用 **fteListAgents** 指令，以列出向特定協調佇列管理程式登錄的所有 Managed File Transfer 代理程式。

第 436 頁的『[fteShowAgentDetails](#)』

使用 **fteShowAgentDetails** 指令顯示特定 Managed File Transfer Agent 的詳細資料。這些是代理程式的 Managed File Transfer 協調佇列管理程式儲存的詳細資料。

第 264 頁的『[認為檔案傳送停滯時應該如何處理](#)』

在負載沈重的系統上，或來源與目的地代理程式之間發生網路問題時，傳送有時會呈現停滯在佇列或回復狀態。有許多因素可能會造成此狀況。

fteRAS: 收集 MFT 疑難排解資訊

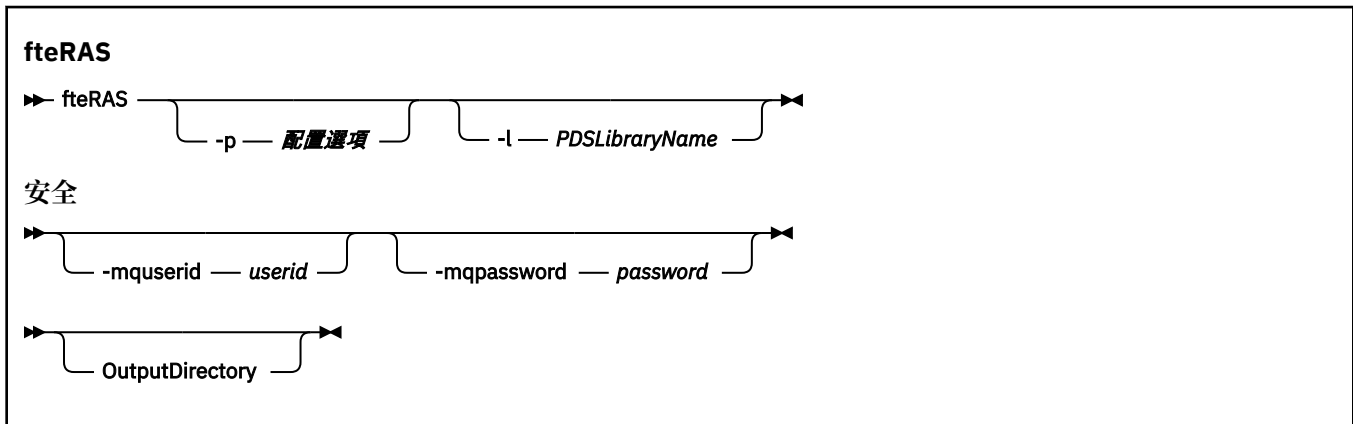
fteRAS 指令會收集 Managed File Transfer 的疑難排解資訊 (MustGather 資料)。**fteRAS** 收集的資訊是特定於正在執行程式之系統上的 Managed File Transfer 安裝。

用途

如果您需要收集疑難排解資訊，以在 Managed File Transfer 代理程式、資料庫日誌程式或其他指令報告問題或無法正常運作時協助尋找解決方案，請使用 **fteRAS** 指令來執行「可靠性、可用性及服務功能資訊 (RAS)」收集工具。

執行 **fteRAS** 指令時，放置所產生保存 (.zip) 檔的輸出目錄可以是預設位置或您選擇的目錄。

語法



參數

-p configuration_options

選用項目。決定一組用來收集疑難排解資訊的配置選項，例如代理程式清單。將配置選項集的名稱用作 **-p** 參數的值。依照慣例，此名稱是協調佇列管理程式的名稱。如果未指定此參數，則會使用一組預設的配置選項。

z/OS -l

選用項目。僅限 z/OS。指定 PDS 程式庫的名稱，該程式庫包含針對特定代理程式或日誌程式呼叫 MQMFT 指令的 JCL Script。從指令 PDS 程式庫的 BFGRAS JCL Script 執行該指令時一律會設定此選項，以便在輸出目錄中擷取 PDS 程式庫的所有成員。

-mquserid user id

選用項目。指定用來向指令佇列管理程式進行鑑別的使用者 ID。

-mqpassword password

選用項目。指定用來向指令佇列管理程式進行鑑別的密碼。您還必須指定 **-mquserid** 參數。如果指定了 **-mquserid**，卻未同時指定 **-mqpassword**，則系統會提示您提供相關聯的密碼。密碼不會顯示在畫面上。

OutputDirectory

選用項目。當您收集 RAS 資料時要使用的目錄，例如，在順利收集資料之後，會儲存輸出檔 (例如，fteRAS.zip)。如果該目錄不存在，則會建立它。預設位置為 mqft 日誌目錄。

-? 或 -h

選用項目。顯示指令語法。

範例

Linux **UNIX** 在 UNIX 和 Linux 上，若要將輸出檔 `fteRAS.zip` 儲存在 `/var/mqm/errors` 目錄中，請執行 **fteRAS**，如下列範例所示：

```
fteRAS /var/mqm/errors
```

下列訊息會確認該指令已順利完成：

```
BFGCL0604I: fteRAS command completed successfully. Output is stored in /var/mqm/errors/fteRAS.zip
```

Windows 在 Windows 上，若要將輸出檔 `fteRAS.zip` 儲存在 IBM MQ 新安裝的預設 `errors` 目錄中，請按下列範例中所示執行 **fteRAS**：

```
fteRAS "C:\ProgramData\IBM\MQ\errors"
```

下列訊息會確認該指令已順利完成：

```
BFGCL0604I: fteRAS command completed successfully. Output is stored in  
C:\ProgramData\IBM\MQ\errors\fteRAS.zip
```

註：對於 IBM MQ 8.0 或更新版本，如果這不是該產品版本的新安裝，則 `errors` 目錄的位置在您系統上可能會不同。如需相關資訊，請參閱 [Windows 上的程式和資料目錄位置](#)。

IBM i 在 IBM i 上，若要將輸出檔複製到 `/QIBM/UserData/mqm/errors`，請按下列範例中所示從 Qshell 執行 **fteRAS** 指令：

```
/QIBM/ProdData/mqm/bin/fteRAS /QIBM/UserData/mqm/errors
```

下列訊息會確認該指令已順利完成：

```
BFGCL0604I: fteRAS command completed successfully. Output is stored in /QIBM/UserData/mqm/errors/  
fteRAS.zip
```

相關參考

第 227 頁的『疑難排解 MFT』

使用下列參照資訊來協助您診斷 Managed File Transfer 中的錯誤：

V 9.0.0 fteSetAgentLogLevel: 設定 MFT 通訊協定橋接器代理程式記載層次

使用 **fteSetAgentLogLevel** 指令，針對 Managed File Transfer 通訊協定橋接器代理程式與 FTP/SFTP/FTPS 檔案伺服器之間的檔案傳送，啟用或停用診斷資訊記載。

用途

使用這個指令來啟用或停用記載 Managed File Transfer 通訊協定橋接器代理程式與 FTP/SFTP/FTPS 檔案伺服器間之檔案傳送的診斷資訊集合。「Managed File Transfer」通訊協定橋接器代理程式記載比追蹤更為小型且與之分開，但兩者可以平行執行。

通訊協定橋接器代理程式傳送至 FTP/SFTP/FTPS 檔案伺服器的指令以及收到的回應都會寫入日誌檔，來簡單診斷檔案傳送期間可能發生的問題。

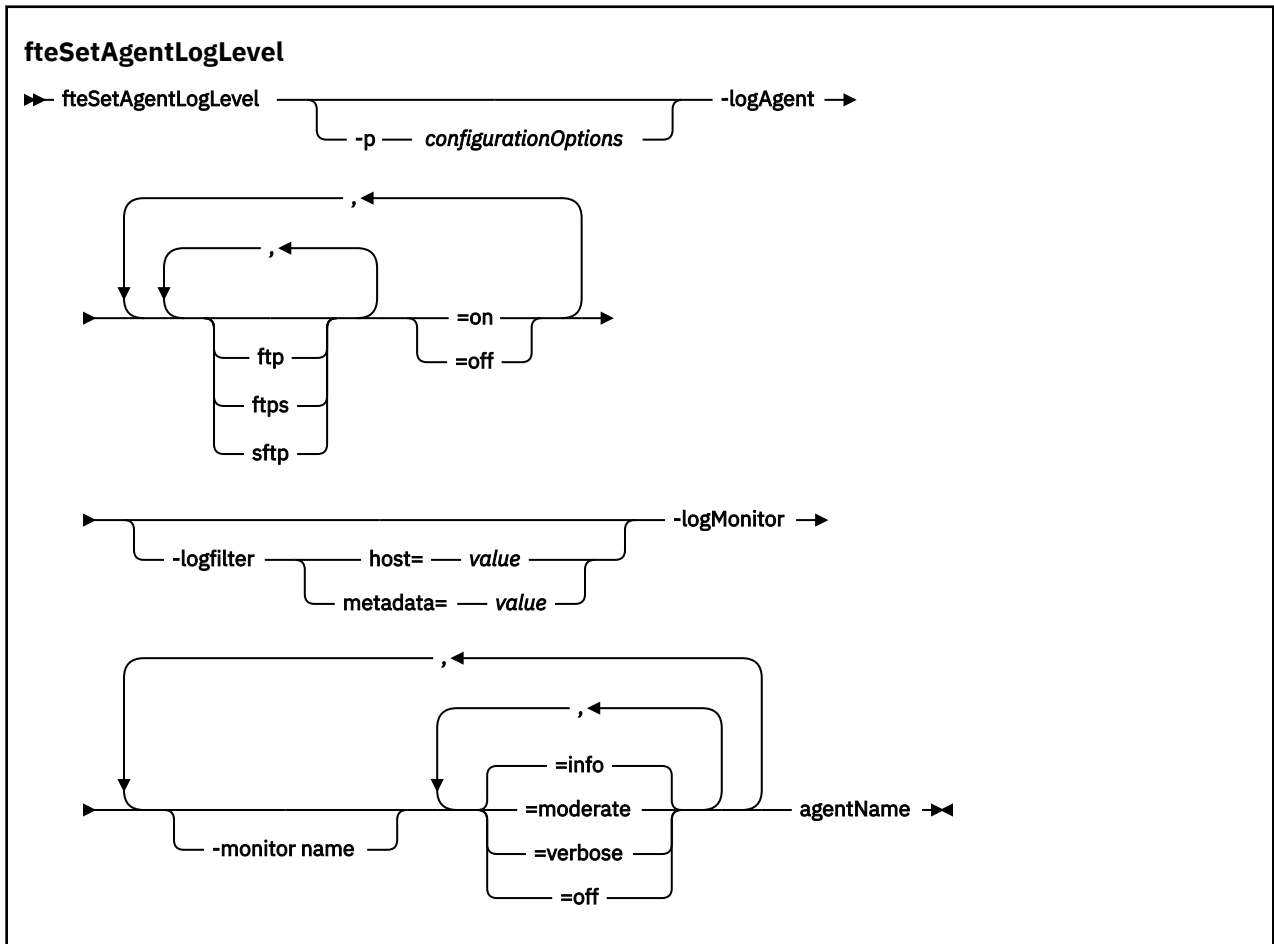
您可以使用兩種方式來啟用、停用及設定所需的記載層次：

- 使用 **fteSetAgentLogLevel** 指令，在通訊協定橋接器代理程式執行時啟用或停用記載。您不需要重新啟動通訊協定橋接器代理程式，即可開始記載。
- 使用 `agent.properties` 檔案，在代理程式啟動時啟用或停用記載。如果您使用此方法來啟用或停用記載，且代理程式正在執行，則必須重新啟動代理程式，才能開始記載。如需相關資訊，請參閱 `agent.properties` 檔案。

通訊協定橋接器代理程式與檔案伺服器之間的互動視為事件，而且每一個事件都會寫入 `agentevent?.log` 檔案。'?' 日誌名稱中的代表數值。名稱中包含 0（零）的檔案包含一組最新的已擷取訊息。日誌檔儲存在 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/` 資料夾中。

您可以在 `agent.properties` 檔案中設定進一步記載內容（例如，限制日誌檔大小以及要保留的日誌檔數目）。

語法



參數

`-logAgent component=operation`

必要項目。

V 9.0.3

重要: 必須僅選取 **logAgent** 和 **logMonitor** 的其中之一。如果嘗試選取兩個選項，則會收到錯誤訊息。

已啟用或停用記載的通訊協定橋接器代理程式名稱。可以啟用或停用三個元件的通訊協定橋接器代理程式記載：`ftp`、`ftps` 及 `sftp` 伺服器。指定三種可能伺服器類型中的其中一種，以及新增作業值來 `off`（關閉）或 `on`（開啟）它們所連接之通訊協定橋接器代理程式的記載。

V 9.0.3

如果元件的開頭是加號 (+)，則加號後面的元件清單會新增至目前正在記載的任何現有記載元件。

有效記載層次作業選項如下所示：

off

停用通訊協定橋接器代理程式的所有記載。此選項是預設值。

```
fteSetAgentLogLevel -logAgent =off PBA1
```

若要停用通訊協定橋接器代理程式所連接之指定元件的記載，請使用這些指令：

```
fteSetAgentLogLevel -logAgent ftp=off PBA1
```

```
fteSetAgentLogLevel -logAgent ftps=off PBA1
```

```
fteSetAgentLogLevel -logAgent sftp=off PBA1
```

on

若要啟用通訊協定橋接器代理程式所連接之所有三個可能的檔案伺服器元件的記載，請使用此指令：

```
fteSetAgentLogLevel -logAgent =on PBA1
```

若要啟用通訊協定橋接器代理程式所連接之指定元件的記載，請使用這些指令：

```
fteSetAgentLogLevel -logAgent ftp=on PBA1
```

```
fteSetAgentLogLevel -logAgent ftps=on PBA1
```

```
fteSetAgentLogLevel -logAgent sftp=on PBA1
```

如需進一步配置選項，請參閱第 425 頁的『範例 1』及第 425 頁的『範例 2』。

-agentName

必要項目。已啟用或停用記載的通訊協定橋接器代理程式名稱。

-logFilter filter=value

選用項目。用來將記載限制為指定的過濾準則。您必須指定 **host** 或使用者指定之檔案傳送 **metadata** 的值。

host

指定三個值的其中之一作為過濾依據：主機名稱、主機名稱清單（以逗點區隔）或 IP 位址清單（以逗點區隔）。若要記載 FTP 伺服器 `ftpprod.ibm.com` 的指令及其回應，請使用此指令：

```
fteSetAgentLogLevel -logAgent ftp=on -logFilter host=ftpprod.ibm.com PBA1
```

若要記載來自 IP 位址開頭為 `9.182.*` 的所有 SFTP 伺服器的指令及回應，請使用此指令：

```
fteSetAgentLogLevel -logAgent sftp=on -logFilter host=9.182.* PBA1
```

metadata

以 `key=value` 格式，指定任何文字（如使用者在傳送建立期間所定義）。例如，**metadata="BANK=WORLD BANK"**。若要啟用連接至通訊協定橋接器代理程式 PBA1 之任何 FTP 伺服器的記載，以及只在它們包含 **metadata** 作為索引鍵及包含 "BANK=WORLD BANK" 作為值時，才會過濾檔案傳送來包括它們，請使用此指令：

```
fteSetAgentLogLevel -logAgent ftp=on metadata="BANK=WORLD BANK" PBA1
```

註：若要依 meta 資料過濾，必須在檔案傳送時於 **-md** 參數下指定過濾依據的值。如需相關資訊，請參閱第 367 頁的『fteCreateTransfer：啟動新的檔案傳送』。

V 9.0.3 -logMonitor monitor name=log level

必要項目。

重要：必須僅選取 **logAgent** 和 **logMonitor** 的其中之一。如果嘗試選取兩個選項，則會收到錯誤訊息。

要啟用或停用記載功能的逗點區隔資源監視器清單，其中：

monitor name

選用項目。要啟用或停用記載功能的資源監視器名稱或資源監視器名稱清單。如果您沒有指定名稱，則會為代理程式的所有資源監視器啟用記載功能。

log level

必要項目。下列其中一項：

info

開啟 info 層次記載功能。這是預設值。

若要為代理程式 AGENT1 的監視器 MON1 啟用 info 層次記載功能，請執行下列動作：

```
fteSetAgentLogLevel -logMonitor MON1=info AGENT1
```

moderate

開啟 moderate 層次記載功能。

若要為代理程式 AGENT1 的監視器 MON1 和 MON2 啟用 moderate 層次記載功能，請執行下列動作：

```
fteSetAgentLogLevel -logMonitor MON1, MON2=moderate AGENT1
```

verbose

開啟 verbose 層次記載功能。

為代理程式 AGENT1 的所有監視器啟用 verbose 層次記載功能：

```
fteSetAgentLogLevel -logMonitor =verbose AGENT1
```

off

關閉記載功能。

若要關閉代理程式 AGENT1 的監視器 MON1 和 MON2 的記載功能，請執行下列動作：

```
fteSetAgentLogLevel -logMonitor MON1, MON2=off AGENT1
```

若要針對代理程式 AGENT1 關閉監視器 MON1 的記載功能並為監視器 MON2 啟用 info 層次記載功能，請執行下列動作：

```
fteSetAgentLogLevel -logMonitor MON1=off, MON2=info AGENT1
```

若要關閉代理程式 AGENT1 的所有監視器的記載功能，請執行下列動作：

```
fteSetAgentLogLevel -logMonitor =off AGENT1
```

V 9.0.3 -p configurationOptions

選用項目。決定用來設定代理程式記載層次的配置選項集。將配置選項集的名稱用作 -p 參數的值。

依照慣例，這是協調佇列管理程式的名稱。如果未指定此參數，則會使用一組預設的配置選項。

-? 或 -h

選用項目。顯示指令語法。

V 9.0.3

如果在指令中重複相同的資源監視器名稱，則元件的最後一個實例視為有效。例如：

```
fteSetAgentLogLevel -logMonitor MON1=info, MON2=off, MON1=off AGENT1
```

關閉資源監視器 MON1 的記載功能。

V 9.0.3



小心：如果已在此指令中指定不存在的資源監視器名稱，則主控台不會顯示任何錯誤。

範例 1

在此範例中，使用指令界定群組，以在一個指令中指定多個元件。啟用 FTP 及 SFTP 伺服器的記載，但停用 PBA1 通訊協定橋接器代理程式所連接之 FTPS 伺服器的記載。

```
fteSetAgentLogLevel -logAgent ftp=on,ftps=off,sftp=on PBA1
```

您也可以使用逗點來區隔元件以達到相同的結果，例如：

```
fteSetAgentLogLevel -logAgent ftp,sftp=on,ftps=off PBA1
```

範例 2

在此範例中，於一個指令中重複相同的元件。*component=operation* 配對的最後一個實例視為有效。此範例會停用 PBA1 通訊協定橋接器代理程式所連接之 FTP 伺服器的記載。

```
fteSetAgentLogLevel -logAgent ftp=on,ftp=off PBA1
```

前一個範例的效果與此範例相同：

```
fteSetAgentLogLevel -logAgent ftp=off PBA1
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

相關概念

第 178 頁的『通訊協定橋接器』

通訊協定橋接器可供 Managed File Transfer (MFT) 網路存取儲存在 MFT 網路之外的檔案伺服器上的檔案（在本端網域或遠端位置）。此檔案伺服器可使用 FTP、FTPS 或 SFTP 網路通訊協定。每一個檔案伺服器需要至少一個專用的代理程式。專用代理程式即所謂的通訊協定橋接器代理程式。橋接器代理程式可以與多個檔案伺服器互動。

相關參考

第 463 頁的『MFT agent.properties 檔案』

每一個 Managed File Transfer Agent 都有自己的內容檔 `agent.properties`，其中必須包含代理程式用來連接至其佇列管理程式的資訊。`agent.properties` 檔案也可以包含變更代理程式行為的內容。

第 367 頁的『fteCreateTransfer: 啟動新的檔案傳送』

fteCreateTransfer 指令從指令行建立並啟動新的檔案傳送。此指令可立即啟動檔案傳送、將檔案傳送排定在未來的時間和日期、將排定的傳送重複一次以上，以及根據特定條件觸發檔案傳送。

fteSetAgentTraceLevel

使用 **fteSetAgentTraceLevel** 指令可動態地修改代理程式的現行追蹤層次。

用途

使用此指令來開啟及關閉代理程式追蹤，或者變更已設定的代理程式追蹤的層次。使用 **fteSetAgentTraceLevel** 指令時，您無需關閉並重新啟動代理程式來修改追蹤層次。產生的追蹤檔案位於 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%/trace%PID%.txt` 中，其中 `%PID%` 是代理程式實例的程序 ID。



小心：

Multi 在 Multiplatforms 上使用 IBM WebSphere MQ 7.5 或更新版本時，只有執行代理程式處理程序的使用者可以執行 **fteSetAgentTraceLevel** 指令。

V 9.0.2 **z/OS** 若為 z/OS，則從 Continuous Delivery 中的 IBM MQ 9.0.2 開始，下列使用者 ID 或成員可以執行 **fteSetAgentTraceLevel** 指令：

- 以其身分執行代理程式程序的使用者 ID。
- 代理程式內容 **adminGroup** 所指定之群組的成員。

V 9.0.0.1 這也適用於 IBM MQ 9.0.0 Fix Pack 1 中的 Long Term Support 版本。

如需相關資訊，請參閱第 463 頁的『MFT agent.properties 檔案』中的 **adminGroup** 內容。

在 IBM WebSphere MQ 7.5 及更新版本中，**fteSetAgentTraceLevel** 指令也會寫入代理程式程序控制器的追蹤。產生的追蹤檔案位於 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/pctrace%PID%/pctrace%PID%.txt` 中，其中 `%PID%` 是代理程式實例的程序 ID。

您也可以使用該指令，讓代理程式程序產生 Javacore。代理程式會在下列目錄中產生 Javacore 檔：`MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name`。

因為執行追蹤會大幅影響效能，而且會產生大量追蹤資料，因此，請謹慎執行追蹤，並且僅在必要時才執行。一般而言，只需要在 IBM 客戶服務代表要求時，再啟用追蹤即可。



小心：

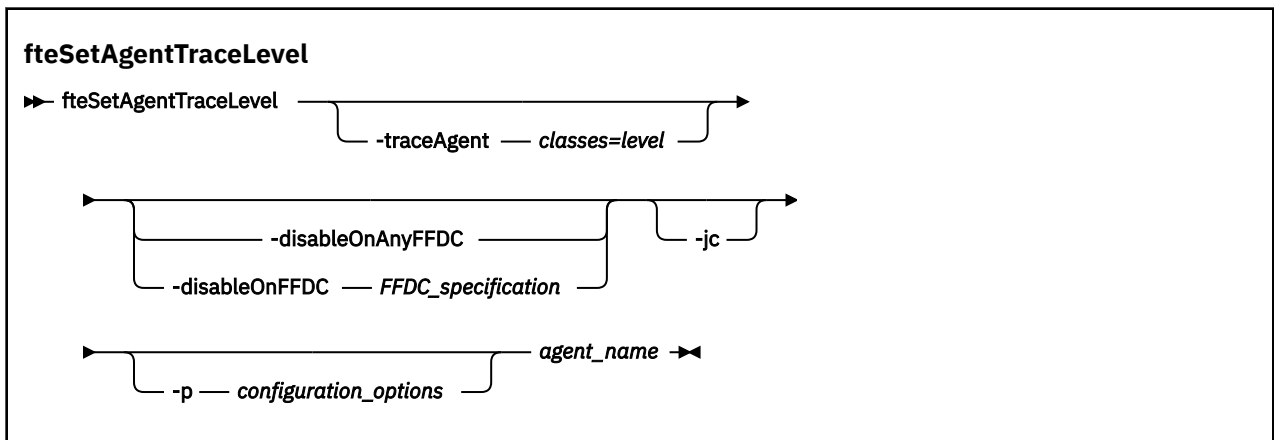
1. 您必須在執行代理程式的系統上執行此指令。
2. 在代理程式重新啟動時，追蹤及記載不會持續保存。

如果代理程式終止並由「處理程序控制器」處理程序重新啟動，則在更新 `agent.properties` 檔案以包括必要的追蹤及日誌內容之前，動態追蹤及日誌不會生效。

您可以在 `agent.properties` 檔中設定進一步追蹤內容，例如，追蹤檔大小及要保留的追蹤檔數目。[進階代理程式內容](#)中說明了這些內容。

僅在要使用一組不同於您的預設集的配置選項時，才為此指令指定選用的 `-p` 參數。如需相關資訊，請參閱第 463 頁的『MFT agent.properties 檔案』。

語法



參數

-traceAgent classes=level

必要項目。為代理程式追蹤設定的層次及追蹤適用的類別。

V 9.0.5 從 IBM MQ 9.0.5 開始，您可以指定類別規格清單（以冒號區隔）。此選項可讓您在不同層次設定不同的類別追蹤。例如：

```
fteSetAgentTraceLevel -traceAgent com.ibm.wmqfte.agent=all:com.ibm.wmqfte.cmdhandler=moderate
AGENT1
```

您仍可指定要套用追蹤層次的類別規格清單（以冒號區隔）。如果未指定此參數，則追蹤層次會套用至所有代理程式類別。請使用下列格式：

```
classes=level
```

例如：

```
com.ibm.wmqfte=all
```

您可以將 *classes* 替換為 Managed File Transfer 套件名稱，只追蹤特定的套件。不過，因為此選項只會擷取代理程式行為的子集，所以一般不建議您使用套件過濾。

如果 (*classes*) 的開頭有加號 (+)，則加號之後的追蹤類別清單，都會加入目前正在追蹤的任何現有追蹤類別中。

有效的追蹤層次選項如下，並且會依照追蹤檔案大小及其詳細資料遞增列出：

off

關閉代理程式追蹤，但繼續將資訊寫入日誌檔。這是預設選項。

flow

擷取與代理程式中的程序流程相關聯的追蹤點資料。

moderate

擷取追蹤中的適量診斷資訊。

verbose

擷取追蹤中的大量診斷資訊。

全部

設定對所有代理程式類別執行代理程式追蹤。

若要啟動代理程式的完整追蹤，請執行下列指令：

```
fteSetAgentTraceLevel -traceAgent =all AGENT_NAME
```

若要停止代理程式的完整追蹤，請執行下列指令：

```
fteSetAgentTraceLevel -traceAgent =off AGENT_NAME
```

-disableOnAnyFFDC

選用項目。如果已指定此參數，則代理程式產生「首次失敗資料擷取 (FFDC)」檔案時，會停用追蹤。

您只能指定其中一個 **-disableOnAnyFFDC** 及 **-disableOnFFDC** 參數。

-disableOnFFDC FFDC_specification

選用項目。如果已指定此參數，則代理程式產生與 *FFDC_specification* 相符的「首次失敗資料擷取 (FFDC)」檔案時，會停用追蹤。*FFDC_specification* 是值的清單（以逗點區隔）。值的格式可以為：

class_name

FFDC 所源自的類別名稱。例如，com.ibm.wmqfte.classA。

class_name:probe_ID

FFDC 所源自的類別中的類別名稱及位置的探測 ID。例如，com.ibm.wmqfte.classB:1。

您只能指定其中一個 **-disableOnAnyFFDC** 及 **-disableOnFFDC** 參數。

-jc

選用項目。要求代理程式產生 Javacore 檔。IBM 服務團隊可能要求您使用此參數來執行該指令，以協助診斷問題。此參數不能與 **-p** 以外的任何其他參數一起使用。

-p configuration_options

選用項目。此參數會決定用於設定代理程式追蹤層次的配置選項集。請使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，指令即會使用與此非預設協調佇列管理程式相關聯的一組內容檔。

如果未指定此參數，則會使用一組以預設協調佇列管理程式為基礎的配置選項。

agent_name

必要項目。您要為其設定追蹤層次的 Managed File Transfer Agent 的名稱。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，AGENT1 的所有類別的追蹤層次設定為 all：

```
fteSetAgentTraceLevel -traceAgent com.ibm.wmqfte=all AGENT1
```

在此範例中，AGENT1 的類別 `com.ibm.wmqfte.agent.Agent` 及 `com.ibm.wmqfte.cmdhandler` 的追蹤層次設定為 all：

```
fteSetAgentTraceLevel -traceAgent com.ibm.wmqfte.agent.Agent,com.ibm.wmqfte.cmdhandler=moderate AGENT1
```

在此範例中，從追蹤中排除子類別，因為 **-traceLevel** 參數設定為關閉。以詳細層次追蹤以 `com.ibm.outer` 開頭的所有類別（以 `com.ibm.outer.inner` 開頭的類別除外）：

```
fteSetAgentTraceLevel -traceAgent com.ibm.outer=verbose AGENT1  
fteSetAgentTraceLevel -traceAgent +com.ibm.outer.inner=off AGENT1
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

fteSetLoggerTraceLevel

使用 **fteSetLoggerTraceLevel** 指令可動態地修改 Managed File Transfer 日誌程式的現行追蹤層次。

用途

使用此指令來開啟及關閉日誌程式追蹤，或變更已設定的日誌程式追蹤的層次。使用 **fteSetLoggerTraceLevel** 指令時，您無需關閉並重新啟動日誌程式來修改追蹤層次。產生的追蹤檔案位於 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%/trace%PID%.txt` 中，其中 `%PID%` 是日誌程式實例的程序 ID。

在 IBM WebSphere MQ 7.5 版及更新版本中，**fteSetLoggerTraceLevel** 指令也會寫入日誌程式程序控制器的追蹤。產生的追蹤檔案位於 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/pctrace%PID%/pctrace%PID%.txt` 中，其中 `%PID%` 是日誌程式實例的程序 ID。

此指令也可以用於讓日誌程式程序產生 Javacore。日誌程式會在下列目錄中產生 Javacore 檔：
`MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name`。

因為執行追蹤會大幅影響效能，而且會產生大量追蹤資料，因此，請謹慎執行追蹤，並且僅在必要時才執行。一般而言，只需要在 IBM 客戶服務代表要求時，再啟用追蹤即可。



小心:

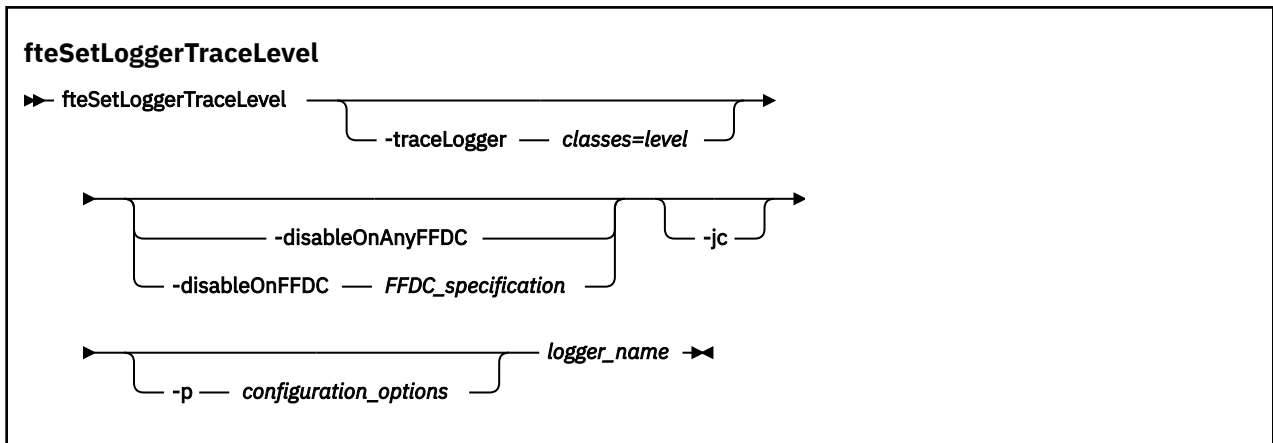
1. 您必須在執行代理程式的系統上執行此指令。
2. 在代理程式重新啟動時，追蹤及記載不會持續保存。

如果代理程式終止並由「處理程序控制器」處理程序重新啟動，則在更新 `agent.properties` 檔案以包括必要的追蹤及日誌內容之前，動態追蹤及日誌不會生效。

您可以在 `logger.properties` 檔案中，設定進一步的追蹤內容，例如，追蹤檔案大小及要保留的追蹤檔案數目。[日誌程式內容](#)中說明了這些內容。

僅在要使用一組不同於您的預設集的配置選項時，才為此指令指定選用的 `-p` 參數。如需相關資訊，請參閱第 99 頁的『MFT 日誌程式配置內容』。

語法



參數

-traceLogger classes=level

必要項目。設定日誌程式追蹤的層次，以及要套用追蹤的類別。

V 9.0.5 從 IBM MQ 9.0.5 開始，您可以指定類別規格清單（以冒號區隔）。此選項可讓您在不同層次設定不同的類別追蹤。例如：

```
fteSetLoggerTraceLevel -traceLogger com.ibm.wmqfte.logger=all:com.ibm.wmqfte.cmdhandler=moderate LOGGER1
```

您仍可指定要套用追蹤層次的類別規格清單（以冒號區隔）。如果未指定此參數，則追蹤層次會套用到所有代理程式類別。請使用下列格式：

```
classes=level
```

例如：

```
com.ibm.wmqfte=all
```

指定希望追蹤層次適用的類別規格清單（以逗號區隔）。如果未指定此參數，追蹤層次將適用於所有日誌程式類別。

如果 (`classes`) 以加號 (+) 開頭，加號後面的追蹤類別清單會新增至目前正在追蹤的所有現有追蹤類別。

有效的追蹤層次選項如下，並且會依照追蹤檔案大小及其詳細資料遞增列出：

off

關閉日誌程式追蹤，但繼續將資訊寫入日誌檔。這是預設選項。

flow

擷取與日誌程式中的處理流程相關聯的追蹤點資料。

moderate

擷取追蹤中的適量診斷資訊。

verbose

擷取追蹤中的大量診斷資訊。

全部

設定對所有日誌程式類別執行日誌程式追蹤。

-disableOnAnyFFDC

選用項目。如果已指定此參數，則日誌程式產生「首次失敗資料擷取 (FFDC)」檔案時，會停用追蹤。

您只能指定其中一個 **-disableOnAnyFFDC** 及 **-disableOnFFDC** 參數。

-disableOnFFDC FFDC_specification

選用項目。如果已指定此參數，則日誌程式產生與 *FFDC_specification* 相符的「首次失敗資料擷取 (FFDC)」檔案時，會停用追蹤。*FFDC_specification* 是值的清單（以逗點區隔）。此值可以是下列其中一種格式：

class_name

FFDC 所源自的類別名稱。例如，`com.ibm.wmqfte.classA`。

class_name:probe_ID

FFDC 所源自的類別中的類別名稱及位置的探測 ID。例如，`com.ibm.wmqfte.classB:1`。

您只能指定其中一個 **-disableOnAnyFFDC** 及 **-disableOnFFDC** 參數。

-jc

選用項目。要求日誌程式產生 Javacore 檔。IBM 服務團隊可能要求您執行搭配此參數的指令，以協助診斷問題。您無法使用 **-jc** 參數來搭配任何其他參數。

-p configuration_options

選用項目。此參數會決定一組用於設定日誌程式追蹤層次的配置選項。請使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，指令即會使用與此非預設協調佇列管理程式相關聯的一組內容檔。

如果未指定此參數，則會使用一組以預設協調佇列管理程式為基礎的配置選項。

logger_name

必要項目。您要為其設定追蹤層次的 Managed File Transfer Logger 的名稱。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，LOGGER1 的所有類別的追蹤層次設定為 all：

```
fteSetLoggerTraceLevel -traceLogger com.ibm.wmqfte=all LOGGER1
```

在此範例中，LOGGER1 的類別 `com.ibm.wmqfte.logger.logger` 及 `com.ibm.wmqfte.cmdhandler` 的追蹤層次設定為 all：

```
fteSetLoggerTraceLevel -traceLogger com.ibm.wmqfte.logger.logger,com.ibm.wmqfte.cmdhandler=moderate
LOGGER1
```

在此範例中，從追蹤中排除子類別，因為 **-traceLevel** 參數設定為關閉。以詳細層次追蹤以 `com.ibm.outer` 開頭的所有類別（以 `com.ibm.outer.inner` 開頭的類別除外）：

```
fteSetLoggerTraceLevel -traceLogger com.ibm.outer=verbose LOGGER1
fteSetLoggerTraceLevel -traceLogger +com.ibm.outer.inner=off LOGGER1
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

z/OS V 9.0.3 fteSetProductId: 設定 z/OS SCRT 錄製產品 ID

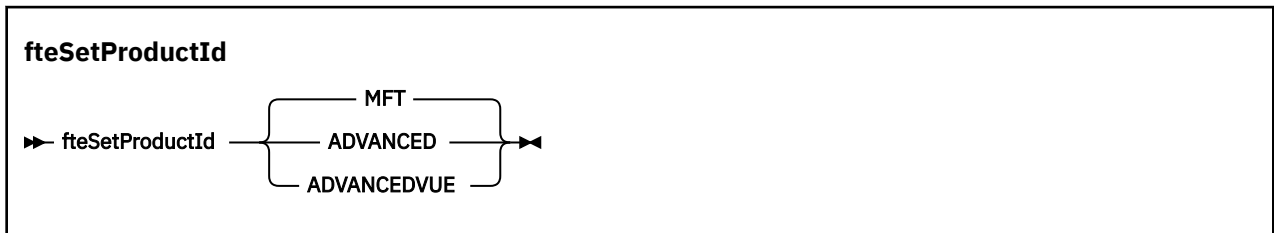
使用 **fteSetProductId** 來設定要針對安裝進行 Managed File Transfer 用法錄製的產品類型。此指令僅適用於 z/OS。

用途

在至少定義了一個協調佇列管理程式後，或者為安裝建立了 `installation.properties` 檔後，可以隨時執行此指令。

如需產品使用情形記錄的相關資訊，請參閱 [報告產品資訊](#)。

語法



參數

用法錄製的產品類型：

請指定下列其中一項：

MFT

用法錄製為獨立式 Managed File Transfer 產品，其產品 ID 為 5655-MF9。

ADVANCED

用法錄製為 IBM MQ Advanced for z/OS 產品的一部分，其產品 ID 為 5655-AV9。

ADVANCEDVUE

用法錄製為 IBM MQ Advanced for z/OS, Value Unit Edition 產品的一部分，其產品 ID 為 5655-AV1。

回覆碼

0

指令已順利完成。

1

指令未順利結束，或未將產品類型設定為所要求的值。

相關工作

第 82 頁的『[配置 MFT 的協調佇列管理程式](#)』

執行 **fteSetupCoordination** 指令之後，請執行 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` 目錄中的 `coordination_qmgr_name.mqsc` Script，以執行協調佇列管理程式所需的配置。但如果要手動執行此配置，請在協調佇列管理程式上執行下列步驟。

fteSetupCommands: 建立 MFT command.properties 檔

fteSetupCommands 指令會建立 Managed File Transfer `command.properties` 檔案。此內容檔針對您發出指令時連接至 IBM MQ 網路的佇列管理程式，指定其詳細資料。

重要: **ULW** 在 IBM MQ for UNIX, Linux, and Windows 上，只有 IBM MQ 管理者（及 mqm 群組的成員）身分的使用者才能執行此指令。如果您嘗試以非 IBM MQ 管理者的使用者身分執行此指令，則會收到錯誤訊息 BFGCL0502E：您未獲授權執行所要求的作業。，且指令將不會執行。

z/OS 在 z/OS 系統上，使用者必須滿足（至少）下列其中一項條件才能執行移轉指令：

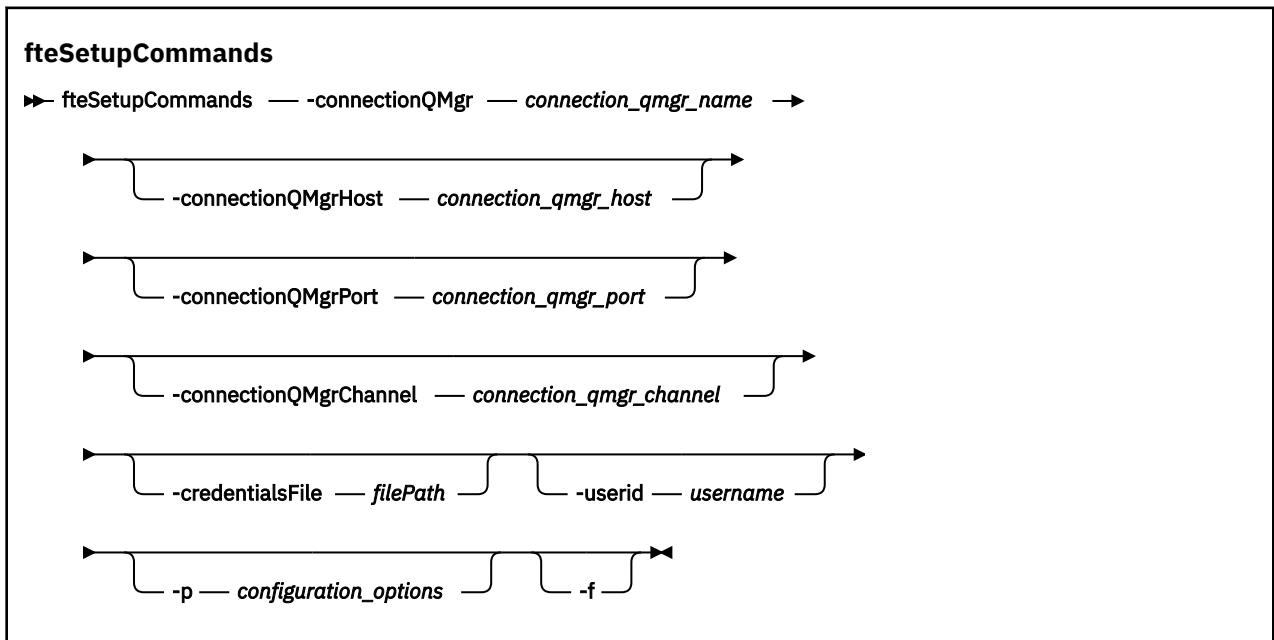
- 是 mqm 群組的成員（如果 mqm 群組已定義在系統上）。
- **V9.0.0.1** 是 BFG_GROUP_NAME 環境變數所指名之群組的成員（如果已指名群組的話）。
- **V9.0.0.1** 指令執行時，BFG_GROUP_NAME 環境變數中未設定任何值。

用途

使用 **fteSetupCommands** 指令，可在協調佇列管理程式配置目錄中建立 `command.properties` 檔案。此指令使用 `MQ_DATA_PATH` 環境變數和 `installation.properties` 檔案，來決定 `command.properties` 檔案所在的位置。發出 **fteSetupCommands** 指令之前，請確定您已建立及配置協調佇列管理程式。

如需內容檔的相關資訊，請參閱 [第 460 頁的『MFT command.properties 檔』](#)。

語法



參數

-connectionQMGr (connection_qmgr_name)

必要項目。用來連接至 IBM MQ 網路以發出指令的佇列管理程式名稱。

-connectionQMGrHost (connection_qmgr_host)

選用項目。連線佇列管理程式的主機名稱或 IP 位址。

如果未指定 **-connectionQMgrHost** 參數，則會採用連結模式連線。因此，如果要使用用戶端模式連線，此參數是必要項目。

如果指定了 **-connectionQMgrHost** 參數的值，但未指定 **-connectionQMgrPort** 及 **-connectionQMgrChannel** 內容的值，則預設會使用 1414 埠號及 SYSTEM.DEF.SVRCONN 通道。

-connectionQMgrPort (connection_qmgr_port)

選用項目。在用戶端模式中用來連接至連線佇列管理程式的埠號。如果指定了 **-connectionQMgrPort** 參數，則必須同時指定 **-connectionQMgrHost** 參數。

-connectionQMgrChannel (connection_qmgr_channel)

選用項目。用來連接至連線佇列管理程式的通道名稱。如果指定了 **-connectionQMgrChannel** 參數，則必須同時指定 **-connectionQMgrHost** 參數。

-p (configuration_options)

選用項目。此參數決定一組用來設定指令佇列管理程式的配置選項。請使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，**fteSetupCommands** 指令即會使用與此非預設協調佇列管理程式相關聯的一組內容檔。

如果未指定此參數，則會使用一組以預設協調佇列管理程式為基礎的配置選項。

-credentialsFile (filePath)

選用項目。現有或新認證檔的完整檔案路徑，IBM MQ 鑑別詳細資料會新增到該認證檔中。

此指令支援將 IBM MQ 鑑別詳細資料的集合新增至名為 Managed File Transfer 的認證檔。已啟用 IBM MQ 連線鑑別時，請使用此指令。如果更新現有的詳細資料，則必須使用 **-f** 強制參數。

-userid (username)

選用項目。用來與認證詳細資料相關聯的使用者 ID。如果未指定使用者 ID，則認證詳細資料將套用至所有使用者。您還必須指定 **-credentialsFile** 參數。

-f

選用項目。強制以此指令中指定的詳細資料來改寫現有的 `command.properties` 檔。

-? 或 -h

選用項目。顯示指令語法。

範例

```
fteSetupCommands -connectionQMgr QM_NEPTUNE -connectionQMgrHost 9.146.157.241  
-connectionQMgrPort 1414 -connectionQMgrChannel SYSTEM.DEF.SVRCONN
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

相關參考

第 460 頁的『[MFT command.properties 檔](#)』

`command.properties` 檔指定當您發出指令時要連接的指令佇列管理程式，以及 Managed File Transfer 聯絡該佇列管理程式所需的資訊。

第 433 頁的『[fteSetupCoordination](#)』

fteSetupCoordination 指令會建立 Managed File Transfer 的內容檔及協調佇列管理程式目錄。

fteSetupCoordination

fteSetupCoordination 指令會建立 Managed File Transfer 的內容檔及協調佇列管理程式目錄。

重要: **ULW** 在 IBM MQ for UNIX, Linux, and Windows 上, 只有 IBM MQ 管理者 (及 mqm 群組的成員) 身分的使用者才能執行此指令。如果您嘗試以非 IBM MQ 管理者的使用者身分執行此指令, 則會收到錯誤訊息 BFGCL0502E: 您未獲授權執行所要求的作業。 , 且指令將不會執行。

z/OS 在 z/OS 系統上, 使用者必須滿足 (至少) 下列其中一項條件才能執行移轉指令:

- 是 mqm 群組的成員 (如果 mqm 群組已定義在系統上)。
- **V9.0.0.1** 是 BFG_GROUP_NAME 環境變數所指名之群組的成員 (如果已指名群組的話)。
- **V9.0.0.1** 指令執行時, BFG_GROUP_NAME 環境變數中未設定任何值。

用途

使用 **fteSetupCoordination** 指令建立下列 Managed File Transfer 物件:

- 協調佇列管理程式目錄
- 資料目錄 mqft (如果不存在的話)
- installation.properties 檔案
- coordination.properties 檔案

這個指令也提供下列 MQSC 指令, 您必須對協調佇列管理程式執行這些指令來配置 Managed File Transfer。MQSC 指令會建立主題、主題字串、SYSTEM.FTE 佇列及預設資料庫日誌程式佇列。這些指令還會更新名稱清單, 並將協調佇列管理程式的 PSMODE 屬性設定為 ENABLED。

z/OS 如果協調佇列管理程式位於 z/OS 上, 則在執行這些 MQSC 指令之前, 必須先確保下列必要物件已存在:

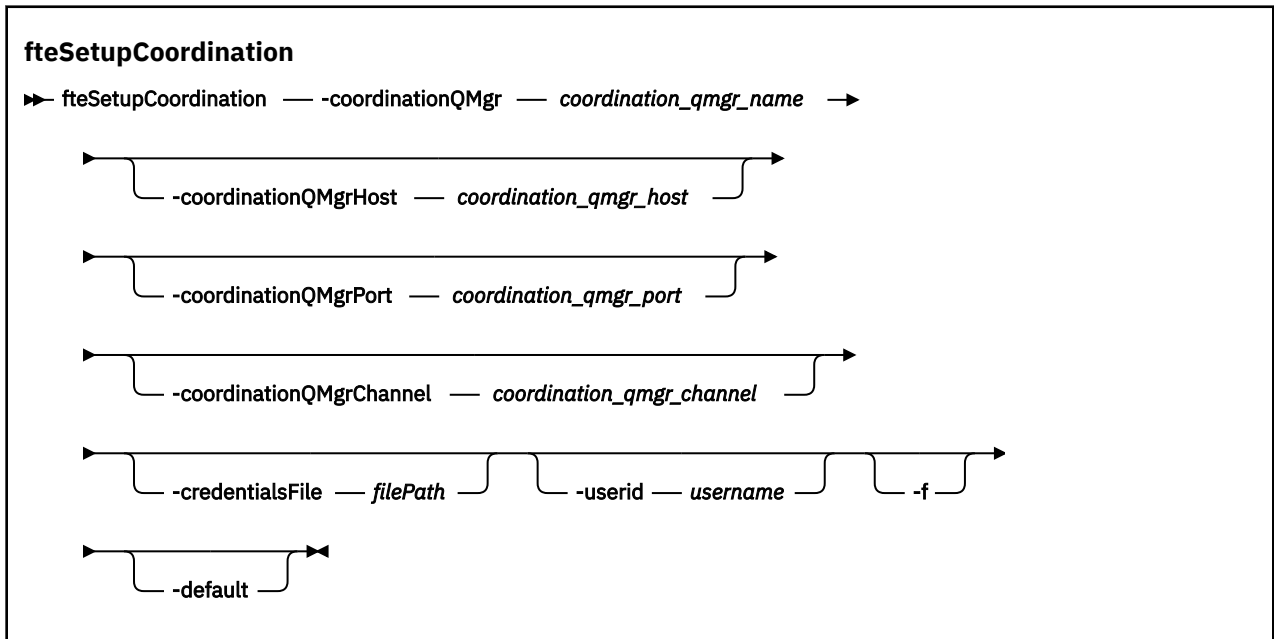
- SYSTEM.BROKER.DEFAULT.STREAM 佇列
- SYSTEM.QPUBSUB.QUEUE.NAMELIST 名稱清單
- SYSTEM.BROKER.DEFAULT.STREAM 和 SYSTEM.BROKER.ADMIN.STREAM 串流

```
DEFINE TOPIC('SYSTEM.FTE') TOPICSTR('SYSTEM.FTE') REPLACE
ALTER TOPIC('SYSTEM.FTE') NPMMSGDLV(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)
```

如需內容檔的相關資訊, 請參閱[配置選項](#)。

z/OS 如果您使用的是 z/OS, 則可以透過產生自 Managed File Transfer 指令範本 PDSE 程式庫資料集的 Script, 從 JCL 發出 **fteSetupCoordination** 指令及其他指令。如需相關資訊, 請參閱第 54 頁的『[建立 MFT 代理程式或日誌程式指令資料集](#)』。

語法



參數

-coordinationQMGr (*coordination_qmgr_name*)

必要項目。協調佇列管理程式的名稱。此佇列管理程式必須是 IBM WebSphere MQ 7.0 或更新版本的佇列管理程式。

-coordinationQMGrHost (*coordination_qmgr_host*)

選用項目。協調佇列管理程式的主機名稱或 IP 位址。

如果未指定 **-coordinationQMGrHost** 參數，則會採用連結模式連線。

如果指定了 **-coordinationQMGrHost** 參數的值，但未指定 **-coordinationQMGrPort** 及 **-coordinationQMGrChannel** 參數的值，則預設會使用 1414 埠號及 SYSTEM.DEF.SVRCONN 通道。

-coordinationQMGrPort (*coordination_qmgr_port*)

選用項目。用於與協調佇列管理程式進行用戶端連線的埠號。如果指定了 **-coordinationQMGrPort** 參數，則必須同時指定 **-coordinationQMGrHost** 參數。

-coordinationQMGrChannel (*coordination_qmgr_channel*)

選用項目。用來連接至協調佇列管理程式的通道名稱。如果指定了 **-coordinationQMGrChannel** 參數，則必須同時指定 **-coordinationQMGrHost** 參數。

-credentialsFile (*filePath*)

選用項目。現有或新認證檔的完整檔案路徑，IBM MQ 鑑別詳細資料會新增到該認證檔中。

此指令支援將 IBM MQ 鑑別詳細資料的集合新增至名為 Managed File Transfer 的認證檔。已啟用 IBM MQ 連線鑑別時，請使用此指令。如果更新現有的詳細資料，則必須使用 **-f** 強制參數。

-userid (*username*)

選用項目。用來與認證詳細資料相關聯的使用者 ID。如果未指定使用者 ID，則認證詳細資料將套用至所有使用者。您還必須指定 **-credentialsFile** 參數。

-f

選用項目。強制以此指令中指定的詳細資料來改寫現有的協調佇列管理程式配置。

-default

選用項目。將預設配置選項更新為與此指令中指定的協調佇列管理程式相關聯的選項。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，對於稱為 QM_SATURN 的協調佇列管理程式（以用戶端模式連接）設定必要物件：

```
fteSetupCoordination -coordinationQMGr QM_SATURN  
-coordinationQMGrHost myhost.ibm.com -coordinationQMGrPort 1415  
-coordinationQMGrChannel SYSTEM.DEF.SVRCONN
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

相關概念

第 49 頁的『[多平台上的 MFT 配置選項](#)』

Managed File Transfer 提供了一組內容檔，這些檔案包含有關您的設定的金鑰資訊，是作業所必要的。這些內容檔位於安裝產品時所定義的配置目錄中。

第 60 頁的『[在 z/OS 上配置 MQMFTCredentials.xml](#)』

如果在已啟用安全的情況下配置 Managed File Transfer (MFT)，則連線鑑別需要所有 MFT 代理程式以及連接至佇列管理程式的指令，才能提供使用者 ID 和密碼認證。

相關工作

第 82 頁的『[配置 MFT 的協調佇列管理程式](#)』

執行 **fteSetupCoordination** 指令之後，請執行 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` 目錄中的 `coordination_qmgr_name.mqsc` Script，以執行協調佇列管理程式所需的配置。但如果要手動執行此配置，請在協調佇列管理程式上執行下列步驟。

相關參考

第 463 頁的『[MFT agent.properties 檔案](#)』

每一個 Managed File Transfer Agent 都有自己的內容檔 `agent.properties`，其中必須包含代理程式用來連接至其佇列管理程式的資訊。`agent.properties` 檔案也可以包含變更代理程式行為的內容。

相關資訊

第 510 頁的『[coordination.properties 檔的 SSL 內容](#)』

fteShowAgentDetails

使用 **fteShowAgentDetails** 指令顯示特定 Managed File Transfer Agent 的詳細資料。這些是代理程式的 Managed File Transfer 協調佇列管理程式儲存的詳細資料。

用途

您可以從任何可連接至協調佇列管理程式的系統執行 **fteShowAgentDetails** 指令。此指令會使用 `coordination.properties` 檔連接至協調佇列管理程式。

僅在要使用一組不同於您的預設集的配置選項時，才為此指令指定選用的 **-p** 參數。如需相關資訊，請參閱第 49 頁的『[多平台上的 MFT 配置選項](#)』。

從代理程式發佈至 SYSTEM.FTE 主題的狀態訊息，產生此指令所產生的代理程式狀態資訊。這些訊息在第 625 頁的『[MFT 代理程式狀態訊息格式](#)』中說明。**fteShowAgentDetails** 指令所產生的狀態資訊，指出前次發佈狀態訊息時的代理程式狀態。這些狀態訊息的頻率視 `agentStatusPublishRateLimit` 內容值而定。如需相關資訊，請參閱第 463 頁的『[MFT agent.properties 檔案](#)』。

若是 IBM WebSphere MQ 7.5.0 Fix Pack 1 或更新版本，如果您想查看本端代理程式的診斷資訊，請為此指令指定選用的 **-d** 參數。此資訊包括現行傳送、排程的傳送、監視器及代理程式佇列的深度。您可以利用此資訊判斷本端代理程式的性能及狀態。

V 9.0.2 z/OS V 9.0.0.1 對於 z/OS, 從 IBM MQ 9.0.2 及 IBM MQ 9.0.0 Fix Pack 1, 僅當 **fteShowAgentDetails** 指令由下列指令執行時, 才能指定 **-d** 參數:

- 以其身分執行代理程式程序的使用者 ID。
- 代理程式內容 **adminGroup** 所指定的群組成員。

如需相關資訊, 請參閱 第 463 頁的『MFT agent.properties 檔案』中的 **adminGroup** 內容。

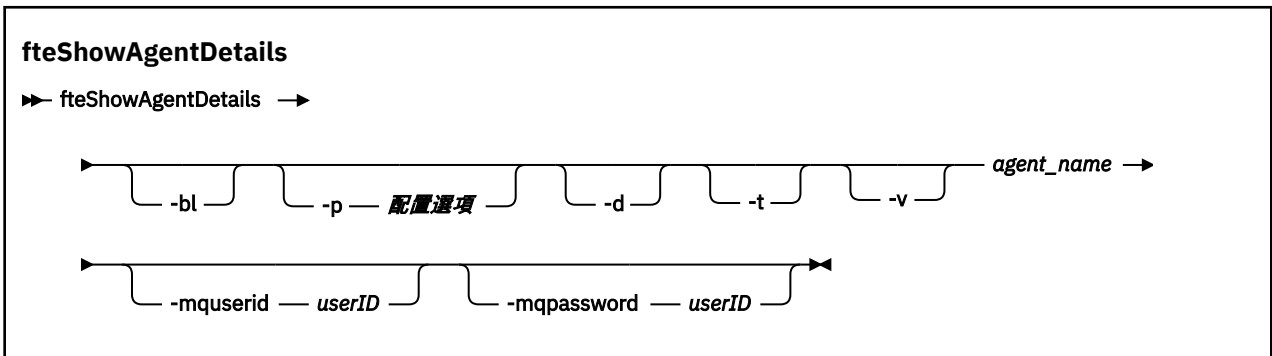
若為 IBM WebSphere MQ 7.5 或更新版本, 如果在與代理程式相同的系統上執行此指令, 則可取得代理程式程序控制器及佇列管理程式的狀態。您可以利用此資訊來協助判斷問題。此外, 當您執行指令的系統和代理程式所在的系統相同時, 如果代理程式非預期地結束, 則可以取得更詳細的代理程式狀態資訊。

如需可能的代理程式狀態值及其意義的清單, 請參閱第 525 頁的『MFT 代理程式狀態值』。

如需代理程式程序控制器的可能狀態值及其意義的清單, 請參閱第 528 頁的『MFT 代理程式程序控制器狀態值』。

如需代理程式追蹤值與 FFDC 規格及其意義的清單, 請參閱 第 229 頁的『fteSetAgentTraceLevel』。

語法



參數

-bl

選用項目。額外輸出代理程式的產品建置層次。

-p (configuration_options)

選用項目。此參數決定一組用來發出要求以顯示代理程式詳細資料的配置選項。請使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後, 指令即會使用與此非預設協調佇列管理程式相關聯的一組內容檔。

如果未指定此參數, 則會使用一組以預設協調佇列管理程式為基礎的配置選項。

-d

選用項目。指定收集 *agent_name* 的診斷資訊。

診斷資訊會輸出至主控台, 並寫入

diagnostics.<yyyyMMdd>.<HHmmss>.<ssss>.<number>.properties 檔案 (位於目錄 MQ_DATA_PATH\mqft\logs\coordination_qmgr_name\agents\agent_name\logs) 中。將最多建立五個歷程檔案來包含代理程式的相關診斷資訊。如果在執行指定了 **-d** 參數的 **fteShowAgentDetails** 指令時, 已為代理程式建立五個歷程檔案, 則將會刪除最舊的歷程檔案並將其取代為包含代理程式最新診斷資訊的新檔案。

您只能在代理程式正在執行中且其位於本端系統上時, 才可使用此參數。

-t

選用項目。指定簡要模式。在 IBM MQ 9.0.2 中, 依預設, 輸出會包含 **Status Age** 資訊。如果您不想看到此資訊, 您可以發出搭配 **-t** 參數的指令。如需相關資訊, 請參閱第 255 頁的『代理程式顯示為處於 UNKNOWN 狀態時要執行的動作』。

-v

選用項目。指定詳細模式，以產生代理程式的其他輸出。這些包括主機名稱、產品版本、產品建置層次、追蹤層次，及「首次失敗資料擷取 (FFDC)」規格，以及每一個現行來源和目的地傳送的傳送狀態清單。

現行傳送資訊取自代理程式狀態發佈，第 625 頁的『MFT 代理程式狀態訊息格式』中提供說明。因此，只有在 `agentStatusPublishRateLimit` 內容值之內，此傳送資訊才正確。如需此內容的詳細資料，請參閱第 463 頁的『MFT `agent.properties` 檔案』。

agent_name

必要項目。您要顯示的 Managed File Transfer Agent 名稱。

-mquserid (userID)

選用項目。指定用來向協調佇列管理程式進行鑑別的使用者 ID。

-mqpassword (password)

選用項目。指定用來向協調佇列管理程式進行鑑別的密碼。您還必須指定 `-mquserid` 參數。如果指定了 `-mquserid`，卻未指定 `-mqpassword`，則系統會提示您提供相關聯的密碼。將不會顯示密碼。

-? 或 -h

選用項目。顯示指令語法。

範例

在下列範例中，執行連結代理程式，在本端對代理程式發出 `fteShowAgentDetails` 指令：

```
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
414d51204d49414f572020202020202020822c5b4a648c0b20	progress
414d51204d49414f572020202020202020822c5b4a346c0b20	progress

在下列範例中，QMGR1 是作為配置選項輸入的非預設協調佇列管理程式，且使用 **-d** 參數要求診斷資訊。
fteShowAgentDetails 指令在具有本端代理程式的 IBM WebSphere MQ 7.5.0 Fix Pack 1 系統上發出：

```
fteShowAgentDetails -p QMGR1 -d AGENT1
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
Agent Information:
  Name: AGENT1
  Type: Standard
  Description:
  Operating System: Linux
  Time Zone: Greenwich Mean Time

Agent Controller Information:
  Controller type: MQMFT Process Controller
  Status: STARTED
  Status Details: The agent process controller has started
                  the agent process.
  Agent Restarts within Interval: 0
  Total Agent Restart Count: 0

Agent Availability Information:
  Status: ACTIVE
  Status Details: The agent is running and is publishing
                  its status at regular intervals. The last
                  update was received within the expected
                  time period. The agent is currently
                  processing one or more transfers.

Queue Manager Information:
  Name: QMGR1
  Transport: Client
  Host: host1.hursley.ibm.com
  Port: 2021
  Channel: SYSTEM.DEF.SVRCONN
  Last Status Reported: UNKNOWN
  Status Details: Information about the queue manager is
                  not available because the agent has a
                  client connection to the queue manager.

Agent Diagnostic Information:

Command Handler Diagnostics:
  Last Command Queue Read Time: 2012-07-30T15:23:10.705Z
  Pending Command Queue Size: 0
  Last Internal Command Type: Resync Request (from sender) -
414d5120514d43414e4445202020202079e20f5064230010
  Last Internal Command Time: 2012-07-30T14:17:10.506Z
  Last External Command Type: New Monitor Request
  Last External Command Time: 2012-07-30T14:10:57.751Z
  Diagnostic Properties File name: C:\Program Files (x86)\IBM\WebSphere
MQ\mqft\logs\MUNGEE\agents\MUNGEE\logs\di
agnostics.20121031.083420.0477.1.properti
es

Command Handler Worker Thread 0 Diagnostics:
  Status: Waiting

Command Handler Worker Thread 1 Diagnostics:
  Status: Waiting

Command Handler Worker Thread 2 Diagnostics:
  Status: Waiting

Command Handler Worker Thread 3 Diagnostics:
  Status: Waiting

Command Handler Worker Thread 4 Diagnostics:
  Status: Waiting

File Transfer Diagnostics:
  Source Transfers: 1
  Destination Transfers: 2

File Transfer 0 Diagnostics:
  Transfer Id: 414d5120514d43414e44452020202079e20f5064230010
  Role: SOURCE
```

```

State: ReSynchronisingTransfer
Status: INACTIVE
Start Time: Not started
Retry Count: 0
CheckPoint Index: 0
CheckPoint Position: 0

File Transfer 1 Diagnostics:
Transfer Id: 414d5120514d43414e44452020202020c8fbd54f144f0d20
Role: DESTINATION
State: RunningTransfer
CheckPoint Index: 0
CheckPoint Position: 0
Write Index: 0
Write Position: 0

File Transfer 2 Diagnostics:
Transfer Id: 414d5120514d43414e4445202020202079e20f5086020010
Role: DESTINATION
State: RunningTransfer
CheckPoint Index: 9
CheckPoint Position: 0
Write Index: 3
Write Position: 140923

Monitor 0 Diagnostics:
Name: MONITOR1
Status: STARTED
Resource Type: directory
Resource: /tmp/monitor
Poll Interval: 1 minutes
Batch Size: 2
Condition: Match
Pattern: * (wildcard)
Executing: false
Last Execute Start Time: 2012-04-04T16:19:01.852Z
Last Execute End Time: 2012-04-04T16:19:01.852Z
Last Execute Match Count: 0

Schedule 1 Diagnostics:
Id: 1
Next Trigger Time: 2012-07-17T16:00+0100
Occurrences So Far: 14
Repeat Interval: hours
Repeat Frequency: 5
Source Agent: AGCANDE
Destination Agent: AGCANDE
Source File: /tmp/source/a.txt, ...
Destination File: /tmp/dest/a.txt, ...

```

在下列範例中，停止連結代理程式，從遠端的代理程式發出 **fteShowAgentDetails** 指令：

```

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.
```

在下列範例中，代理程式佇列管理程式已停止，連結代理程式正在等待重新啟動。代理程式已在 **Total Agent Restart Count: 1** 之前重新啟動一次，可能是因為先前重新啟動代理程式佇列管理程式：

註：針對 Last Status Reported 的 Last Error MQRC，以取得佇列管理程式資訊；即使佇列管理程式變成可用，此資訊仍會保留。

```
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.
```

在下列範例中，用戶端模式代理程式剛剛非預期地結束，而代理程式程序控制器嘗試在 **maxRestartDelay** 代理程式內容值所指定的延遲時間之後重新啟動，使狀況回復。預設 **maxRestartDelay** 代理程式內容值為 -1，這會導致代理程式處理程序控制器終止；因此在此範例中，**maxRestartDelay** 內容值必須設為大於 0 的值。**Current Agent Restart Count: 4** 暗示在 **maxRestartInterval** 代理程式內容時段內已重新啟動 4 次。如果 **maxRestartCount** 代理程式內容是 4，則在 **maxRestartInterval** 內重新啟動 4 次之後，代理程式程序控制器在重新啟動代理程式之前會等待 **maxRestartDelay** 秒，就像這裡的案例一樣。**Total Agent restart Count: 8** 表示這之前已發生過。此範例並非典型範例，只有在代理程式記憶體不足或自訂使用者結束程式造成某種執行時期錯誤時，代理程式才可能非預期地結束。有關代理程式非預期結束的原因，在代理程式的 **output0.log** 檔案中會有完整資料：

```
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

```

在下列範例中，顯示 Connect:Direct 橋接器代理程式的結果：

```

fteShowAgentDetails AG_CD1
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
Agent Information:
Name:                          AG_CD1
Type:                          Connect:Direct bridge
Description:
Connect:Direct Node Name:      CDNODE
Connect:Direct Node Host:      localhost:1363
Operating System:              Windows Server 2003
Time Zone:                     Greenwich Mean Time


Agent Controller Information:
Controller type:                MQMFT Process Controller
Status:                        UNKNOWN
Status Details:                Information about the agent controller
                                is not available, either because the
                                agent is not running or the agent is
                                running on a different system.

Agent Restarts within Interval: 0
Total Agent Restart Count:     0

Agent Availability Information:
Status:                        STOPPED
Status Details:                The agent has been stopped. It was shut
                                down in a controlled manner.

Queue Manager Information:
Name:                          QM_JUPITER
Transport:                      Bindings
Last Status Reported:          UNKNOWN
Status Details:                Information about the queue manager is
                                not available, either because the agent
                                is not running or the agent is running
                                on a different system.

```

 在下列範例中，執行於 z/OS 上的代理程式會向 Automatic Restart Manager (ARM) 進行登錄：

```

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 (ELEMENTYPE: 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.

回覆碼

0

指令已順利完成。

1

指令未順利結束。

相關參考

第 398 頁的『[fteListAgents](#): 列出協調佇列管理程式的 MFT 代理程式』

使用 **fteListAgents** 指令，以列出向特定協調佇列管理程式登錄的所有 Managed File Transfer 代理程式。

第 525 頁的『[MFT 代理程式狀態值](#)』

fteListAgents 及 **fteShowAgentDetails** 指令可產生代理程式狀態資訊。此狀態有數個可能的值。

第 528 頁的『[MFT 代理程式程序控制器狀態值](#)』

fteShowAgentDetails 指令會產生代理程式程序控制器狀態資訊。此狀態有數個可能的值。

fteShowLoggerDetails

使用 **fteShowLoggerDetails** 指令可顯示特定 Managed File Transfer 日誌程式的詳細資料。

用途

您必須在與日誌程式相同的系統上執行 **fteShowLoggerDetails** 指令。它會顯示日誌程式程序控制器及日誌程式佇列管理程式的狀態，讓您用來判斷問題。**fteShowLoggerDetails** 指令會列出特定 Managed File Transfer 日誌程式的下列詳細資料：

- 日誌程式控制器狀態。
- 日誌程式在時間間隔內重新啟動
- 日誌程式重新啟動總次數
- 日誌程式可用性狀態
- 日誌程式佇列管理程式名稱
- 日誌程式佇列管理程式傳輸類型
- 報告的日誌程式佇列管理程式前次狀態（僅適用於連結傳輸模式）

僅在要使用一組不同於您的預設集的配置選項時，才為此指令指定選用的 **-p** 參數。如需相關資訊，請參閱第 49 頁的『[多平台上的 MFT 配置選項](#)』。

如需可能的日誌程式狀態值及其意義的清單，請參閱第 528 頁的『[MFT 日誌程式狀態值](#)』。

如需日誌程式程序控制器的可能狀態值及其意義的清單，請參閱第 529 頁的『[MFT 日誌程式程序控制器狀態值](#)』。

V 9.0.4

從 IBM MQ 9.0.4 開始，指令輸出會顯示日誌程式用來連接至佇列管理程式的連線資訊。

V 9.0.4

如果是以用戶端模式連接日誌程式，則下列輸出：

Last Status Reported

會顯示為 UNKNOWN

Status Details

會顯示為 Information about the queue manager is not available because the logger has a client connection to queue manager.

語法

fteShowLoggerDetails

```
► fteShowLoggerDetails ──────────── logger_name ◄◄  
    └─ -p ── 配置選項 ───────────┘
```

參數

-p configuration_options

選用項目。此參數可決定一組用來發出要求以顯示日誌程式詳細資料的配置選項。請使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，指令即會使用與此非預設協調佇列管理程式相關聯的一組內容檔。

如果未指定此參數，則會使用一組以預設協調佇列管理程式為基礎的配置選項。

logger_name

必要項目。您要顯示的 Managed File Transfer 日誌程式名稱。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，啟動的日誌程式在本端對日誌程式發出 **fteShowLoggerDetails** 指令：

```
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.
```

在此範例中，因佇列管理程式無法使用而等待的日誌程式，在本端對日誌程式發出 **fteShowLoggerDetails** 指令：

```
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.

```

在本範例中，z/OS 上有一個正在執行的日誌程式（未向 ARM 登錄）：

```
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.

```

在本範例中，在 z/OS 上有一個未執行或在其他系統上執行的日誌程式：

```
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.

```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

相關參考

第 528 頁的『MFT 日誌程式狀態值』

fteShowLoggerDetails 指令會產生日誌程式狀態資訊。此狀態有數個可能的值。

第 529 頁的『MFT 日誌程式程序控制器狀態值』

fteShowLoggerDetails 指令會產生日誌程式程序控制器狀態資訊。此狀態有數個可能的值。

fteStartAgent

fteStartAgent 指令會從指令行啟動 Managed File Transfer 代理程式。

用途

使用 **fteStartAgent** 指令來啟動 Managed File Transfer 代理程式。必須先啟動代理程式，才能將它用來執行檔案傳送。**fteStartAgent** 指令會在您發出指令的系統上啟動代理程式：您無法在遠端系統上啟動代理程式。

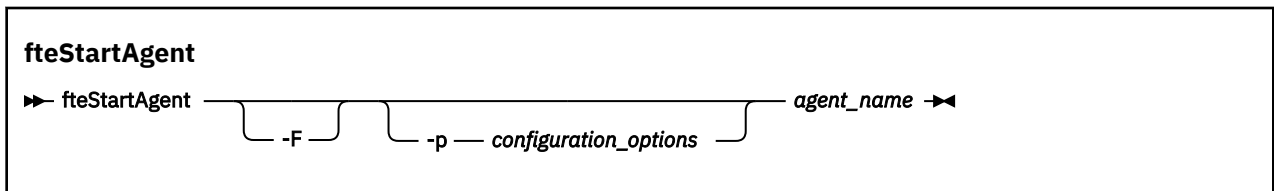
若為 IBM WebSphere MQ 7.5 或更新版本，代理程式程序控制器將管理代理程式的啟動。不過，代理程式程序控制器可能會等待一段時間（例如，代理程式的失敗率很高時），再嘗試重新啟動代理程式。作為 IBM MQ 管理者，您可以使用 **fteStartAgent** 指令來置換這項等待，並起始啟動代理程式。如果代理程式程序控制器在等待佇列管理程式變成可用，這個指令也會起始代理程式程序控制器以嘗試重新連接佇列管理程式。

Windows 如果您已使用 **fteCreateAgent** 或 **fteModifyAgent** 指令，將代理程式配置成以 Windows 服務方式執行，則執行 **fteStartAgent** 指令會啟動 Windows 服務。

如果代理程式未動或已啟動，此指令會傳回錯誤。代理程式會根據 `agent.properties` 檔案中定義的值，與其佇列管理程式進行通訊。

請僅在要使用一組不同於您的預設集的配置選項時，才為此指令指定選用的 **-p** 參數。如需相關資訊，請參閱第 463 頁的『MFT `agent.properties` 檔案』。

語法



參數

-F

選用項目。此參數以前景程序方式來執行代理程式常駐程式。依預設是在背景中執行代理程式常駐程式。

如果您是在 Windows 上執行，且已使用 **fteCreateAgent** 或 **fteModifyAgent** 指令將代理程式配置成以 Windows 服務方式執行，則 **-F** 參數會置換此配置。

-p configuration_options

選用項目。此參數決定一組用來發出要求以啟動代理程式的配置選項。請使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，指令即會使用與此非預設協調佇列管理程式相關聯的一組內容檔。

如果未指定此參數，則會使用一組以預設協調佇列管理程式為基礎的配置選項。

agent_name

必要項目。您想要啟動的 Managed File Transfer 代理程式名稱。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，AGENT2 在前景中啟動及執行。

```
fteStartAgent -F AGENT2
```

在下列範例中（UNIX 及 Linux 系統），AGENT2 是隨著非預設協調佇列管理程式 QM_SATURN 一起啟動：

```
./fteStartAgent -p QM_SATURN AGENT2
```

您也可以如下所示指定 **fteStartAgent** 的路徑來執行此指令：

```
path/fteStartAgent agentname
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

回應

在某些情況下，您在執行 **fteStartAgent** 指令之後可能會看見錯誤訊息：

- 如果您執行 **fteStartAgent** 指令時，看到下列錯誤訊息，表示您的環境中可能有其他的程式庫路徑與 Managed File Transfer 衝突：

```
BFGCL0001E: An internal error has occurred. 異常狀況為: 'CC=2;RC=2495;AMQ8568:  
找不到原生 JNI 程式庫 'mqjbnd'。[3=mqjbnd]
```

當代理程式是以 Java 32 位元版本執行時（目前大部分平台都是如此），如果 LD_LIBRARY_PATH 或 LIBPATH 環境變數設為在 32 位元版本之前先參照 64 位元版本程式庫，則會發生此錯誤。

若要解決此問題，請設定 Managed File Transfer 代理程式內容 javaLibraryPath，以參照正確的程式庫位置。例如，若為 AIX 上的 mqjbnd，請設為：/usr/mqm/java/lib。若為 Linux 上的 mqjbnd，請設為：/opt/mqm/java/lib

相關工作

[第 129 頁的『以 Windows 服務方式啟動 MFT 代理程式』](#)

您可以 Windows 服務方式啟動代理程式，則當您登出 Windows 時，代理程式會繼續執行且可以接收檔案傳送。

[第 176 頁的『列出 MFT 代理程式』](#)

您可以使用指令行或 IBM MQ Explorer 來列出向特定佇列管理程式登錄的 Managed File Transfer 代理程式。

[第 177 頁的『停止 MFT 代理程式』](#)

您可以從指令行停止 Managed File Transfer 代理程式。停止代理程式時，代理程式開始靜止，但允許代理程式在停止之前完成現行檔案傳送。您也可以從指令行指定 **-i** 參數，以立即停止代理程式。代理程式停止之後，一直到重新啟動之前，您都無法使用該代理程式來傳送檔案。

相關參考

[第 132 頁的『在 z/OS 上啟動 MFT 代理程式』](#)

在 z/OS 上，除了從「UNIX 系統服務」階段作業中執行 **fteStartAgent** 指令之外，您還可以從 JCL 以啟動作業的方式來啟動代理程式，而不需要互動式階段作業。

fteStartLogger

fteStartLogger 指令會啟動 Managed File Transfer 記載應用程式。

用途

使用 **fteStartLogger** 指令啟動日誌程式。日誌程式可以是與協調佇列管理程式執行於相同系統上的檔案或資料庫應用程式。如需相關資訊，請參閱第 89 頁的『配置 MFT 日誌程式』主題。若為 IBM WebSphere MQ 7.5 或更新版本，日誌程式程序控制器將管理日誌程式的啟動。但日誌程式程序控制器可能會等待一段時間（例如當日誌程式的失敗率很高時），再嘗試重新啟動日誌程式。身為 IBM MQ 管理者，您可以使用 **fteStartLogger** 指令來置換這項等待，並起始啟動日誌程式。如果日誌程式程序控制器在等待佇列管理程式變成可用，這個指令也會起始日誌程式程序控制器以試圖重新連接佇列管理程式。

如果您已使用 **fteModifyLogger** 指令將日誌程式配置成以 Windows 服務方式執行，則執行 **fteStartLogger** 指令會啟動 Windows 服務。

如果日誌程式未啟動或已啟動，此指令會傳回錯誤。日誌程式會根據 `logger.properties` 檔中定義的值，與其佇列管理程式進行通訊。

只有在要使用一組不同於預設值的配置選項集時，才可為此指令指定 **-p** 參數。如需 `logger` 內容的相關資訊，請參閱第 99 頁的『MFT 日誌程式配置內容』。

語法

fteStartLogger

```
►► fteStartLogger -p 配置選項 -F logger_name ◄◄
```

參數

logger_name

必要項目。您想要啟動的 Managed File Transfer 日誌程式名稱。

-p configuration_options

選用項目。此參數可決定一組用來發出要求以啟動日誌程式的配置選項。請使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，**fteStartLogger** 會使用與此非預設協調佇列管理程式相關聯的一組內容檔。

如果未指定此參數，則會使用一組以預設協調佇列管理程式為基礎的配置選項。

-F

選用項目。以前景程序方式執行日誌程式（而非預設的背景程序）。如果您已使用 **fteModifyLogger** 指令，將日誌程式配置成以 Windows 服務方式執行，則 **-F** 參數會置換此配置。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，先前已建立一個名為 `logger1` 的日誌程式。此指令顯示如何啟動日誌程式以前景程序方式執行：

```
fteStartLogger -F logger1
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

相關概念

第 89 頁的『配置 MFT 日誌程式』

相關參考

第 415 頁的『fteModify 日誌程式 (以 Windows 服務方式執行 MFT 日誌程式)』

使用 **fteModifyLogger** 指令來修改 Managed File Transfer 日誌程式，以便它可以 Windows 服務方式執行。您只能在 Windows 平台上使用此指令，必須由 IBM MQ 管理者及 `mqm` 群組成員的使用者執行，且必須先使用 **fteStopLogger** 指令停止日誌程式。

第 450 頁的『fteStopLogger』

fteStopLogger 指令會停止 Managed File Transfer 日誌程式。

第 278 頁的『MFT 日誌程式錯誤處理及拒絕』

Managed File Transfer 日誌程式可識別兩種類型的錯誤：個別訊息錯誤及一般錯誤。

fteStopAgent

使用 **fteStopAgent** 指令，以受管制的方式停止 Managed File Transfer 代理程式，或在必要時使用 **-i** 參數立即停止代理程式。

用途

使用 **fteStopAgent** 指令停止代理程式時，可允許代理程式在停止之前完成現行檔案傳送，或者，即使代理程式目前正在傳送檔案，也立即停止代理程式。代理程式停止之後，一直到重新啟動代理程式之前，您都無法使用該代理程式來傳送檔案。

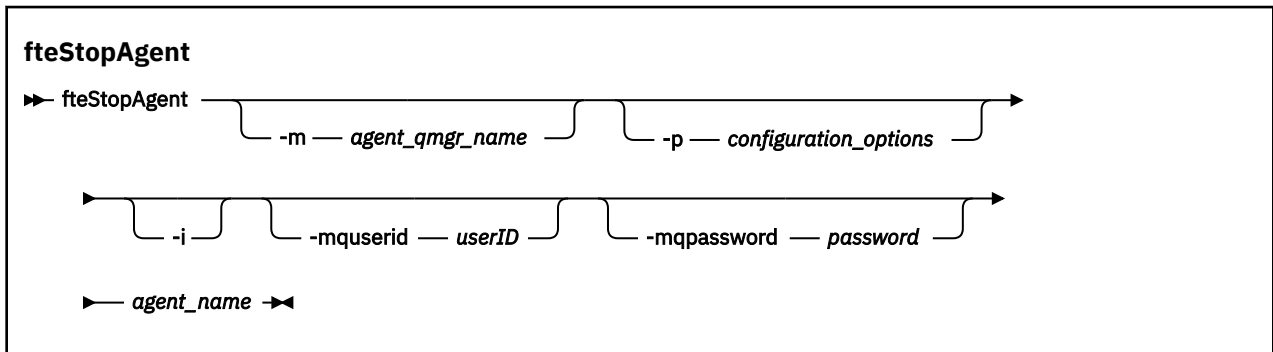
如果您要停止的代理程式已連接至 IBM MQ 網路，則可以從任何可連接至 IBM MQ 網路並遞送至代理程式佇列管理程式的系統執行 **fteStopAgent** 指令。具體而言，為了讓指令能夠執行，您必須在此系統上安裝並配置 Managed File Transfer 元件（「服務」或「代理程式」），才能與 IBM MQ 網路進行通訊。如果沒有可用的連線功能詳細資料，則會對本端系統上的預設佇列管理程式建立連結模式連線。如果 `command.properties` 不存在，則會產生錯誤。

如果您要停止的代理程式未連接至 IBM MQ 網路（例如，如果 IBM MQ 網路目前無法使用），您只能從執行代理程式的系統中執行 **fteStopAgent** 指令。為了停止未連接至 IBM MQ 網路的代理程式，您必須以執行代理程式的相同使用者身分來執行 **fteStopAgent** 指令。或者，如果代理程式是在 Windows 系統上執行，您也可以以管理者身分來執行此指令。

僅在要使用一組不同於您的預設集的配置選項時，才為此指令指定選用的 **-p** 參數。如需相關資訊，請參閱第 463 頁的『[MFT agent.properties 檔案](#)』。

如果您的代理程式以 Windows 服務方式執行，則執行 **fteStopAgent** 指令會停止 Windows 服務。如需相關資訊，請參閱第 129 頁的『[以 Windows 服務方式啟動 MFT 代理程式](#)』。

語法



參數

-m (agent_qmgr_name)

選用項目。您想要停止的代理程式所連接的佇列管理程式名稱。

如果代理程式位於遠端系統上或者代理程式位於本端系統上，但您不是啟動該代理程式的使用者，則必須使用 **-m** 參數並具有適當的權限。如需權限的相關資訊，請參閱第 296 頁的『[限制 MFT 特定資源的群組權限](#)』。

-p (configuration_options)

選用項目。此參數決定一組用來發出要求以停止代理程式的配置選項。請使用非預設協調佇列管理程式的名稱作為此參數的輸入。然後，指令即會使用與此非預設協調佇列管理程式相關聯的一組內容檔。

如果未指定此參數，則會使用一組以預設協調佇列管理程式為基礎的配置選項。

-i

選用項目。立即停止代理程式。代理程式不會完成目前進行的任何傳送。

如果未指定 **-i** 參數，代理程式會完成目前正在進行的任何傳送，但代理程式不會啟動任何新的傳送。

-mquserid (userID)

選用項目。指定用來向指令佇列管理程式進行鑑別的使用者 ID。

-mqpassword (password)

選用項目。指定用來向指令佇列管理程式進行鑑別的密碼。您還必須指定 **-mquserid** 參數。如果指定了 **-mquserid**，卻未指定 **-mqpassword**，則系統會提示您提供相關聯的密碼。將不會顯示密碼。

agent_name

必要項目。您想要停止的 Managed File Transfer 代理程式的名稱。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，停止佇列管理程式 QM_JUPITER 上的代理程式 AGENT2。因為 AGENT2 所連接的這個佇列管理程式與那一組配置選項所指定的佇列管理程式不相同，所以使用 **-m** 參數。

```
fteStopAgent -m QM_JUPITER AGENT2
```

回覆碼**0**

指令已順利完成。

1

指令未順利結束。

相關工作

[第 177 頁的『停止 MFT 代理程式』](#)

您可以從指令行停止 Managed File Transfer 代理程式。停止代理程式時，代理程式開始靜止，但允許代理程式在停止之前完成現行檔案傳送。您也可以在指令行指定 **-i** 參數，以立即停止代理程式。代理程式停止之後，一直到重新啟動之前，您都無法使用該代理程式來傳送檔案。

相關參考

[第 445 頁的『fteStartAgent』](#)

fteStartAgent 指令會從指令行啟動 Managed File Transfer 代理程式。

[第 177 頁的『在 z/OS 上停止 MFT 代理程式』](#)

如果您從 JCL 以已啟動作業的方式在 z/OS 上執行 Managed File Transfer Agent，則除了 **fteStopAgent** 指令之外，代理程式還接受 z/OS 操作員指令 **MODIFY** 及 **STOP**。

fteStopLogger

fteStopLogger 指令會停止 Managed File Transfer 日誌程式。

用途

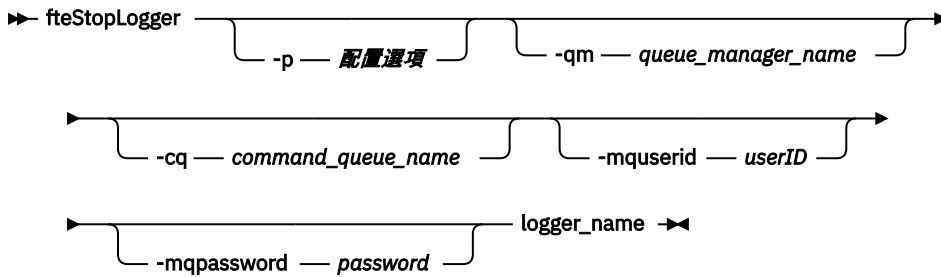
使用 **fteStopLogger** 指令停止日誌程式。日誌程式可以是將受管理檔案傳送活動歷程記錄至檔案的檔案日誌程式，也可以是將歷程記錄至資料庫的資料庫日誌程式。

停止獨立式日誌程式程序的其他注意事項

如果日誌程式是以 Windows 服務方式執行，則執行 **fteStopLogger** 指令會停止 Windows 服務。

語法

fteStopLogger



參數

-p (*configuration_options*)

選用項目。判斷一組用以停止日誌程式的配置選項。將配置選項集的名稱用作 **-p** 參數的值。依照慣例，此值是協調佇列管理程式的名稱。如果未指定此參數，則會使用一組預設的配置選項。

-qm (*queue_manager_name*)

選用項目。依預設，假設日誌程式的指令佇列位於 **-p** 參數（或其預設值）所指定的協調佇列管理程式上。如果您要將日誌程式指令傳送至其他位置的指令佇列，請使用 **-qm** 參數來指定替代目的地。無論如何，此指令會連接至 **-p** 參數指示的指令佇列管理程式，而不論訊息的最終目的地。

-cq (*command_queue_name*)

選用項目。指定傳送停止訊息至其中的指令佇列。在大部分情況下，日誌程式會使用預設佇列名稱，意味著此參數並非必要。

-mquserid (*userID*)

選用項目。指定用來向指令佇列管理程式進行鑑別的使用者 ID。

-mqpassword (*password*)

選用項目。指定用來向指令佇列管理程式進行鑑別的密碼。您還必須指定 **-mquserid** 參數。如果指定了 **-mquserid**，卻未指定 **-mqpassword**，則系統會提示您提供相關聯的密碼。將不會顯示密碼。

logger_name

必要項目。您想要停止的 Managed File Transfer 日誌程式名稱。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，先前已建立一個名為 logger1 的日誌程式，且目前正在執行中。此指令顯示如何停止日誌程式：

```
fteStopLogger logger1
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

相關概念

第 89 頁的『配置 MFT 日誌程式』

相關參考

第 415 頁的『fteModify 日誌程式 (以 Windows 服務方式執行 MFT 日誌程式)』

使用 **fteModifyLogger** 指令來修改 Managed File Transfer 日誌程式，以便它可以 Windows 服務方式執行。您只能在 Windows 平台上使用此指令，必須由 IBM MQ 管理者及 mqm 群組成員的使用者執行，且必須先使用 **fteStopLogger** 指令停止日誌程式。

第 447 頁的『fteStartLogger』

fteStartLogger 指令會啟動 Managed File Transfer 記載應用程式。

MFT 配置參照

在 MFT 內容中使用環境變數

從 IBM WebSphere MQ 7.5 開始，可能會在代表檔案或目錄位置的 Managed File Transfer 內容中使用環境變數。這可讓執行產品的某部分時所使用的檔案或目錄，隨著環境變更而改變其所在位置。例如，執行程序的使用者為何。

下列是接受檔案或目錄位置，因而可包含環境變數的內容：

- 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

範例

在此範例中，使用者 fteuser 在 Windows 系統上使用環境變數 USERPROFILE：

```
wmqfte.database.credentials.file=%USERPROFILE%\logger\mqmftcredentials.xml
```

解析為下列檔案路徑：

```
C:\Users\fteuser\logger\mqmftcredentials.xml
```

在此範例中，使用者 `fteuser` 在 UNIX 系統上使用環境變數 `HOME`：

```
transferRoot=$HOME/fte/
```

解析為下列檔案路徑：

```
/home/fteuser/fte/
```

MFT installation.properties 檔

`installation.properties` 檔案指定預設配置選項集的名稱。此項目將 Managed File Transfer 指向一組結構化目錄以及包含要使用之配置的內容檔。一組配置選項的名稱通常是關聯的協調佇列管理程式的名稱。

此檔案由安裝程式建立，可使用 **`fteChangeDefaultConfigurationOptions`** 指令進行變更。

`installation.properties` 檔案位於 `MQ_DATA_PATH` 目錄中。例如，在 Windows 上，預設檔案位置為 `MQ_DATA_PATH\mqft\installations\installation_name`，而在 UNIX 及 Linux 系統上，預設檔案位置為 `/var/mqm/mqft/installations/installation_name`。

V 9.0.1 對於 Redistributable Managed File Transfer Agent，將在您執行 **`fteCreateEnvironment`** 指令時設定資料路徑。如果您執行此指令並使用 **`-d`** 參數指定您選擇的位置，則會為此位置設定資料路徑。如果您沒有使用 **`fteCreateEnvironment`** 指令指定此位置，則會在 Redistributable Managed File Transfer Agent 進行解壓縮的根目錄下建立 `mftdata` 目錄。Redistributable Managed File Transfer Agent 的 `installation.properties` 檔位於 `MQ_DATA_PATH\mqft\installations\MFTZipInstall` 目錄中。

`installation.properties` 檔包含下列值：

表 39: 基本內容

內容名稱	說明	預設值
commandMessagePriority	<p>針對 fteStopAgent、fteCancelTransfer 及 ftePingAgent 指令，設定內部訊息及指令訊息的優先順序。</p> <p>如果提交大量的傳送要求以快速連續地傳送許多小型檔案，新的傳送要求可能會變成置於來源代理程式的指令佇列中。外部及內部訊息具有預設的 IBM MQ 訊息優先順序，而讓新的傳送要求封鎖內部訊息。如此可能會導致超出傳送交涉時間，而使得傳送進入回復狀態。</p> <p>您也可以使用 <code>commandMessagePriority</code> 內容，來設定內部確認及需要確認的訊息優先順序。</p> <p>若要將內部 Managed File Transfer 訊息的優先順序設定為高於新的傳送要求，請將此內容設定為 1（最低）到 9（最高）之間的值。</p> <p><code>commandMessage</code> 優先順序內容的預設值是 8。這表示如果代理程式指令佇列上的 IBM MQ 屬性 <code>DEFPRTY</code>（預設優先順序）小於或等於 7，則內部協議訊息優先於新的傳送要求。如果 <code>DEFPRTY</code> 屬性的值設定為 8 或 9，為了維持 <code>commandMessagePriority</code> 內容的有效性，您必須變更 <code>DEFPRTY</code> 或 <code>commandMessagePriority</code> 內容。</p>	<p>對於 IBM MQ 9.0.0.0，預設值為 8。</p> <p>對於較舊版次，且在 APAR IT06213 之前，預設值為 <code>MQPRI_PRIORITY_AS_Q_DEF</code> 常數，其值為 -1。</p>
defaultProperties	<p>預設配置選項組的名稱。此值是位於配置目錄中的目錄名稱，該目錄包含子目錄及指定配置資訊的內容檔。</p>	無預設值
enableFunctionalFixPack	<p>要啟用的修正套件功能層次。依預設，不會啟用修正套件隨附的任何新功能。將此內容設為版本 ID，以啟用該版本隨附的新特性。</p> <p>您可以指定含或不含句點字元 (.) 的版本 ID。例如，若要使用 IBM MQ 8.0.0 Fix Pack 2 提供的函數，請將此內容設為 8002 或 8.0.0.2。</p>	無預設值

表 39: 基本內容 (繼續)

內容名稱	說明	預設值
<p>     messagePublicationFormat </p>	<p>可讓您指定 MFT 代理程式用於其狀態 XML 訊息的訊息發佈格式。此內容可以設定為下列值：</p> <p>messagePublicationFormat=mixed</p> <p>除了在 /LOG 主題樹狀結構下以 MQFMT_STRING 的 MQMD 格式發佈的訊息外，將在不使用 MQMD FORMAT (MQFMT_NONE) 的情況下發佈訊息。</p> <p>messagePublicationFormat=MQFMT_NONE</p> <p>在不使用 MQMD FORMAT 的情況下發佈訊息。</p> <p>messagePublicationFormat=MQFMT_STRING</p> <p>以字串格式發佈訊息。</p> <p> IBM MQ 9.0.2 中存在撰寫程式錯誤。在該版本中，當您指定 messagePublicationFormat=mixed 時，效果與指定 messagePublicationFormat=MQFMT_NONE 相同。</p> <p> 此錯誤在 IBM MQ 9.0.3 中已修正。</p> <p>在 IBM MQ 8.0 之前，MFT 代理程式已將 XML 狀態訊息發佈至 SYSTEM.FTE 主題 (MQFMT_STRING)。如有可能，必須更新先前使用 IBM WebSphere MQ 7.5 的應用程式，以處理 IBM MQ 8.0 或更新版本格式的訊息。如果無法變更應用程式，請將 messagePublicationFormat 內容設定為 MQFMT_STRING，以回復至 IBM WebSphere MQ 7.5 行為。</p>	<p>messagePublicationFormat=mixed</p>
<p>   z/OS 專用： </p>		

表 39: 基本內容 (繼續)

內容名稱	說明	預設值
  productId	要記錄其 MFT 用法的產品類型： <ul style="list-style-type: none"> • 獨立式 Managed File Transfer 產品。（MFT 是 productId）。 • IBM MQ Advanced 產品的一部分。（ADVANCED 是 productId）。 • IBM MQ Advanced for z/OS, Value Unit Edition 產品的一部分。（ADVANCEDVUE 是 productId）。 如需產品使用情形記錄的相關資訊，請參閱 報告產品資訊 。  在 Multiplatforms 上，系統會忽略此內容。	MFT

下列文字是 `installation.properties` 檔案內容範例。

```
defaultProperties=ERIS
```

ERIS 是與 `installation.properties` 檔位於相同目錄中的目錄名稱。ERIS 目錄包含子目錄及可說明一組配置選項的內容檔。

相關概念

第 49 頁的『[多平台上的 MFT 配置選項](#)』

Managed File Transfer 提供了一組內容檔，這些檔案包含有關您的設定的金鑰資訊，是作業所必要的。這些內容檔位於安裝產品時所定義的配置目錄中。

相關參考

第 321 頁的『[fteChangeDefaultConfigurationOptions](#)』

使用 **`fteChangeDefaultConfigurationOptions`** 指令，可變更您希望 Managed File Transfer 使用的預設配置選項。這些配置選項的值定義 Managed File Transfer 使用的內容檔群組。

MFT coordination.properties 檔

`coordination.properties` 檔案指定協調佇列管理程式的連線詳細資料。因為數個 Managed File Transfer 安裝可能共用相同的協調佇列管理程式，所以您可以使用共用磁碟機上一般 `coordination.properties` 檔案的符號鏈結。

`coordination.properties` 檔由安裝程式或 **`fteSetupCoordination`** 指令建立。您可以搭配使用 **`fteSetupCoordination`** 指令與 **`-f`** 旗標，以變更此檔案中的基本協調佇列管理程式內容。若要變更或新增進階協調佇列管理程式內容，您必須在文字編輯器中編輯檔案。

`coordination.properties` 檔案位於 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` 目錄中。

`coordination.properties` 檔包含下列值：

表 40: 協調佇列管理程式內容		
內容名稱	說明	預設值
<code>coordinationQMgr</code>	協調佇列管理程式的名稱。	無預設值

表 40: 協調佇列管理程式內容 (繼續)

內容名稱	說明	預設值
coordinationQMGrHost	協調佇列管理程式的主機名稱或 IP 位址。	無預設值
coordinationQMGrPort	用於與協調佇列管理程式進行用戶端連線的埠號。	1414
coordinationQMGrChannel	用來連接至協調佇列管理程式的 SVRCONN 通道名稱。	SYSTEM.DEF.SVRCONN

如果未指定 coordinationQMGrHost 內容的值，則預設會使用連結模式。

如果指定了 coordinationQMGrHost 內容的值，但未指定 coordinationQMGrPort 及 coordinationQMGrChannel 內容的值，則預設會使用 1414 埠號及 SYSTEM.DEF.SVRCONN 通道。

表 41: 進階協調佇列管理程式內容

內容名稱	說明	預設值
代理程式內容:		
agentStatusJitterTolerance	代理程式狀態訊息被視為逾期之前可延遲發佈的最長時間。此值的測量單位為毫秒。 狀態訊息的經歷時間以它在協調佇列管理程式上發佈的時間為基準。不過，代理程式會在協調佇列管理程式接收訊息之前先發出訊息，以使它有足夠的時間在 IBM MQ 網路間傳遞。如果此傳輸一律耗用相同的時間量，則不論實際的傳輸時間為何，相隔 60 秒建立的訊息都會以 60 秒的間隔發佈。但若訊息之間的傳輸時間不同，它們可能相隔 60 秒建立，但發佈時間隔可能為 61、59、58 及 62 秒。60 的最大偏差（此範例中為 2 秒）是不穩定。此內容決定訊息由於不穩定而被視為逾期之前的延遲上限。	3000
字碼頁內容:		
coordinationCcsid	指令用來連接至協調佇列管理程式的字碼頁。代理程式也是使用此字碼頁對協調佇列管理程式執行各種發佈。如果指定 coordinationCcsid 的值，您也必須指定 coordinationCcsidName 的值。	1208
coordinationCcsidName	coordinationCcsid 的 Java 表示法。如果指定 coordinationCcsidName 的值，您也必須指定 coordinationCcsid 的值。	UTF8

表 41: 進階協調佇列管理程式內容 (繼續)		
內容名稱	說明	預設值
連線內容:		
javaLibraryPath	在連結模式下連接至佇列管理程式時，Managed File Transfer 必須能夠存取 IBM MQ Java 連結程式庫。依預設，Managed File Transfer 會在 IBM MQ 所定義的預設位置中尋找連結程式庫。如果連結程式庫位於不同位置，請使用此內容來指定連結程式庫的位置。	MQ_INSTALLATION_PATH/ java/lib
多重實例佇列管理程式內容:		
coordinationQMGrStandby	針對 coordinationQMGr 內容所定義的多重實例協調佇列管理程式的待用實例，用於用戶端連線的主機名稱及埠號（採用 IBM MQ CONNAME 格式）。例如， <i>host_name(port_number)</i>	無預設值
佇列內容:		
dynamicQueuePrefix	此內容定義用於產生暫時佇列名稱的 IBM MQ 字首。 dynamicQueuePrefix 內容的格式遵循 IBM MQ MQOD 結構的 DynamicQName 欄位格式。如需相關資訊，請參閱 建立動態佇列 。 如果您想要對需要代理程式回應的指令所產生的暫時回覆佇列使用特定的 IBM MQ 字首，則也可以在 <code>command.properties</code> 檔案中定義此內容。	WMQFTE.*
modelQueueName	此內容定義用於產生暫時佇列的 IBM MQ 模型佇列。 如果您要對需要代理程式回應的指令所產生的暫時回覆佇列使用特定的 IBM MQ 模型佇列，您也可以在此 <code>command.properties</code> 檔案中定義此內容。如需相關資訊，請參閱第 460 頁的『 MFT command.properties 檔 』。	SYSTEM.DEFAULT.MODEL.QUEUE
安全內容:		
userIdForClientConnect	透過用戶端連線傳送至 IBM MQ 的使用者 ID。如果已指定 <i>java</i> ，則會隨著 IBM MQ 連線要求而傳送 JVM 所報告的使用者名稱。此內容值可以是無或 <i>java</i> 。	無

表 41: 進階協調佇列管理程式內容 (繼續)		
內容名稱	說明	預設值
coordinationQMgrAuthenticationCredentialsFile	包含用於協調佇列管理程式連線之 MQ 連線認證的檔案路徑。	<p>Z/OS 如需建立「鑑別認證檔」的詳細資料，請參閱 建立 MFT 認證檔</p> <p>ULW 如需此檔案的位置及許可權的相關資訊，請參閱 配置 MQMFTCredentials.xml。</p> <p>ULW 關於建立「鑑別認證檔」的進一步詳細資料位於 MFT 和 IBM MQ 連線鑑別 中</p>
訂閱內容:		
coordinationSubscriptionTopic	<p>請利用這個內容來指定 SYSTEM.FTE，以取得 IBM MQ 網路狀態的相關發佈。所有工具仍會發佈至 SYSTEM.FTE 主題，但您可以變更 IBM MQ 拓撲，以根據其內容將這些出版品配送至不同的主題。然後，您可以使用此功能強制工具訂閱其中的另一個主題。</p> <p>對於 IBM WebSphere MQ 7.5 以及更新版本的修正套件，您需要 APAR IC96850 的臨時修正程式，才能讓「IBM MQ Explorer」及 fteListMonitors 指令能夠辨識內容。</p>	SYSTEM.FTE

下列文字是 coordination.properties 檔案內容範例。

```
coordinationQMgr=ERIS
coordinationQMgrHost=kuiper.example.com
coordinationQMgrPort=2005
coordinationQMgrChannel=SYSTEM.DEF.SVRCONN
```

ERIS 是位於系統 kuiper.example.com 上的 IBM MQ 佇列管理程式名稱。佇列管理程式 ERIS 是接收 Managed File Transfer 所傳送日誌資訊的佇列管理程式。

相關概念

第 49 頁的『[多平台上的 MFT 配置選項](#)』

Managed File Transfer 提供了一組內容檔，這些檔案包含有關您的設定的金鑰資訊，是作業所必要的。這些內容檔位於安裝產品時所定義的配置目錄中。

相關參考

第 433 頁的『[fteSetupCoordination](#)』

fteSetupCoordination 指令會建立 Managed File Transfer 的內容檔及協調佇列管理程式目錄。

第 508 頁的『[MFT 的 SSL 內容](#)』

對 IBM MQ 及 Managed File Transfer 使用 SSL 或 TLS，可避免代理程式與佇列管理程式之間未獲授權的連線，以及加密代理程式與佇列管理程式之間的訊息資料流量。

MFT command.properties 檔

command.properties 檔指定當您發出指令時要連接的指令佇列管理程式，以及 Managed File Transfer 聯絡該佇列管理程式所需的資訊。

command.properties 檔由安裝程式或 **fteSetupCommands** 指令建立。您可以搭配使用 **fteSetupCommands** 指令與 **-f** 旗標，以變更此檔案的基本指令佇列管理程式內容。若要變更或新增進階指令佇列管理程式內容，您必須在文字編輯器中編輯檔案。

部分 Managed File Transfer 指令會連接至代理程式佇列管理程式或協調佇列管理程式，而不是指令佇列管理程式。如需哪些指令連接至哪些佇列管理程式的相關資訊，請參閱第 307 頁的『哪些 MFT 指令和程序連接至哪個佇列管理程式』。

command.properties 檔案位於 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` 目錄中。

command.properties 檔包含下列值：

內容名稱	說明	預設值
connectionQMgr	用來連接至 IBM MQ 網路的佇列管理程式的名稱。	無預設值
connectionQMgrHost	連線佇列管理程式的主機名稱或 IP 位址。	無預設值
connectionQMgrPort	在用戶端模式中用來連接至連線佇列管理程式的埠號。	1414
connectionQMgrChannel	用來連接至連線佇列管理程式的 SVRCONN 通道名稱。	SYSTEM.DEF.SVRCONN





如果未指定 connectionQMgrHost 內容的值，則預設會使用連結模式。

如果指定了 connectionQMgrHost 內容的值，但未指定 connectionQMgrPort 及 connectionQMgrChannel 內容的值，則預設會使用 1414 埠號及 SYSTEM.DEF.SVRCONN 通道。

內容名稱	說明	預設值
字碼頁內容：		
connectionCcsid	指令用來連接至指令佇列管理程式的字碼頁。如果指定 connectionCcsid 的值，您也必須指定 connectionCcsidName 的值。	1208
connectionCcsidName	connectionCcsid 的 Java 表示法。如果指定 connectionCcsidName 的值，您也必須指定 connectionCcsid 的值。	UTF8
多重實例佇列管理程式內容：		
connectionQMgrStandby	針對 connectionQMgr 內容所定義的多重實例指令佇列管理程式的待命實例，用於用戶端連線的主機名稱及埠號，採用 IBM MQ CONNAME 格式。例如， <code>host_name(port_number)</code>	無預設值
安全內容：		

表 43: 進階指令佇列管理程式內容 (繼續)

內容名稱	說明	預設值
userIdForClientConnect	透過用戶端連線傳送至 IBM MQ 的使用者 ID。如果已指定 <i>java</i> , 則會隨著 IBM MQ 連線要求而傳送 JVM 所報告的使用者名稱。此內容值可以是無或 <i>java</i> 。	無
connectionQMGrAuthenticationCredentials 檔案	包含用於指令佇列管理程式連線之 MQ 連線認證的檔案路徑。	<p>z/OS 如需建立「鑑別認證檔」的詳細資料, 請參閱 建立 MFT 認證檔</p> <p>ULW 如需此檔案的位置及許可權的相關資訊, 請參閱 配置 MQMFTCredentials.xml。</p> <p>ULW 關於建立「鑑別認證檔」的進一步詳細資料位於 MFT 和 IBM MQ 連線鑑別 中</p>
佇列內容:		
dynamicQueuePrefix	<p>對於需要代理程式回應的指令, 此內容定義用於產生暫時回覆佇列名稱的 IBM MQ 字首。</p> <p>dynamicQueuePrefix 內容的格式遵循 IBM MQ MQOD 結構的 DynamicQName 欄位格式。如需相關資訊, 請參閱 建立動態佇列。</p> <p>如果您想要對 WMQFTE 所產生的暫時佇列使用特定的 IBM MQ 字首, 也可以在 <code>coordination.properties</code> 檔中定義此內容。</p>	WMQFTE.*
modelQueueName	<p>對於需要代理程式回應的指令, 此內容定義用於產生暫時回覆佇列的 IBM MQ 模型佇列。</p> <p>如果您想要對 WMQFTE 所產生的暫時佇列使用特定的 IBM MQ 模型佇列, 您也可以在此 <code>coordination.properties</code> 檔中定義此內容。如需相關資訊, 請參閱 第 456 頁的『MFT coordination.properties 檔』。</p>	SYSTEM.DEFAULT.MODEL.QUEUE
連線內容:		

表 43: 進階指令佇列管理程式內容 (繼續)		
內容名稱	說明	預設值
javaLibraryPath	在連結模式下連接至佇列管理程式時，Managed File Transfer 必須能夠存取 IBM MQ Java 連結程式庫。依預設，Managed File Transfer 會在 IBM MQ 所定義的預設位置中尋找連結程式庫。如果連結程式庫位於不同位置，請使用此內容來指定連結程式庫的位置。	/opt/mqm/java/lib
  legacyXMLMessageMQMDFormat	Managed File Transfer 指令 XML 訊息現在會傳送至具有空白 MQMD 格式欄位的佇列。舊版的產品將 MQMD 格式欄位設為 MQSTR（文字訊息字串）。將此內容設為 true，可將 Managed File Transfer 指令 XML 訊息傳送至 MQMD 格式欄位為 MQSTR 的佇列。 如果 MQMD 格式欄位設為 MQSTR，而 MQ 網路中有通道啟用了資料轉換，則 Managed File Transfer 指令 XML 訊息有可能已毀損。	false
代理程式內容		
    failCleanAgentWithNoArguments	依預設，此內容的值為 true，這表示如果僅指定代理程式名稱參數，則 fteCleanAgent 指令無法執行。如果將此內容設為 false，則表示如果僅設定代理程式名稱參數，則 fteCleanAgent 指令的行為等價於指定 -all 參數。	true

下列文字是 `command.properties` 檔案內容範例。

```
connectionQMGr=PLUTO
connectionQMGrHost=kuiper.example.com
connectionQMGrPort=1930
connectionQMGrChannel=SYSTEM.DEF.SVRCONN
```

PLUTO 是位於系統 `kuiper.example.com` 上的 IBM MQ 佇列管理程式名稱。佇列管理程式 PLUTO 是 Managed File Transfer 指令所連接的佇列管理程式。

相關概念

第 49 頁的『[多平台上的 MFT 配置選項](#)』

Managed File Transfer 提供了一組內容檔，這些檔案包含有關您的設定的金鑰資訊，是作業所必要的。這些內容檔位於安裝產品時所定義的配置目錄中。

相關參考

第 508 頁的『[MFT 的 Java 系統內容](#)』

有一些 Managed File Transfer 指令及代理程式內容必須定義為 Java 系統內容，因為它們為無法使用指令或代理程式內容機制的早期功能定義配置。

第 508 頁的『[MFT 的 SSL 內容](#)』

對 IBM MQ 及 Managed File Transfer 使用 SSL 或 TLS，可避免代理程式與佇列管理程式之間未獲授權的連線，以及加密代理程式與佇列管理程式之間的訊息資料流量。

第 432 頁的『[fteSetupCommands: 建立 MFT command.properties 檔](#)』

fteSetupCommands 指令會建立 Managed File Transfer `command.properties` 檔案。此內容檔針對您發出指令時連接至 IBM MQ 網路的佇列管理程式，指定其詳細資料。

第 322 頁的『[fteCleanAgent: 清除 MFT 代理程式](#)』


使用 **fteCleanAgent** 指令，可透過從代理程式所使用的持續性及非持續性佇列中刪除訊息，來清除 Managed File Transfer Agent 使用的佇列。如果因資訊保留在代理程式使用的佇列中，而可能導致您在啟動代理程式時遇到問題，請使用 **fteCleanAgent** 指令。

MFT agent.properties 檔案

每一個 Managed File Transfer Agent 都有自己的內容檔 `agent.properties`，其中必須包含代理程式用來連接至其佇列管理程式的資訊。`agent.properties` 檔案也可以包含變更代理程式行為的內容。

`agent.properties` 檔案由安裝程式或 **fteCreateAgent**、**fteCreateBridgeAgent** 或 **fteCreateCDAgent** 指令建立。您可以搭配使用這些指令與 **-f** 旗標，以變更基本的代理程式佇列管理程式內容，以及與所建立的代理程式類型相關聯的進階代理程式內容。若要變更或新增進階代理程式內容，您必須在文字編輯器中編輯檔案。

代理程式的 `agent.properties` 檔案位於 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` 目錄中。

 在 z/OS 上，`agent.properties` 檔案位置是 `$BFG_CONFIG variable/mqft/config/coordination_qmgr_name/agents/agent_name`。

如果變更 `agent.properties` 檔，您必須重新啟動代理程式，以讓變更生效。

若為 IBM WebSphere MQ 7.5 或更新版本，您可以在代表檔案或目錄位置的一些 Managed File Transfer 內容中使用環境變數。這可讓您在執行產品的某部分時，隨著環境變更（例如執行程序的使用者為何）而改變檔案或目錄的所在位置。如需相關資訊，請參閱第 452 頁的『[在 MFT 內容中使用環境變數](#)』。

基本代理程式內容

每一個 `agent.properties` 檔案包含下列基本內容。

內容名稱	說明	預設值
agentName	代理程式的名稱。代理程式的名稱必須符合 IBM MQ 物件命名慣例。如需相關資訊，請參閱第 524 頁的『 MFT 物件命名慣例 』。	無預設值
agentDesc	代理程式的說明 - 如果選擇建立說明。	無預設值
agentQMgr	代理程式佇列管理程式名稱。	無預設值
agentQMgrHost	代理程式佇列管理程式的主機名稱或 IP 位址。	無預設值
agentQMgrPort	用戶端連接代理程式佇列管理程式所使用的埠號。	1414
agentQMgrChannel	用於連接至代理程式佇列管理程式的 SVRCONN 通道名稱。	SYSTEM.DEF.SVRCONN
agentType	代理程式的類型： <ul style="list-style-type: none"> 標準非橋接器代理程式 (STANDARD) 通訊協定橋接器代理程式 (BRIDGE) Connect:Direct 橋接器代理程式 (CD_BRIDGE) IBM Integration Bus 所使用的內嵌式代理程式 (EMBEDDED) Sterling File Gateway 內嵌式代理程式 (SFG) 	標準

如果未指定 `agentQMgrHost` 內容的值，則預設會使用連結模式。

如果指定了 agentQMGrHost 內容的值，但未指定 agentQMGrPort 及 agentQMGrChannel 內容的值，則預設會使用 1414 埠號及 SYSTEM.DEF.SVRCONN 通道。

進階代理程式內容

Managed File Transfer 也提供其他進階代理程式內容，以協助您配置代理程式。如果您想要使用下列任何內容，請手動編輯 agent.properties 檔，以新增所需的進階內容。當您在 Windows 上指定檔案路徑時，請確保將分隔字元反斜線 (\) 輸入為雙反斜線 (\\)，即跳出反斜線 (\)。或者，您可以使用單一正斜線 (/) 字元作為分隔字元。如需 Java 內容檔中字元跳出的相關資訊，請參閱 Oracle 文件 [Javadoc](#) 中的 Properties 類別。




- [代理程式大小內容](#)
- [字碼頁內容](#)
- [指令內容](#)
- [連線內容](#)
- [Connect:Direct 橋接器內容](#)
- [檔案至訊息及訊息至檔案代理程式內容](#)
- [一般代理程式內容](#)
- [輸入/輸出內容](#)
- [多通道支援內容](#)
- [多重實例內容](#)
- [程序控制器內容](#)
- [通訊協定橋接器內容](#)
-  [通訊協定橋接器代理程式日誌內容](#)
- [佇列內容](#)
- [資源監視內容](#)
- [根目錄內容](#)
- [排程器內容](#)
- [安全內容](#)
- [逾時內容](#)
-  [傳送回復逾時內容](#)
- [追蹤及記載內容](#)
- [傳送限制內容](#)
- [使用者結束程式內容](#)
- [IBM MQ 用戶端壓縮內容](#)
-  [z/OS 專用內容](#)
- [其他內容](#)

表 45: 進階代理程式內容		
內容名稱	說明	預設值
代理程式大小內容:		
agentCheckpointInterval	設定檢查點以進行回復的資料完整框架間隔。此為進階內容，且對大多數的 Managed File Transfer 配置而言，並不需要修改其值。 若有導致傳送進入回復狀態的問題，則傳送僅能回復至某個檢查點。因此，若此值愈大（包含 agentChunkSize、agentWindowSize 及 agentFrameSize 的值愈大），代理程式回復傳送所需的時間就愈長。若為穩定的 Managed File Transfer 網路，傳送很少會進入回復狀態，加大此值可能有利於增進整體效能。	1

表 45: 進階代理程式內容 (繼續)		
內容名稱	說明	預設值
agentChunkSize	檔案資料傳輸的每一個傳送區塊的大小。因而可代表來源與目的地代理程式之間傳送的 IBM MQ 訊息的大小上限。此為進階內容，且對大多數的 Managed File Transfer 配置而言，並不需要修改其值。 此值由來源代理程式與目的地代理程式協議達成，採用兩者之中較大的值。如果您想要變更此內容的值，請同時在來源代理程式及目的地代理程式上變更值。 agentChunkSize 是整數值。例如：agentChunkSize=10240 將區塊大小設為 10 KB。	262144 位元組 (等於 256KB)
agentFrameSize	傳送框架的視窗數目。此為進階內容，且對大多數的 Managed File Transfer 配置而言，並不需要修改其值。 就具有高延遲的網路而言，加大此值會使代理程式有較多的訊息區塊同時在作用中，因此可增進整體效能。 此內容值乘以 agentWindowSize 再乘以 agentChunkSize，即為每次傳輸代理程式所能使用的記憶體上限。例如，262144 位元組區塊 x 10 x 5 = 每個傳送作業有 12.5MB。 附註：若在單一傳送作業中傳送的檔案大小小於 12.5MB，則加大此內容將不會影響傳送的效能。	5
agentWindowSize	每一個視窗的區塊數目。此為進階內容，且對大多數的 Managed File Transfer 配置而言，並不需要修改其值。 就具有高延遲的網路而言，加大此值可增進整體效能。因為如此會使得代理程式具有較多的訊息區塊同時在作用中，並可減少確認通知訊息傳回來源代理程式的頻率。 此內容的值乘上 agentFrameSize 和 agentChunkSize，代表代理程式為每個傳送作業耗用的記憶體上限，也代表目的地代理程式之指令佇列上 IBM MQ 訊息資料的上限。例如，262144 位元組區塊 x 10 x 5 = 每個傳送作業的上限為 12.5MB。 附註：若在單一傳送作業中傳送的檔案大小小於 12.5MB，則加大此內容的值將不會影響傳送的效能。	10
字碼頁內容：		
agentCcsid	代理程式用來連接至其代理程式佇列管理程式的字碼頁。如果您指定了 agentCcsid 值，則必須同時指定 agentCcsidName 值。如需如何檢視 JVM 已知字碼頁的相關資訊，請參閱 fteCreateBridgeAgent 指令中的 -hsc 參數。	1208
agentCcsidName	agentCcsid 的 Java 表示法。若您指定了 agentCcsidName 的值，就必須也指定 agentCcsid 的值。	UTF8
指令內容：		
maxCommandHandlerThreads	控制傳送指令訊息的起始剖析及處理可用的執行緒數目。執行緒在作用中時，需要連接至佇列管理程式，但執行緒閒置時會釋放連線。	5
maxCommandOutput	為指令輸出而儲存的位元組數上限。此內容適用於針對受管理呼叫指定的指令，以及受管理傳送的 preSource、postSource、preDestination 及 postDestination 指令。其限制在 SYSTEM.FTE 主題上寫入傳送日誌的指令輸出長度。	10240
maxCommandRetries	代理程式允許的指令重試次數上限。此內容適用於針對受管理呼叫指定的指令，以及受管理傳送的 preSource、postSource、preDestination 及 postDestination 指令。	9
maxCommandWait	代理程式允許的重試間隔等待上限 (以秒為單位)。此內容適用於針對受管理呼叫指定的指令，以及受管理傳送的 preSource、postSource、preDestination 及 postDestination 指令。	60
immediateShutdownTimeout	若要立即關閉代理程式，您可以使用此內容來指定強制關閉之前代理程式等待傳送完成的最長時間量 (以秒為單位)。 註：請不要將此內容的值變更為小於預設值 10 秒。立即關閉代理程式需要足夠時間來結束任何外部程序。如果此內容的值太小，可能會遺留仍在執行中的程序。 如果此內容值指定為 0，代理程式會等待所有未完成的傳送停止。如果指定給這個內容的值無效，則會使用預設值。	10
連線內容：		
javaLibraryPath	在連結模式下連接至佇列管理程式時，Managed File Transfer 必須能夠存取 IBM MQ Java 連結程式庫。依預設，Managed File Transfer 會在 IBM MQ 所定義的預設位置中尋找連結程式庫。如果連結程式庫位於不同位置，請使用此內容指定連結程式庫的位置。	無
Connect:Direct 橋接器內容：		

表 45: 進階代理程式內容 (繼續)		
內容名稱	說明	預設值
cdNode	如果您想要使用 Connect:Direct 橋接器，此為必要內容。 Connect:Direct 節點名稱，用來將訊息從 Connect:Direct 橋接器代理程式傳送至目的地 Connect:Direct 節點。此節點屬於 Connect:Direct 橋接器，而非傳送來源或目的地的遠端節點。如需相關資訊，請參閱第 192 頁的『Connect:Direct 橋接器』。	無預設值
cdNodeHost	Connect:Direct 節點的主機名稱或 IP 位址，用來將檔案從 Connect:Direct 橋接器代理程式傳送至目的地節點 (Connect:Direct 橋接器節點)。 在大部分情況下，Connect:Direct 橋接器節點會與 Connect:Direct 橋接器代理程式位於相同的系統上。在這些情況下，此內容的預設值 (即本端系統的 IP 位址) 是正確的。如果您的系統有多個 IP 位址，或 Connect:Direct 橋接器節點與 Connect:Direct 橋接器代理程式位於不同系統，且其系統共用一個檔案系統，請使用此內容來指定 Connect:Direct 橋接器節點的正確主機名稱。 如果未設定 cdNode 內容，則會忽略此內容。	本端系統的主機名稱或 IP 位址
cdNodePort	Connect:Direct 橋接器節點的埠號，供用戶端應用程式用來與節點進行通訊。在 Connect:Direct 產品文件中，此埠也稱為 API 埠。 如果未設定 cdNode 內容，則會忽略此內容。	1363
cdTmpDir	將檔案傳送至目的地 Connect:Direct 節點之前，在 Connect:Direct 橋接器代理程式執行所在系統上暫時儲存檔案的位置。 此內容指定檔案暫存目錄的完整路徑。比方說，例如 cdTmpDir 設為 /tmp，則檔案暫時放置在 /tmp 目錄。 如果未設定 cdNode 內容，則會忽略此內容。 此內容的值可以包含環境變數。 如需相關資訊，請參閱第 452 頁的『在 MFT 內容中使用環境變數』。	<code>value_of_java.io.tmpdir</code> <code>/cdbridge-agentName</code> 在 Windows 上， <code>value_of_java.io.tmpdir</code> <code>\cdbridge-agentName</code>
cdTrace	代理程式是否會追蹤在 Connect:Direct 橋接器代理程式與其 Connect:Direct 節點之間傳送的資料。此內容值可以是 true 或 false。	false
cdMaxConnectionRetries	尚未成功連線的檔案傳送至傳送失敗之前的 Connect:Direct 連線嘗試次數上限。	-1 (嘗試次數無限)
cdMaxPartialWorkConnectionRetries	針對先前的成功連線嘗試且已完成傳送工作的檔案傳送，在判定傳送失敗之前可試圖進行 Connect:Direct 連線的次數上限。	-1 (嘗試次數無限)
cdMaxWaitForProcessEndStats	從程序結束之後到判定檔案傳送失敗之前，等待 Connect:Direct 程序完成資訊在 Connect:Direct 節點統計資訊內變成可用的時間上限 (以毫秒為單位)。資訊通常立即可用，但在某些失敗狀況下，不會發佈資訊。在這些狀況下，於等待此內容指定的時間量之後，檔案傳送作業即告失敗。	60000
cdAppName	Connect:Direct 橋接器代理程式用來連接至屬於橋接器的 Connect:Direct 節點的應用程式名稱。	Managed File Transfer <i>current version</i> ，其中 <i>current version</i> 是產品的版本號碼。
cdNodeLocalPortRange	在 Connect:Direct 橋接器代理程式與屬於橋接器的 Connect:Direct 節點之間，用於 Socket 連線的本端埠範圍。此值的格式為以逗點區隔的值清單或範圍清單。 依預設，作業系統會選取本端埠號。	無
cdNodeProtocol	Connect:Direct 橋接器代理程式用來連接至屬於橋接器的 Connect:Direct 節點的通訊協定。下列為有效值： • TCPIP • SSL • TLS	TCPIP
cdNodeKeystore	在 Connect:Direct 橋接器代理程式與屬於橋接器的 Connect:Direct 節點之間，用於安全通訊的金鑰儲存庫路徑。 如果未將 cdNodeProtocol 內容設為 SSL 或 TLS，則會忽略此內容。 此內容的值可以包含環境變數。 如需相關資訊，請參閱第 452 頁的『在 MFT 內容中使用環境變數』。	無
cdNodeKeystoreType	cdNodeKeystore 內容所指定的金鑰儲存庫之檔案格式。下列為有效值：jks 及 pkcs12。 如果未將 cdNodeProtocol 內容設為 SSL 或 TLS，則會忽略此內容。	jks

表 45: 進階代理程式內容 (繼續)






內容名稱	說明	預設值
cdNodeKeystoreCredentialsFile	包含 cdNodeKeystore 認證的檔案路徑。 此內容的值可以包含環境變數。 如需相關資訊，請參閱 第 452 頁的『在 MFT 內容中使用環境變數』。	 如需建立「鑑別認證檔」的詳細資料，請參閱 建立 MFT 認證檔  如需此檔案的位置及許可權的相關資訊，請參閱 配置 MQMFTCredentials.xml 。  關於建立「鑑別認證檔」的進一步詳細資料位於 MFT 和 IBM MQ 連線鑑別 中
cdNodeTruststore	在 Connect:Direct 橋接器代理程式與屬於橋接器的 Connect:Direct 節點之間，用於安全通訊的信任儲存庫路徑。 如果未將 cdNodeProtocol 內容設為 SSL 或 TLS，則會忽略此內容。 此內容的值可以包含環境變數。 如需相關資訊，請參閱 第 452 頁的『在 MFT 內容中使用環境變數』。	無
cdNodeTruststoreType	cdNodeTruststore 內容所指定的信任儲存庫檔案格式。下列為有效值: jks 及 pkcs12。 如果未將 cdNodeProtocol 內容設為 SSL 或 TLS，則會忽略此內容。	jks
cdNodeTruststoreCredentialsFile	包含 cdNodeTruststore 認證的檔案路徑。 此內容的值可以包含環境變數。 如需相關資訊，請參閱 第 452 頁的『在 MFT 內容中使用環境變數』。	請參閱 建立 MFT 認證檔 及 配置 MQMFTCredentials.xml  建立 MFT 認證檔中也有建立「鑑別認證檔」的進一步詳細資料。  關於建立「鑑別認證檔」的進一步詳細資料位於 MFT 和 IBM MQ 連線鑑別 中
logCDProcess	在 output0.log 檔的代理程式事件日誌中記錄的 Connect:Direct 程序記載層次。此內容值可以是無、失敗或全部。	無
檔案轉為訊息及訊息轉為檔案代理程式內容:		
deleteTmpFileAfterRenameFailure	將此內容設定為 false 值，可確保在重新命名作業失敗時，不會從目的地刪除暫存檔。在此情況下，已傳送的資料會保留在目的地的暫存檔 (.part) 中。稍後您可以手動重新命名此檔案。依預設，此內容的值為 true。此內容適用於「訊息轉為檔案」的傳送和檔案到檔案的傳送。	true
enableQueueInputOutput	依預設，在傳送過程中，代理程式無法讀取來源佇列中的資料，或將資料寫入目的地佇列。將此值設為 true，可讓代理程式執行「檔案轉為訊息」及「訊息轉為檔案」傳送。此內容值可以是 true 或 false。	false
enableSystemQueueInputOutput	指定代理程式是否可以從中讀取或寫入 IBM MQ 系統佇列。系統佇列以限定元 SYSTEM 為字首。 註: IBM MQ、Managed File Transfer 及其他應用程式會使用系統佇列來傳輸重要資訊。變更此內容可讓代理程式存取這些佇列。如果您啟用此內容，請以使用者沙盤推演來限制代理程式可存取的佇列。	false
enableClusterQueueInputOutput	指定代理程式是否可以從中讀取或寫入 IBM MQ 叢集佇列。 註: 除了 enableQueueInputOutput 內容外，您還必須指定 enableClusterQueueInputOutput 代理程式內容。	false
maxDelimiterMatchLength	在「檔案轉為訊息」的傳送過程中，可從將文字檔分割成多個訊息的 Java 正規表示式進行比對的字元數上限。	5
maxInputOutputMessageLength	代理程式從來源佇列讀取或寫入目的地佇列的訊息長度上限 (以位元組為單位)。在傳送時，來源代理程式的 maxInputOutputMessageLength 內容決定可從來源佇列上的訊息讀取的位元組數。在傳送時，目的地代理程式的 maxInputOutputMessageLength 內容決定可寫入目的地佇列上的訊息的位元組數。如果訊息長度超過此內容值，傳送會發生錯誤而失敗。此內容不影響 Managed File Transfer 內部佇列。如需變更此內容的相關資訊，請參閱第 270 頁的『設定與訊息大小相關聯的 MQ 屬性及 MFT 內容的指引』。	1048576

表 45: 進階代理程式內容 (繼續)		
內容名稱	說明	預設值
monitorGroupRetryLimit	<p>如果訊息群組仍存在於佇列上，監視器再次觸發「訊息轉為檔案」傳送的次數上限。從群組中第一個訊息的 MQMD 取消計數，來判定「訊息轉為檔案」的傳送觸發次數。</p> <p>如果重新啟動代理程式，則即使傳送的觸發次數超出 monitorGroupRetryLimit 的值，監視器仍會再次觸發傳送。如果此行為導致傳送的觸發次數超出 monitorGroupRetryLimit 的值，則代理程式會將錯誤寫入其事件日誌中。</p> <p>如果將此內容的值指定為 -1，監視器就會不限次數地重新觸發傳送，直至不符合觸發條件為止。</p>	10
一般代理程式內容:		
agentStatusPublishRateLimit	代理程式因為檔案傳送狀態變更而重新發佈其狀態的速率上限（以秒為單位）。如果此內容值設定太小，IBM MQ 網路的效能可能會受到負面影響。	30
agentStatusPublishRateMin	代理程式發佈其狀態的速率下限（以秒為單位）。此值必須大於或等於 agentStatusPublishRateLimit 內容的值。	300
enableMemoryAllocationChecking	<p>決定在接受傳送之前，Managed File Transfer Agent 是否確認為有足夠的記憶體可用來執行傳送。將會對來源及目的地代理程式進行此檢查。如果可用記憶體不足，將拒絕傳送。</p> <p>計算傳送所需的記憶體時，將會使用傳送所需的記憶體上限。因此，該值可能大於傳送所使用的實際記憶體。由於此原因，如果將 enableMemoryAllocationChecking 內容設定為 true，則可能會減少可以執行的並行傳送數目。建議您僅在遇到有關 Managed File Transfer 因記憶體不足錯誤而失敗的問題時，才將此內容設定為 true。可能會耗用大量記憶體的傳送，是訊息很大的「檔案轉為訊息」及「訊息轉為檔案」傳送。</p>	false
enableDetailedReplyMessages	<p>將此內容設定為 true，可指定受管理的傳輸要求回覆包含傳輸檔案的相關詳細資訊。此詳細資訊及格式，和進度訊息中發佈到傳送日誌的詳細資訊與格式相同，亦即 <transferSet> 元素。如需相關資訊，請參閱第 637 頁的『檔案傳送日誌訊息格式』。</p> <p>僅當受管理的傳輸要求指定必須包含詳細回覆資訊時，才會包含詳細回覆資訊。To specify this requirement, set the detailed attribute of the <reply> element of the managedTransfer XML request message sent to the source agent. 如需相關資訊，請參閱第 668 頁的『檔案傳送要求訊息格式』。</p> <p>可以為每個傳送要求產生多個回覆訊息。此數目等於傳送的傳送日誌進度訊息數加 1（其中，第一個回覆訊息為簡單 ACK 回覆）。除了 ACK 回覆訊息外，所有訊息中都包含詳細資訊，但整體傳送結果只包含在最後一個詳細回覆訊息中。</p>	true
enableUserMetadataOptions	<p>決定您是否可以在新的傳送要求中，以使用者定義 meta 資料的已知索引鍵，提供更多傳送選項。這些已知的索引鍵一律會使用下列字首 com.ibm.wmqfte. 作為開頭。因此，enableUserMetadataOptions 內容設為 true 時，使用此字首的索引鍵即不支援使用者定義之用。enableUserMetadataOptions 內容設為 true 時，目前支援的索引鍵如下：</p> <ul style="list-style-type: none"> com.ibm.wmqfte.insertRecordLineSeparator com.ibm.wmqfte.newRecordOnLineSeparator com.ibm.wmqfte.convertLineSeparators <p>如需這些索引鍵所代表意義的相關資訊，請參閱第 367 頁的『fteCreateTransfer: 啟動新的檔案傳送』。</p> <p>此內容值可以是 true 或 false。</p>	false
failTransferOnFirstFailure	<p>容許將代理程式配置成在受管理傳送內的傳送項目失敗時使受管理傳送失敗。</p> <p>若要啟用此功能，必須同時對來源代理程式及目的地代理程式套用 APAR ITO3450，並且在來源代理程式的 agent.properties 檔案中，failTransferOnFirstFailure 內容必須設為 true。在目的地代理程式中將此內容設定為 true 是選用項目。</p> <p>將 failTransferOnFirstFailure 內容設定為 true 時，代理程式會像平常一樣開始處理受管理傳送要求。但是，一旦某個傳送項目失敗，便會將受管理傳送標示為失敗，且不會處理任何其他傳送項目。在受管理傳送失敗之前已順利處理的傳送項目，將按下列方式進行處理：</p> <ul style="list-style-type: none"> 允許使用這些傳送項目的來源處置方式。例如，如果傳送項目的來源處置方式設定為 delete，則會刪除來源檔案。 已寫入的目的地檔案會保留在目的地檔案系統中，並且不會刪除。 <p>如果未將 failTransferOnFirstFailure 內容設定為 true，且受管理檔案傳送包含多個檔案，而其中一個檔案無法傳送（例如，因為目的地檔案已經存在，且 overwrite 內容設定為 error），則來源代理程式會繼續並嘗試傳送要求中的任何剩餘檔案。</p>	false


表 45: 進階代理程式內容 (繼續)		
內容名稱	說明	預設值
itemsPerProgressMessage	代理程式發佈其下一個進度日誌訊息之前，傳送的檔案數。使用此內容可控制傳送期間將進度日誌訊息發佈至協調佇列管理程式的比率。 此內容的上限值可設定為 1000。 註: 進度訊息包含從前次發佈進度訊息起，每個所傳送之檔案的相關資訊。增加此值會增加進度訊息的大小，從而會影響效能。	50
maxInlineFileSize	若為單一「檔案轉為檔案」或「檔案轉為訊息」的傳送，可自動將檔案大小上限（以位元組為單位）包含在起始傳送要求訊息中。 您可以使用此內容來提高傳送的速度，但如果您將檔案大小設定為太大的值，這可能會降低效能。此內容的建議起始大小為 100KB，但建議您徹底測試不同的值，直至您找到適用於系統的最佳檔案大小。	0
輸入/輸出內容:		
doNotUseTempOutputFile	依預設，代理程式會寫入目的地的暫存檔，而在檔案傳送完成之後會將此暫存檔重新命名為所需的檔案名稱。將此值設為 true，會讓代理程式直接寫入最終目的地檔案。  在 z/OS 系統上，此行為不適用於循序資料集，但適用於 PDS 資料集成員。 傳送的這個內容值是由目的地代理程式所定義。	false
enableMandatoryLocking	存取一般檔案時，Managed File Transfer 對讀取採取共用鎖定，對寫入採取專用鎖定。  在 Windows 上，檔案鎖定只是建議。此內容設為 true 時，Managed File Transfer 會施行檔案鎖定。在 Windows 上，這表示如果另一個應用程式已開啟檔案，則在檔案關閉之前不會觸發監視該檔案。含有該檔案的 Managed File Transfer 傳送會失敗。  在 UNIX 類型平台上，檔案鎖定會遍及程序。在 UNIX 類型平台上，設定此內容沒有效果。 此內容僅適用於一般 Managed File Transfer 代理程式。Managed File Transfer 不支援檔案伺服器上的檔案鎖定機制。因此，此內容不適用於通訊協定橋接器代理程式，因為通訊協定橋接器代理程式不會在傳送檔案時鎖定檔案伺服器上的檔案。 此內容值可以是 true 或 false。	false
ioIdleThreadTimeout	在執行緒關閉之前，檔案系統輸入/輸出執行緒持續閒置的時間（以毫秒為單位）。	10000
ioQueueDepth	排入佇列的輸入/輸出要求數目上限。	10
ioThreadPoolSize	可用的檔案系統輸入/輸出執行緒數目上限。通常每一個傳送都會使用自己的檔案系統輸入/輸出執行緒，但如果並行傳送次數超過此限制，傳送則會共用檔案系統輸入/輸出執行緒。 如果您認為進行中的並行傳送數可能經常性地大於 ioThreadPoolSize 值，則提高此值讓每一個傳送有自己的檔案系統輸入/輸出執行緒，應該就會改善情況。	10
textReplacementCharacterSequence	若為文字模式傳送，如果有任何資料位元組無法從來源字碼頁轉換為目的地字碼頁，則預設行為是讓檔案傳送失敗。 透過插入指定的字元值來設定此內容，以讓傳送順利完成。此內容值是單一字元。通常，問號 (?) 可用於任何無法對映的字元。例如，使用此格式 textReplacementCharacterSequence=?, 其中問號 (?) 是取代字元。空格字元不可作為取代字元。	無
多通道支援:		
agentMultipleChannelsEnabled	將此內容設為 true 可讓 Managed File Transfer Agent 跨多個 IBM MQ 通道傳送傳送資料訊息。在某些情況下，設定此內容或許可以改善效能。但請先確定效能有明顯改善，才啟用多通道支援。只有放置在 SYSTEM.FTE.DATA.destinationAgentName 佇列的訊息，才會跨多個通道傳送。其他所有訊息的行為維持不變。 將此內容設為 true 時，您也必須完成下列其中一個主題的 IBM MQ 配置步驟，以啟用多通道支援： • 第 478 頁的『在叢集中針對多個通道配置 MFT 代理程式』 • 第 479 頁的『針對多個通道配置 MFT 代理程式：非叢集』 此外，您還必須完成 Managed File Transfer 代理程式所需的標準 IBM MQ 配置步驟，這些步驟詳述於第 80 頁的『初次使用時配置 MFT』中。 此內容值可以是 true 或 false。	false

表 45: 進階代理程式內容 (繼續)		
內容名稱	說明	預設值
agentMessageBatchSize	配置多個通道時，來源代理程式會輪流透過每一個通道來傳送一個傳送作業的資料訊息。此內容會控制每一個通道每次傳送的訊息數目。	5
多重實例佇列管理程式內容：		
agentQMgrStandby	為 agentQMgr 所定義的多重實例代理程式佇列管理程式的待命實例，用於用戶端連線的主機名稱及埠號（格式為 IBM MQ CONNAME）。例如， <code>host_name(port_number)</code> 當代理程式偵測到連線中斷錯誤（例如 MQRC 2009）時，代理程式會嘗試連接至待命佇列管理程式。代理程式連接至待命佇列管理程式之後，代理程式會保持已連接狀態，直到待命佇列管理程式變成無法使用為止。	無預設值
程序控制器內容：		
agentQMgrRetryInterval	代理程式的程序控制器檢查佇列管理程式的可用性之間的間隔（以秒為單位）。	30
maxRestartCount	在 maxRestartInterval 內容值所指定的時間間隔內，可發生的重新啟動次數上限。當超出這個值時，代理程式的程序控制器會停止重新啟動代理程式，而改為根據 maxRestartDelay 內容值執行動作。	4
maxRestartInterval	代理程式的程序控制器測量代理程式重新啟動的間隔（以秒為單位）。如果在此間隔內重新啟動次數超出 maxRestartCount 內容值，代理程式的程序控制器會停止重新啟動代理程式。代理程式程序控制器會改為根據 maxRestartDelay 內容值執行動作。	120
maxRestartDelay	決定當代理程式重新啟動率超出 maxRestartCount 和 maxRestartInterval 內容值時，代理程式的程序控制器的行為。如果您指定的值小於或等於零，代理程式的程序控制器會停止。若您指定的值大於零，則必須先等待此秒數之後，才能再重設代理程式的程序控制器所保留的重新啟動歷程資訊，以及重新啟動代理程式。	-1
通訊協定橋接器內容：		
protocolBridgeCredentialConfiguration	此內容的值以字串形式傳遞給 protocolBridgeCredentialExitClasses 所指定之結束類別的 initialize() 方法。	空值
protocolBridgeCredentialExitClasses	指定以逗點區隔的類別清單，這些類別實作通訊協定橋接器認證使用者結束常式。如需相關資訊，請參閱第 188 頁的『使用結束類別對映檔案伺服器的認證』。	無預設值。
protocolBridgeDataTimeout	通訊協定橋接器代理程式等待建立與 FTP 伺服器的資料連線，或等待透過已建立的連線從 FTP 伺服器接收資料的逾時值（毫秒）。如果您將此內容設為 0 值，則通訊協定橋接器代理程式會無限期地等待。如果超過逾時值，通訊協定橋接器代理程式就會關閉所有與 FTP 伺服器的現有資料連線，並嘗試在回復現行傳送之前建立新的資料連線。如果嘗試建立新的資料連線失敗，則現行傳送也會隨之失敗。	0
protocolBridgeLogoutBeforeDisconnect	指定在關閉 FTP 階段作業及中斷連線之前，通訊協定橋接器代理程式是否將使用者登出檔案伺服器。如果您將此內容設定為 true，則通訊協定橋接器代理程式會向檔案伺服器發出 FTP QUIT 指令。	false
protocolBridgePropertiesConfiguration	作為其中一個橋接器內容而傳遞給 protocolBridgeServerPropertiesExitClasses 內容所指定之結束類別的 initialize() 方法。	無預設值
protocolBridgePropertiesExitClasses	指定以逗點區隔的類別清單，這些類別實作通訊協定橋接器伺服器內容使用者結束常式。 如需相關資訊，請參閱第 183 頁的『ProtocolBridgePropertiesExit2: 查閱通訊協定檔案伺服器內容』。	無預設值
通訊協定橋接器代理程式記載內容：		
  agentLog	鍵值組元件及作業，用來啟用或停用「通訊協定橋接器代理程式」與 FTP/SFTP/FTPS 檔案伺服器之間的 FTP 指令及回應的記載。 例如： agentLog=on 開啟所有元件的記載 agentLog=off 關閉所有元件的記載 agentLog=ftp=on, sftp=on, ftps=off 開啟 FTP 及 SFTP 的記載，並關閉 FTPS 的記載	無預設值
  agentLogFileSize	定義擷取檔的大小上限（以 MB 為單位）。與一般追蹤預設檔案大小的預設值相同。	20

表 45: 進階代理程式內容 (繼續)


內容名稱	說明	預設值
  agentLogFiles	定義在捨棄最舊檔案之前所保留的擷取檔數目上限。	10
  agentLogFilter	依預設，會擷取與代理程式所連接之所有 FTP 伺服器的通訊。 例如： <ul style="list-style-type: none"> 根據主機/IP 位址過濾 <pre>host=ftpprod.ibm.com, ftp2.ibm.com host=9.182.*</pre> 根據 meta 資料過濾 <pre>metadata="outbound files to xyz corp"</pre> 	*
佇列內容:		
publicationMDUser	與送出交由協調佇列管理程式發佈的訊息相關聯的 MQMD 使用者 ID。如果未設定此內容，則會根據用於設定 MQMD 使用者 ID 的 IBM MQ 規則來設定 MQMD 使用者 ID。	無預設值
資源監視內容:		
monitorFilepathPlatformSeparator	指定是否要在 \$FILEPATH 變數內使用平台專用的路徑分隔字元。值為 true 會使用平台專用的路徑分隔字元。值為 false 則會在所有平台上使用 UNIX 樣式的正斜線 (/) 路徑分隔字元。	true
monitorMaxResourcesInPoll	指定每一個輪詢間隔要觸發的受監視資源數目上限。例如，如果指定監視型樣為 *.txt、輪詢間隔為 10 秒，並將 monitorMaxResourcesInPoll 內容設為 10，則 monitorMaxResourcesInPoll 內容會限制代理程式在每一個輪詢間隔最多觸發 10 個相符項。超過上限 10 的相符資源將於後續的輪詢間隔觸發。 此外，您可以使用 monitorMaxResourcesInPoll 內容並結合 fteCreateMonitor 指令上相符的 -bs 參數，例如，用來限制每一個輪詢間隔只能觸發一個傳送。 指定的值小於或等於零，表示輪詢間隔中觸發的監視器資源數沒有限制。	-1
monitorReportTriggerFail	指定在監視器中偵測到有關環境及配置的失敗狀況，是否以日誌訊息形式回報給 SYSTEM.FTE 主題。值為 true 會記載訊息。值為 false 不會記載訊息。	true
monitorReportTriggerNotSatisfied	指定未滿足的觸發是否將包含詳細資料的日誌訊息傳送至 SYSTEM.FTE 主題。值為 true 會記載訊息。值為 false 不會記載訊息。	false
monitorReportTriggerSatisfied	指定滿足的觸發是否將包含詳細資料的日誌訊息傳送至 SYSTEM.FTE 主題。值為 true 會記載訊息。值為 false 不會記載訊息。	false
monitorSilenceOnTriggerFailure	資源監視器觸發的連續失敗次數，超過後就不再報告失敗。	5
monitorStopOnInternalFailure	資源監視器連續發生內部 FFDC 狀況的次數，超過後監視器的狀態就變成停止。	10
根目錄內容:		

表 45: 進階代理程式內容 (繼續)

內容名稱	說明	預設值
commandPath	<p>指定可呼叫指令的一組路徑，可以使用下列其中一種方法：</p> <ul style="list-style-type: none"> Agent Ant call、filecopy 或 filemove 作業 在傳遞給代理程式的 XML 訊息中，使用其中一個支援的 Managed File Transfer Agent 指令 XML 綱目（例如，managedCall 或 managedTransfer）。 <p>如需 commandPath 內容值有效語法的相關資訊，請參閱第 306 頁的『commandPath MFT 內容』。</p> <p>重要：在設定此內容時請特別小心，因為可以從遠端用戶端系統有效地呼叫其中一個指定的 commandPaths 中的任何指令，從而可以將指令傳送至代理程式。因此，依預設，當您指定 commandPath 時：</p> <ul style="list-style-type: none"> 任何現有代理程式沙盤推演都會由代理程式在啟動時進行配置，讓所有 commandPath 目錄於傳送時自動新增至拒絕存取的目錄清單中。 任何現有使用者沙盤推演都會在代理程式啟動時進行更新，以讓所有 commandPath 目錄（及其子目錄）都作為 <exclude> 元素新增至 <read> 及 <write> 元素。 如果代理程式未配置為使用代理程式沙盤推演或使用者沙盤推演，當代理程式啟動時，會建立新的代理程式沙盤推演，將 commandPath 目錄指定為被拒絕的目錄。 <p>此內容的值可以包含環境變數。</p> <p>如需相關資訊，請參閱第 452 頁的『在 MFT 內容中使用環境變數』。</p> <p>V9.0.0.1 為了與下列版次相容，您可以將 addCommandPathToSandbox 內容設為 false，以置換此預設行為：</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 元件（或更舊版本）。 IBM WebSphere MQ 7.5.0 Fix Pack 2 Managed File Transfer 元件（或更舊版本），其所在的安裝環境中並未設定 enableFunctionalFixPack=7502 安裝內容。 <p>重要：請注意，此置換實際上會讓用戶端將全部指令傳送至代理程式系統並呼叫該指令，因此使用時要非常小心。</p>	無 - 無法呼叫任何指令
<p>V9.0.0.1</p> <p>V9.0.0.1</p> <p>addCommandPathToSandbox</p>	<p>指定 commandPath 內容所指定的目錄（及其所有子目錄）是否應該新增至：</p> <ul style="list-style-type: none"> 現有代理程式沙盤推演的被拒絕目錄。 已定義之任何使用者沙盤推演的 <read> 和 <write> 元素的 <exclude> 元素。 新的代理程式沙盤推演（如果代理程式尚未配置其代理程式沙盤推演或一個以上的使用者沙盤推演）。 <p>這提供與下列版次的相容性：</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 元件（或更舊版本）。 IBM WebSphere MQ 7.5.0 Fix Pack 2 Managed File Transfer 元件（或更舊版本），其所在的安裝環境中並未設定 enableFunctionalFixPack=7502 安裝內容。 <p>如需相關資訊，請參閱第 306 頁的『commandPath MFT 內容』。</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>如果代理程式已配置成利用使用者或代理程式沙盤推演以便限制可以與代理程式來回傳送檔案的位置，則指定是否要對該代理程式的萬用字元傳送執行其他檢查。</p> <p>如果此內容設為 true，則啟用其他檢查。如果傳送要求嘗試讀取的位置與萬用字元相符，但位於定義給檔案的沙盤推演外部，則傳送會失敗。如果在一個傳送要求中進行多個傳送，而且其中一個要求由於嘗試讀取沙盤推演外部的位址而失敗，則整個傳送都會失敗。如果檢查失敗，則錯誤訊息會提供失敗原因（請參閱第 43 頁的『對萬用字元傳送進行其他檢查』）。</p> <p>如果內容已省略或設為 false，則不會對萬用字元傳送進行其他檢查。</p>	無

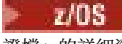


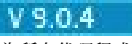
表 45: 進階代理程式內容 (繼續)		
內容名稱	說明	預設值
sandboxRoot	<p>指定使用沙盤推演時要併入及排除的一組根路徑。如需此特性的相關資訊，請參閱第 39 頁的『使用 MFT 代理程式沙盤推演』。</p> <p>請以平台專用路徑分隔字元來區隔路徑。以驚嘆號 (!) 字元作為路徑的字首，以表示從沙盤推演中排除的路徑。如果在包括的根路徑下要排除某個子目錄，此特性會很有幫助。</p> <p>通訊協定橋接器代理程式不支援 sandboxRoot 內容。</p> <p>您不可同時指定 sandboxRoot 內容及 userSandboxes 內容。</p> <p>此內容的值可以包含環境變數。</p> <p>如需相關資訊，請參閱第 452 頁的『在 MFT 內容中使用環境變數』。</p>	無 - 無沙盤推演
transferRoot	<p>指定給代理程式之相對路徑的預設根目錄。</p> <p>此內容的值可以包含環境變數。</p> <p>如需相關資訊，請參閱第 452 頁的『在 MFT 內容中使用環境變數』。</p>	啟動代理程式程序之使用者的起始目錄。
transferRootHLQ	指定給代理程式的非完整資料集的預設 HLQ (使用者 ID)	啟動代理程式程序之使用者的使用者名稱。
userSandboxes	<p>根據要求傳送的使用者的 MQMD 使用者名稱，限制可來回傳送檔案的檔案系統區域。如需相關資訊，請參閱第 40 頁的『使用 MFT 使用者沙盤推演』。</p> <p>通訊協定橋接器代理程式不支援 userSandboxes 內容。</p> <p>您不可同時指定 sandboxRoot 內容及 userSandboxes 內容。</p>	false
排程器內容:		
maxSchedulerRunDelay	<p>代理程式等待檢查排定傳送的間隔上限 (以分鐘為單位)。指定正整數以啟用此內容。如需為何使用此內容的相關資訊，請參閱第 264 頁的『排定的檔案傳送未執行或延遲時應該如何處理』。</p> <p>在排定傳送到期而應該執行時，因為代理程式可能正從其指令佇列中讀取指令，所以排定傳送開始之前可能會稍微延遲。在此情況下，排程器會在指令完成後立即執行。</p>	-1
安全內容:		
agentQMgrAuthenticationCredentials File	包含 MQ 連線認證的檔案路徑。	<p> 如需建立「鑑別認證檔」的詳細資料，請參閱 建立 MFT 認證檔</p> <p> 如需此檔案的位置及許可權的相關資訊，請參閱 配置 MQMFTCredentials.xml。</p> <p> 關於建立「鑑別認證檔」的進一步詳細資料位於 MFT 和 IBM MQ 連線鑑別 中</p>
authorityChecking	<p>指定是否啟用第 301 頁的『限制 MFT 代理程式動作的使用者權限』中所述的安全特性。</p> <p>  從 IBM MQ 9.0.0 Fix Pack 3 及 IBM MQ 9.0.4 開始，inquire 為所有代理程式權限佇列的必要權限。</p>	false
logAuthorityChecks	在 output0.log 檔的代理程式事件日誌中記錄的權限檢查記載層次。此內容值可以是無、失敗或全部。	無
userIdForClientConnect	透過用戶端連線傳送至 IBM MQ 的使用者 ID。如果指定了 java，則會隨著 IBM MQ 連線要求而傳送 JVM 所回報的使用者名稱。此內容值可以是無或 java。	無
逾時內容:		
maxTransferNegotiationTime	<p>傳送等待讓目的地代理程式完成協議的時間上限 (以毫秒為單位)。如果協議未在此時間內完成，傳送會進入重新同步狀態，並允許執行另一個傳送 (若有的話)。</p> <p>在來源或目的地代理程式處於高負載的情況下，可能會因為預設值過低，而使得代理程式回應協議要求的速度不夠快。當來源代理程式定義了大量的資源監視器時，或是其資源監視器正在監視包含大量檔案的目錄時，就可能發生這種情況。另外，提交給代理程式大量的傳送要求時，也可能會發生此情況。在此情況下，可能需要將此值加大至 200,000 或更高。</p>	30 000
recoverableTransferRetryInterval	在偵測可回復的傳送錯誤與嘗試回復傳送之間等待的時間 (以毫秒為單位)。	60 000

表 45: 進階代理程式內容 (繼續)		
內容名稱	說明	預設值
senderTransferRetryInterval	由於目的地執行的傳送次數已達上限，重試被拒絕的傳送之前等待的時間（以毫秒為單位）。下限值是 1000。	30 000
transferAckTimeout	發出重試之前，傳送作業等待確認通知或另一端資料的逾時（以毫秒為單位）。此為進階內容，且對大多數的 Managed File Transfer 配置而言，並不需要修改其值。 每當收到完整範圍的資料時，接收端代理程式會傳送確認通知給傳送端代理程式。如果頻寬受限或網路不穩定且 agentWindowSize 及 agentChunkSize 設定較大，預設值可能不夠。這會造成代理程式之間進行不必要的資料重新傳送。因此，提高此值可能比較有利，也可以降低因為網路太慢使得傳送進入回復模式的可能性。	60 000
transferAckTimeoutRetries	在代理程式放棄而造成傳送進入回復狀態之前，無回應的重試傳送確認通知次數上限	5
xmlConfigReloadInterval	在執行時期，代理程式重新載入 XML 配置檔之間的時間（以秒為單位）。若要防止代理程式在執行時期重新載入 XML 配置檔，請將此內容設定為 -1。下列 XML 配置檔會受此內容影響： <ul style="list-style-type: none"> ConnectDirectCredentials.xml ConnectDirectNodeProperties.xml ConnectDirectProcessDefinitions.xml ProtocolBridgeCredentials.xml ProtocolBridgeProperties.xml UserSandboxes.xml 	30
追蹤及記載內容：		
javaCoreTriggerFile	代理程式所監視之檔案位置的完整路徑。如果檔案存在於指定的位置，代理程式啟動會觸發 Javacore。啟動代理程式之後，如果您更新此位置的檔案，代理程式會再次觸發 Javacore 檔案。 有一個個別執行緒會每隔 30 秒輪詢一次該檔案，以檢查是否已建立或更新檔案。如果在前次輪詢之後，已經建立或更新檔案，代理程式會在下列其中一個目錄中產生 Javacore 檔案： <ul style="list-style-type: none"> UNIX: MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name Linux: MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name Windows: MQ_DATA_PATH\mqft\logs\coordination_qmgr_name\agents\agent_name 指定此內容時，代理程式啟動時會輸出下列訊息： <pre>BFGAG0092I The <insert_0> file will be used to request JVM diagnostic information.</pre> 此內容的值可以包含環境變數。 如需相關資訊，請參閱第 452 頁的『在 MFT 內容中使用環境變數』。	無
trace	代理程式啟動時，在啟用追蹤的情況下執行代理程式的追蹤指定。追蹤指定是以逗點區隔的類別清單、等於字元及追蹤層次。例如，com.ibm.wmqfte.agent.Agent,com.ibm.wmqfte.commandhandler=all。您可以使用以冒號區隔的清單來指定多個追蹤指定。例如，com.ibm.wmqfte.agent.Agent=all:com.ibm.wmqfte.commandhandler=moderate。	無
outputLogFiles	要保留的 output.log 檔總數。此值適用於代理程式的程序控制器，以及代理程式本身。	5
outputLogSize	每一個 output.log 檔在輸出覆蓋到下一個檔案之前的大小上限（以 MB 為單位）。此值適用於代理程式的程序控制器，以及代理程式本身。	1
outputLogEncoding	代理程式寫入 output.log 檔時使用的字元編碼。	代理程式執行所在平台的預設字元編碼。
traceFiles	要保留的追蹤檔總數。此值適用於代理程式的程序控制器，以及代理程式本身。	5
traceSize	每一個追蹤檔在追蹤覆蓋到下一個檔案之前的大小上限（以 MB 為單位）。此值適用於代理程式的程序控制器，以及代理程式本身。	20
traceMaxBytes	在追蹤檔中輸出的訊息資料量限制。	4096 個位元組

表 45: 進階代理程式內容 (繼續)		
內容名稱	說明	預設值
logTransferRecovery	當此內容設為值 true 時，每當傳送進入回復狀態，就會在 output0.log 檔中向代理程式的事件日誌報告診斷事件。	對於 IBM MQ 9.0.0.0，預設值為 true
logCapture	擷取提交給此代理程式的傳送要求訊息，以及代理程式發佈給協調佇列管理程式的日誌訊息。這些擷取的訊息有助於進行傳送問題的除錯。擷取的訊息儲存在代理程式日誌目錄中稱為 capture?.log 的檔案。? 是數值。包含數字 0 的檔案，保存有最新的擷取訊息。	false
logCaptureFileSize	定義擷取檔的大小上限（以 MB 為單位）。	10
logCaptureFiles	定義在捨棄最舊檔案之前所保留的擷取檔數目上限。	10
logCaptureFilter	Java 正規表示式，代理程式可用此比對訊息的主題名稱。只會擷取符合正規表示式的訊息。	* (全部符合)
  resourceMonitorLog	<p>要開啟或關閉記載功能之資源監視器和作業的鍵值組。</p> <p>可能值包括：</p> <ul style="list-style-type: none"> • info • moderate • verbose • off <p>例如：</p> <ul style="list-style-type: none"> • resourceMonitorLog=MON1,MON2=info:MON3=off 開啟 MON1 和 MON2 的記載功能，並關閉 MON3 的記載功能。 • resourceMonitorLog=info 為所有資源監視器開啟 info 層次記載功能。 <p>資源監視器日誌會寫入名為 resmoneventN.log 的檔案，其中 N 代表數字；例如 resmonevent0.log。</p> <p> 小心：一個代理程式的所有資源監視器都會寫入相同的日誌檔。</p> <p>如需相關資訊，請參閱記載 MFT 資源監視器。</p>	info
  resourceMonitorLogFileSize	定義擷取檔的大小上限（以 MB 為單位）。	20
  resourceMonitorLogFiles	定義在捨棄最舊檔案之前所保留的擷取檔數目上限。	10
傳送限制內容：		
maxDestinationTransfers	<p>目的地代理程式在任何復原點處理的並行傳送數目上限。不論傳送多少檔案來滿足要求，提交至代理程式的每一個傳送要求都會計入此總計。這表示不論傳送要求是傳送單一檔案或 10 個檔案，其計算方式相同。</p> <p>當目的地代理程式達到 maxDestinationTransfers 內容所指定的限制時，代理程式就會將傳送排入佇列中。</p> <p>如果下列代理程式內容值的總和：maxSourceTransfers + maxDestinationTransfers + maxQueuedTransfers 超出狀態儲存佇列 (SYSTEM.FTE.STATE.agent name) 的 MAXDEPTH 設定值，代理程式不會啟動。</p>	<p>25 (適用於 Connect:Direct 之外的所有代理程式)</p> <p>5 (適用於 Connect:Direct 橋接器代理程式)</p>
maxFilesForTransfer	<p>容許單一受管理傳送包含的傳送項目數上限。如果受管理傳送包含的項目數大於 maxFilesForTransfer 的值，則受管理傳送會失敗且不會處理任何傳送項目。</p> <p>設定此內容可避免因為傳送要求不當而不慎傳送太多檔案，例如，如果使用者不慎將傳送指定為 UNIX 系統上的根目錄 /。</p>	5000
maxSourceTransfers	<p>來源代理程式在任何復原點處理的並行傳送數目上限。不論傳送多少檔案來滿足要求，提交至代理程式的每一個傳送要求都會計入此總計。這表示不論傳送要求是傳送單一檔案或 10 個檔案，其計算方式相同。</p> <p>當目的地代理程式達到 maxSourceTransfers 內容所指定的限制時，代理程式就會將傳送排入佇列中。</p> <p>如果下列代理程式內容值的總和：maxSourceTransfers + maxDestinationTransfers + maxQueuedTransfers 超出狀態儲存佇列 (SYSTEM.FTE.STATE.agent name) 的 MAXDEPTH 設定值，代理程式不會啟動。</p>	<p>25 (適用於 Connect:Direct 橋接器代理程式之外的所有代理程式)</p> <p>5 (適用於 Connect:Direct 橋接器代理程式)</p>

表 45: 進階代理程式內容 (繼續)		
內容名稱	說明	預設值
maxQueuedTransfers	<p>在代理程式拒絕新的傳送要求之前，可由來源代理程式排入佇列的擱置傳送數目上限。設定此內容後，不管是符合或超出 maxDestinationTransfers 及 maxSourceTransfers 的限制，都會接受您目前提交的任何新傳送要求並排入佇列中，於稍後執行。</p> <p>排入佇列的傳送要求會依其優先順序及排入佇列中的時間長短來決定處理順序。優先選取較舊及較高優先順序的擱置傳送。優先順序較低且在佇列中擱置很久的傳送，其選取順序優先於較新、較高優先順序的傳送。</p> <p>如果下列代理程式內容值的總和：maxSourceTransfers + maxDestinationTransfers + maxQueuedTransfers 超出狀態儲存佇列 (SYSTEM.FTE.STATE.agent name) 的 MAXDEPTH 設定值，代理程式不會啟動。</p>	1000
<p>V 9.0.1 V 9.0.1</p> <p>傳送回復逾時內容:</p>		
<p>V 9.0.1</p> <p>V 9.0.1</p> <p>transferRecoveryTimeout</p>	<p>設定時間量 (以秒為單位)，在此期間內，來源代理程式會一直嘗試回復已停止的檔案傳送。</p> <p>若未設定此內容，代理程式的預設行為是一直重試，直至順利回復傳送為止。您可以為傳送回復逾時內容設定下列值：</p> <p>-1 代理程式繼續嘗試回復已停止的傳送，直至傳送完成為止。使用此選項相當於代理程式在未設定此內容時的預設行為。</p> <p>0 一旦進入回復，代理程式即停止檔案傳送。</p> <p>>0 代理程式繼續嘗試回復已停止的傳送，直至達到已指定的正整數值所設定的時間量 (以秒為單位) 為止。例如，transferRecoveryTimeout=21600 指出代理程式從進入回復的 6 小時內會一直嘗試回復傳送。此參數的最大值為 99999999。</p>	-1
使用者結束程式內容:		
agentForceConsistentPathDelimiters	強制提供給傳送結束程式的來源檔案及目的地檔案資訊中的路徑定界字元為 UNIX 樣式: 正斜線 (/)。有效選項為 true 和 false。	false
destinationTransferEndExitClasses	指定以逗點區隔的類別清單，這些類別實作目的地傳送使用者結束程式常式。	無預設值
destinationTransferStartExitClasses	指定以逗點區隔的類別清單，這些類別實作目的地傳送開始使用者結束程式常式。	無預設值
exitClassPath	指定平台專用的、字元定界的目錄清單，這些目錄充當使用者結束程式常式的類別路徑。 在此類別路徑中的任何項目之前，會先搜尋代理程式結束程式目錄。	代理程式的 exits 目錄
exitNativeLibraryPath	指定平台專用的、字元定界的目錄清單，這些目錄充當使用者結束程式常式的原生程式庫路徑。	代理程式的 exits 目錄
ioMaxRecordLength	記錄導向檔案可支援的記錄長度上限 (以位元組為單位)。Managed File Transfer 可支援寫入任何記錄長度的記錄導向檔案。不過，較大的記錄長度可能會導致記憶體不足錯誤，因此為了避免這些錯誤，依預設，記錄長度上限會限制為 64 K。從記錄導向檔案讀取時，整個記錄必須適合單一傳送區塊，因此記錄長度額外受到傳送區塊大小的限制。此內容僅適用於 I/O 使用者結束程式記錄導向的檔案。	64KB
monitorExitClasses	指定以逗點區隔的類別清單，這些類別實作監視器結束常式。如需相關資訊，請參閱 第 755 頁的『MFT 資源監視器使用者結束程式』。	無預設值
protocolBridgeCredentialExitClasses	指定以逗點區隔的類別清單，這些類別實作通訊協定橋接器認證使用者結束常式。如需相關資訊，請參閱 第 188 頁的『使用結束類別對映檔案何服务器的認證』。	無預設值。
sourceTransferEndExitClasses	指定以逗點區隔的類別清單，這些類別實作來源傳送結束使用者結束程式。	無預設值
sourceTransferStartExitClasses	指定以逗點區隔的類別清單，這些類別實作來源傳送開始使用者結束程式。	無預設值
IOExitClasses	指定以逗點區隔的類別清單，這些類別實作 I/O 使用者結束常式。僅列出實作 IOExit 介面的類別，亦即，不列出實作其他 I/O 使用者結束程式介面的類別，例如 IOExitResourcePath 及 IOExitChannel。如需相關資訊，請參閱 第 218 頁的『使用 MFT 傳送 I/O 使用者結束程式』。	無預設值。
IBM MQ 用戶端壓縮:		

表 45: 進階代理程式內容 (繼續)		
內容名稱	說明	預設值
agentDataCompression	僅用戶端連線才支援此內容。 檔案資料的傳送與遠端 IBM MQ 伺服器協議的壓縮類型清單 (以逗點區隔)。您可以在下列主題中找到這些壓縮類型的相關資訊: 訊息資料壓縮清單。 會檢查值是否有效, 然後以內容形式將外觀傳遞至代理程式用戶端通道。然後, IBM MQ 用戶端會處理此用戶端通道與遠端伺服器通道之間的協議, 在兩個通道上的壓縮內容之間找出相符的最小公分母。如果找不到相符項目, 則一律選取 MQCOMPRESS_NONE。	MQCOMPRESS_NONE
agentHeaderCompression	僅用戶端連線才支援此內容。 標頭資料的傳送與遠端 IBM MQ 伺服器協議的壓縮類型清單 (以逗點區隔)。接受值是 MQCOMPRESS_NONE 或 MQCOMPRESS_SYSTEM。您可以在下列主題中找到這些壓縮類型的相關資訊: 訊息標頭壓縮清單。 會檢查值是否有效, 然後以內容形式將外觀傳遞至代理程式用戶端通道。然後, IBM MQ 用戶端會處理此用戶端通道與遠端伺服器通道之間的協議, 在兩個通道上的壓縮內容之間找出相符的最小公分母。如果找不到相符項目, 則一律選取 MQCOMPRESS_NONE。	MQCOMPRESS_NONE
  z/OS 專用:		
      adminGroup	安全管理員群組。此群組的成員可以: <ul style="list-style-type: none"> • 使用 fteStartAgent 指令來啟動代理程式。 • 使用 fteStopAgent 指令來停止代理程式。 • 使用 fteSetAgentTraceLevel 指令來啟用或停用代理程式追蹤。 • 執行已指定 -d 參數的 fteShowAgentDetails 指令來顯示本端代理程式的詳細資料。 定義安全管理員群組 (例如 MFTADMIN) , 然後新增已啟動作業使用者 ID 和管理者 TSO ID 至此群組。編輯代理程式內容檔, 並將 adminGroup 內容設為此安全管理員群組的名稱。 <pre>adminGroup=MFTADMIN</pre>	無
  bpxwdynAllocAdditionalOptions	Managed File Transfer 使用 BPXWDYN 文字介面來建立及開啟 z/OS 資料集。資料集配置使用 BPXWDYN 時, 依預設, Managed File Transfer 會盡可能確定已裝載資料裝置 (對於磁碟型資料集而言非必要項目, 但對於磁帶資料集而言為必要項目)。因為某些環境可能不支援選項, 請使用此內容來變更此行為。另外, 傳送至資料集時, 也可以在指令行指定 BPXWDYN 的選項; 這些是此內容指定的選項之外的額外選項。 在 agent.properties 檔中使用 bpxwdynAllocAdditionalOptions 內容時, 不得指定部分 BPXWDYN 選項。如需這些內容的清單, 請參閱第 540 頁的『不得與 MFT 一起使用的 BPXWDYN 內容』。	預設值如下所示: <ul style="list-style-type: none"> • MOUNT, 適用於 z/OS V1R8 以及更新版本
  armELEMENTYPE	選用內容。如果代理程式配置為由 Automatic Restart Manager (ARM) 重新啟動, 則將此內容設為相關聯 ARM 原則中指定的 ARM ELEMENTYPE 參數值。對於代理程式, 請將 ELEMENTYPE 設為 SYSBFGAG。	未設定
  armELEMENT	選用內容。如果代理程式配置為由 Automatic Restart Manager (ARM) 重新啟動, 則將此內容設為相關聯 ARM 原則中指定的 ARM ELEMENT 參數值。您可以將 ELEMENT 值設為對應於代理程式名稱。	未設定
其他內容:		
  legacyXMLMessageMQMDFormat	由代理程式產生的 Managed File Transfer XML 訊息 (例如, 日誌及傳送進度訊息) 現在會傳送至具有空白 MQMD 格式欄位的佇列。舊版的產品將 MQMD 格式欄位設為 MQSTR (文字訊息字串)。將此內容設為 true, 可將代理程式所產生的 Managed File Transfer XML 訊息傳送至 MQMD 格式欄位為 MQSTR 的佇列。 註: 代理程式回覆給指令的訊息, 將會以符合對應指令要求的訊息格式傳送。 如果 MQMD 格式欄位設為 MQSTR, 而 MQ 網路中有通道啟用了資料轉換, 則 Managed File Transfer 指令 XML 訊息有可能已毀損。	false
 adjustScheduleTimeForDaylightSavings	如果您的企業每天執行排程傳送, 因為排程傳送是使用建立的: <ul style="list-style-type: none"> • -oi 參數設為天, 以及 • -tb 參數設為 source 例如, 在 fteCreateTransfer 指令上, 將此內容設為 true 會在時鐘往前一小時, 以及在時鐘往回一小時時, 將排定的傳送時間往前移動一小時。 例如, 如果排定的傳送應在凌晨 1:00 執行, 當時鐘往前時, 傳送將在凌晨 2:00 執行, 當時鐘往回時, 傳送會回復為凌晨 1:00。	true

相關概念

第 49 頁的『[多平台上的 MFT 配置選項](#)』

Managed File Transfer 提供了一組內容檔，這些檔案包含有關您的設定的金鑰資訊，是作業所必要的。這些內容檔位於安裝產品時所定義的配置目錄中。

第 206 頁的『[回復中檔案傳送的逾時選項](#)』

您可以設定時間量（以秒為單位），在此期間內，來源代理程式會一直嘗試回復已停止的檔案傳送。如果代理程式達到重試間隔的逾時值時傳送未順利完成，則傳送失敗。

第 39 頁的『[MFT 沙盤推演](#)』

您可以限制代理程式在傳送過程中可存取的檔案系統區域。代理程式受限的區域稱為沙盤推演。您可以將限制套用至代理程式或要求傳送的使用者。

相關參考

第 508 頁的『[MFT 的 Java 系統內容](#)』

有一些 Managed File Transfer 指令及代理程式內容必須定義為 Java 系統內容，因為它們為無法使用指令或代理程式內容機制的早期功能定義配置。

第 508 頁的『[MFT 的 SSL 內容](#)』

對 IBM MQ 及 Managed File Transfer 使用 SSL 或 TLS，可避免代理程式與佇列管理程式之間未獲授權的連線，以及加密代理程式與佇列管理程式之間的訊息資料流量。

第 326 頁的『[fteCreateAgent（建立 MFT 代理程式）](#)』

fteCreateAgent 指令會建立 Managed File Transfer Agent 及其相關聯的配置。

第 330 頁的『[fteCreateBridgeAgent（建立及配置 MFT 通訊協定橋接器代理程式）](#)』

fteCreateBridgeAgent 指令會建立 Managed File Transfer 通訊協定橋接器代理程式及其相關聯的配置。為您要傳送檔案至其中及從中接收檔案的每一台檔案伺服器建立通訊協定橋接器代理程式。

第 339 頁的『[fteCreateCDAgent（建立 Connect:Direct 橋接器代理程式）](#)』

fteCreateCDAgent 指令可建立 Managed File Transfer Agent 及其與 Connect:Direct 橋接器搭配使用的關聯配置。

第 452 頁的『[在 MFT 內容中使用環境變數](#)』

從 IBM WebSphere MQ 7.5 開始，可能會在代表檔案或目錄位置的 Managed File Transfer 內容中使用環境變數。這可讓執行產品的某部分時所使用的檔案或目錄，隨著環境變更而改變其所在位置。例如，執行程序的使用者為何。

在叢集中針對多個通道配置 MFT 代理程式

如果您想要在叢集配置中使用 IBM MQ 多通道支援，請先將 `agentMultipleChannelsEnabled` 內容設定為 `true`，然後完成此主題中的步驟。

關於這項作業

在叢集中，僅透過目的地代理程式的佇列管理程式上的 IBM MQ 定義啟用多通道支援。

除了 Managed File Transfer 代理程式所需的標準 IBM MQ 配置步驟之外，您還必須完成本主題中的步驟，這些步驟列在 [第 80 頁的『初次使用時配置 MFT』](#) 中。

下列配置範例使用 `runmqsc` 指令。

程序

1. 對您要使用的每一個通道定義叢集接收端通道。比方說，例如您使用以下兩個通道：

```
DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(CLUSRCVR) CLUSTER(MFTCLUSTER)
DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(CLUSRCVR) CLUSTER(MFTCLUSTER)
```

其中：

- `DESTQMGRNAME` 是目的地代理程式的佇列管理程式名稱。
- `MFTCLUSTER` 是 IBM MQ 叢集的名稱。

建議讓通道使用 `MFTCLUSTER.DESTMGRNAME_n` 命名慣例，但非必要。

2. 定義對應於每一個通道的佇列管理程式別名。 例如：

```
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_1) RQMNAME(DESTQMGRNAME) CLUSTER(MFTCLUSTER)
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_2) RQMNAME(DESTQMGRNAME) CLUSTER(MFTCLUSTER)
```

因為傳送代理程式會搜尋 SYSTEM.FTE.DESTQMGRNAME_n 格式的佇列管理程式別名，所以您佇列管理程式別名必須使用此命名慣例。您用於 n 的數字必須以 1 開頭，且是連續的。您必須將定義設定為叢集全域，因此可用於來源代理程式的佇列管理程式。

為使來源代理程式及目的地代理程式都能正確地決定佇列管理程式別名的數目，請**不要**為佇列管理程式定義預設 XMITQ。

相關概念

第 80 頁的『初次使用時配置 MFT』

初次使用 Managed File Transfer 代理程式及佇列管理程式時，必須先執行某些配置作業。

相關參考

第 463 頁的『MFT agent.properties 檔案』

每一個 Managed File Transfer Agent 都有自己的內容檔 agent.properties，其中必須包含代理程式用來連接至其佇列管理程式的資訊。agent.properties 檔案也可以包含變更代理程式行為的內容。

針對多個通道配置 MFT 代理程式：非叢集

如果您想要在非叢集配置中使用 IBM MQ 多通道支援，請先將 agentMultipleChannelsEnabled 內容設定為 true，然後完成此主題中的步驟。

關於這項作業

在非叢集配置中，透過來源代理程式及目的地代理程式的佇列管理程式上的 IBM MQ 定義啟用多通道支援。

除了 Managed File Transfer 代理程式所需的標準 IBM MQ 配置步驟之外，您還必須完成本主題中的步驟，這些步驟列在 第 80 頁的『初次使用時配置 MFT』中。

下列步驟假設將使用傳送端至接收端通道，在來源與目的地佇列管理程式之間通訊。

下列配置範例使用 **runmqsc** 指令。

程序

1. 在目的地代理程式的佇列管理程式上，為要使用的每一個通道定義接收端通道。比方說，例如您使用以下兩個通道：

```
DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(RCVR) TRPTYPE(TCP)
DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(RCVR) TRPTYPE(TCP)
```

其中：DESTQMGRNAME 是目的地代理程式的佇列管理程式名稱。

建議您對通道使用 TO.DESTMGRNAME_n 命名慣例，但此慣例不是必要的。接收端通道名稱必須與來源代理程式的佇列管理程式上對應的傳送端通道相符。

2. 在來源代理程式的佇列管理程式上，為要使用的每一個通道定義傳輸佇列。比方說，例如您使用以下兩個通道：

```
DEFINE QLOCAL(DESTQMGRNAME_1) USAGE(XMITQ)
DEFINE QLOCAL(DESTQMGRNAME_2) USAGE(XMITQ)
```

建議您對傳輸佇列使用 DESTMGRNAME_n 命名慣例，但此慣例不是必要的。透過下列步驟中的傳送端通道定義及佇列管理程式別名定義，參照您定義的傳送佇列。

3. 在來源代理程式的佇列管理程式上，為要使用的每一個通道定義傳送端通道。比方說，例如您使用以下兩個通道：

```
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)
```

建議您對通道使用 TO.DESTMGRNAME_n 命名慣例，但此慣例不是必要的。傳送端通道名稱必須與目的地代理程式的佇列管理程式上對應的接收端通道相符。

4. 在來源代理程式的佇列管理程式上，定義對應於每一個通道的佇列管理程式別名。例如：

```
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_1) RQMNAME(DESTQMGRNAME) XMITQ(DESTQMGRNAME_1)
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_2) RQMNAME(DESTQMGRNAME) XMITQ(DESTQMGRNAME_2)
```

您必須對佇列管理程式別名使用 SYSTEM.FTE.DESTQMGRNAME_n 命名慣例，因為傳送代理程式會搜尋此格式的佇列管理程式別名。您用於 n 的數字必須以 1 開頭，且是連續的。

為使代理程式正確地決定佇列管理程式別名的數目，請不要為佇列管理程式定義預設 XMITQ。

相關概念

第 80 頁的『初次使用時配置 MFT』

初次使用 Managed File Transfer 代理程式及佇列管理程式時，必須先執行某些配置作業。

相關參考

第 463 頁的『MFT agent.properties 檔案』

每一個 Managed File Transfer Agent 都有自己的內容檔 agent.properties，其中必須包含代理程式用來連接至其佇列管理程式的資訊。agent.properties 檔案也可以包含變更代理程式行為的內容。

enableUserMetadataOptions：支援的 MFT 使用者定義 meta 資料索引鍵

如果代理程式內容 enableUserMetadataOptions 設為 true 值，在指定給新的傳送要求時，會支援下列使用者定義 meta 資料索引鍵。

金鑰名稱	說明	預設值
com.ibm.wmqfte.insertRecordLineSeparator	適用於文字傳送。此索引鍵若是設為 true，會指定在讀取記錄導向的檔案（例如 z/OS 資料集）時，於記錄之間插入行分隔字元。 此索引鍵若是設為 false，會指定在讀取記錄導向的檔案時，不要在記錄之間插入行分隔字元。	true
com.ibm.wmqfte.newRecordOnLineSeparator	適用於文字傳送。此索引鍵若是設為 true，會指定在寫入記錄導向的檔案（例如 z/OS 資料集）時，行分隔字元代表新記錄，而不是作為資料的一部分寫入。 此索引鍵若是設為 false，會指定在寫入記錄導向的檔案時，將行分隔字元視為其他任何字元（換言之便是沒有記錄岔斷）。	true
com.ibm.wmqfte.convertLineSeparators	適用於文字傳送。指定是否要將行分隔字元順序 CRLF 及 LF 轉換為目的地的所需的行分隔字元順序。此轉換目前僅適用下列狀況： 1. 當使用者定義 meta 資料索引鍵 com.ibm.wmqfte.newRecordOnLineSeparator 設為 false，而傳送的對象是記錄導向的檔案。 2. 當使用者定義 meta 資料索引鍵 com.ibm.wmqfte.com.ibm.wmqfte.insertRecordLineSeparator 設為 false，而傳送來源是記錄導向的檔案。	true

相關資訊

第 464 頁的『表 45』

fteCreateTransfer -md 參數

其他 MFT 代理程式配置檔

除了 agent.properties 檔案之外，Managed File Transfer 代理程式在其配置目錄中還可以有一些 XML 配置檔。

配置檔

下列 XML 配置檔可用於指定代理程式使用的其他資訊：

ProtocolBridgeCredentials.xml

如果您的代理程式是通訊協定橋接器代理程式，您可以使用此檔案來指定認證，以用於登入代理程式所連接的 FTP 或 SFTP 伺服器。

ProtocolBridgeProperties.xml

如果您的代理程式是通訊協定橋接器代理程式，您可以使用此檔案來定義代理程式所連接的非預設通訊協定檔案伺服器。**fteCreateBridgeAgent** 指令會在此檔案中為您建立預設通訊協定檔案伺服器。

ConnectDirectCredentials.xml

如果您的代理程式是 Connect:Direct 橋接器代理程式，您可以使用此檔案來指定認證，以用於連接至傳送中所涉及的 Connect:Direct 節點。

ConnectDirectNodeProperties.xml

如果您的代理程式是 Connect:Direct 橋接器代理程式，您可以使用此檔案來指定有關傳送中所涉及 Connect:Direct 節點的作業系統資訊。

ConnectDirectProcessDefinition.xml

如果您的代理程式是 Connect:Direct 橋接器代理程式，您可以使用此檔案來指定使用者定義的 Connect:Direct 程序，以在檔案傳送時進行呼叫。

UserSandboxes.xml

您可以使用此檔案來指定代理程式可以從中或寫入至其中的檔案系統區域。

更新配置檔

與 `agent.properties` 檔案不同，您可以更新 XML 配置檔，並讓代理程式取得變更，而無需重新啟動代理程式。

在您提交傳送時，如果自前次代理程式檢查 XML 配置檔以來超過 10 秒鐘，代理程式會檢查前次修改 XML 配置檔的時間。如果自前次代理程式讀取 XML 配置檔以來已修改此檔案，代理程式會再次讀取此檔案。在與 XML 綱目比較後，如果此檔案的內容是有效的，代理程式會更新其資訊。如果檔案內容無效，代理程式會使用舊版檔案中的資訊，並將訊息寫入 `output0.log` 檔案。

通訊協定橋接器認證檔案格式

Managed File Transfer Agent 配置目錄中的 `ProtocolBridgeCredentials.xml` 檔案定義通訊協定橋接器代理程式用來向通訊協定伺服器授權其本身的使用者名稱及認證資訊。

`ProtocolBridgeCredentials.xml` 檔必須符合 `ProtocolBridgeCredentials.xsd` 綱目。`ProtocolBridgeCredentials.xsd` 綱目文件位於 MQMFT 安裝的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目錄中。使用者必須自行手動建立 `ProtocolBridgeCredentials.xml` 檔案，因為 **fteCreateBridgeAgent** 指令不會再建立此檔案。範例檔案位於 MQMFT 安裝的 `MQ_INSTALLATION_PATH/mqft/samples` 目錄中。

IBM WebSphere MQ 7.5 introduced a new `<agent>` element that contains the `<server>` or `<serverHost>` element for the named agent.

代理程式會定期重新載入 `ProtocolBridgeCredentials.xml` 檔案，因此對該檔案所做的任何有效變更，都會影響代理程式的行為。預設重新載入間隔是 30 秒。透過指定 `agent.properties` 檔案中的代理程式內容 `xmlConfigReloadInterval`，即可變更此間隔。

綱目 - 7.5 或更新版本

下列綱目說明 IBM MQ 8 的 `ProtocolBridgeCredentials.xml` 檔中哪些元素有效。

```
<schema targetNamespace="http://wmqfte.ibm.com/ProtocolBridgeCredentials" elementFormDefault="qualified"
  xmlns="https://www.w3.org/2001/XMLSchema" xmlns:tns="http://wmqfte.ibm.com/
ProtocolBridgeCredentials">
  <!--
    <?xml version="1.0" encoding="UTF-8"?>
    <tns:credentials xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeCredentials"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeCredentials
ProtocolBridgeCredentials.xsd ">
```

```

<tns:agent name="agent1">
  <tns:serverHost name="myserver">
    <tns:user name="fred" serverPassword="pwd" serverUserId="bill"/>
    <tns:user name="jane" serverUserId="june" hostKey="1F:2e:f3">
      <tns:privateKey associationName="test" keyPassword="pwd2">
        ... private key ...
      </tns:privateKey>
    </tns:user>
  </tns:serverHost>
</tns:agent>

<tns:agent name="agent2">
  <tns:server name="server*" pattern="wildcard">
    <tns:user name="fred" serverPassword="pwd" serverUserId="bill"/>
    <tns:user name="jane" serverUserId="june" hostKey="1F:2e:f3">
      <tns:privateKey associationName="test" keyPassword="pwd2">
        ... private key ...
      </tns:privateKey>
    </tns:user>
  </tns:server>
</tns:agent>

<tns:agent name="agent3">
  <tns:serverHost name="ftpsServer"
    keyStorePassword="keypass"
    trustStorePassword="trustpass">
    <tns:user name="fred" serverPassword="pwd" serverUserId="bill"/>
  </tns:serverHost>
</tns:agent>

-->
</tns:credentials>
-->
<element name="credentials" type="tns:credentialsType"/>

<complexType name="credentialsType">
  <sequence>
    <element name="agent" type="tns:agentType" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>

<complexType name="agentType">
  <choice minOccurs="0" maxOccurs="1">
    <element name="serverHost" type="tns:serverHostType" minOccurs="0" maxOccurs="unbounded"/>
    <element name="server" type="tns:serverType" minOccurs="0" maxOccurs="unbounded"/>
  </choice>
  <attribute name="name" type="string" use="required"/>
</complexType>

<complexType name="serverHostType">
  <sequence>
    <element ref="tns:user" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="name" type="string" use="required"/>
  <attribute name="keyStorePassword" type="string" use="optional"/>
  <attribute name="keyStorePasswordCipher" type="string" use="optional"/>
  <attribute name="trustStorePassword" type="string" use="optional"/>
  <attribute name="trustStorePasswordCipher" type="string" use="optional"/>
</complexType>

<complexType name="serverType">
  <sequence>
    <element ref="tns:user" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="name" type="string" use="required"/>
  <attribute name="pattern" type="tns:patternType" use="optional"/>
  <attribute name="keyStorePassword" type="string" use="optional"/>
  <attribute name="keyStorePasswordCipher" type="string" use="optional"/>
  <attribute name="trustStorePassword" type="string" use="optional"/>
  <attribute name="trustStorePasswordCipher" type="string" use="optional"/>
</complexType>

<element name="user" type="tns:userType"/>

<complexType name="userType">
  <sequence>
    <element ref="tns:privateKey" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="name" type="string" use="required"/>
  <attribute name="serverUserId" type="string" use="optional"/>
  <attribute name="serverUserIdCipher" type="string" use="optional"/>
  <attribute name="serverPassword" type="string" use="optional"/>

```

```

<attribute name="serverPasswordCipher" type="string" use="optional"/>
<attribute name="hostKey" 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>

```

瞭解 ProtocolBridgeCredentials.xml 檔

下列清單中說明 ProtocolBridgeCredentials.xml 檔中使用的元素及屬性。

<credentials>

群組元素，包含的元素說明供通訊協定橋接器代理程式用來連接至通訊協定伺服器的認證。

<agent>

包含指定代理程式之 <server> 或 <serverHost> 定義的元素。

<server>

通訊協定橋接器所連接的通訊協定伺服器。

The <server> element is not supported for 7.0.4 or earlier.

屬性	說明
名	通訊協定伺服器的名稱。
型樣	如果您使用萬用字元或正規表示式來指定通訊協定伺服器名稱的型樣，請使用 wildcard 或 regex 。
trustStorePassword 或 trustStorePasswordCipher	當 <server> 元素參照 FTPS 伺服器時需要。 用來存取信任儲存庫的密碼。如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。
keyStorePassword 或 keyStorePasswordCipher	選用項目。用來存取金鑰儲存庫的密碼。如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。

<serverHost>

通訊協定橋接器所連接的通訊協定伺服器的主機名稱。

The ProtocolBridgeCredentials.xml file can either contain <serverHost> elements or <server> elements but you cannot use a mixture of the two different types. 使用 <serverHost> 時，名稱符合通

訊協定伺服器的主機名稱。 When you use <server>, the name is matched against the protocol server's name (as defined in the ProtocolBridgeProperties.xml file).

屬性	說明
名	通訊協定伺服器的主機名稱或 IP 位址。
trustStorePassword 或 trustStorePasswordCipher	當 <serverHost> 元素參照 FTPS 伺服器時需要。用來存取信任儲存庫的密碼。如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。
keyStorePassword 或 keyStorePasswordCipher	選用項目。用來存取金鑰儲存庫的密碼。此為選用內容，除非設定 keyStore 屬性，在此情況下則為必要內容。如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。

<user>

從 Managed File Transfer 使用者名稱對映至通訊協定伺服器使用者名稱的使用者。

屬性	說明
名	用於 Managed File Transfer 的使用者名稱。
serverUserId 或 serverUserIdCipher	用於通訊協定伺服器的使用者名稱。如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。
serverPassword 或 serverPasswordCipher	在通訊協定伺服器上使用的使用者名稱的密碼。如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。
hostKey	伺服器主機 SSH 指紋。

<privateKey>

使用者的私密金鑰。

屬性	說明
keyPassword 或 keyStorePasswordCipher	私密金鑰的密碼。如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。
associationName	用於追蹤及記載的名稱。

通訊協定橋接器內容檔格式

代理程式配置目錄中的 ProtocolBridgeProperties.xml 檔案，會定義通訊協定檔案伺服器的內容。

ProtocolBridgeProperties.xml 檔必須符合 ProtocolBridgeProperties.xsd 綱目。

ProtocolBridgeProperties.xsd 綱目文件位於 Managed File Transfer 安裝的

MQ_INSTALLATION_PATH/mqft/samples/schema 目錄中。 **fteCreateBridgeAgent** 指令會在代理程式配置目錄中建立範本檔 ProtocolBridgeProperties.xml。

代理程式會定期重新載入 ProtocolBridgeProperties.xml 檔案，因此對該檔案所做的任何有效變更，都會影響代理程式的行為。預設重新載入間隔是 30 秒。透過指定 agent.properties 檔案中的代理程式內容 xmlConfigReloadInterval，即可變更此間隔。

綱目

下列綱目說明 ProtocolBridgeProperties.xml 檔。

註: IBM WebSphere MQ 7.5 或 IBM WebSphere MQ File Transfer Edition 7.0.2 或更新版本不支援 maxReconnectRetry 及 reconnectWaitPeriod 屬性。

```
<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>

<!--
and The type for matching valid server names. Server names must be at least 2 characters in length
    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>
```

瞭解 ProtocolBridgeProperties.xml 檔

下列清單中說明 ProtocolBridgeProperties.xml 檔中使用的元素及屬性：

<serverProperties>

XML 文件的根元素

<credentialsFile>

包含認證的檔案路徑。若為 IBM WebSphere MQ 7.5 或更新版本，此內容值可包含環境變數。如需相關資訊，請參閱第 452 頁的『在 MFT 內容中使用環境變數』

<defaultServer>

通訊協定檔案伺服器，作為檔案傳送的預設伺服器

<ftpServer>

FTP 檔案伺服器

<sftpServer>

SFTP 檔案伺服器

<ftpsServer>

FTPS 檔案伺服器

適用於所有類型通訊協定檔案伺服器的一般伺服器屬性：

屬性	說明
名	必要項目。通訊協定檔案伺服器名稱。通訊協定伺服器名稱的長度必須至少為兩個字元、不區分大小寫，且限於英數字元及下列字元： <ul style="list-style-type: none"> 句點 (.) 底線 (_) 正斜線 (/) 百分比符號 (%)
host	必要項目。您想要傳送或接收檔案的通訊協定檔案伺服器之主機名稱或 IP 位址。
port	選用項目。您想要傳送或接收檔案的通訊協定檔案伺服器之埠號。
platform	必要項目。您想要傳送或接收檔案的通訊協定檔案伺服器之平台。指定 UNIX 或 WINDOWS。請根據您在 FTP、FTPS 或 SFTP 伺服器上輸入路徑的方式，設定此內容。比方說，如果您在 Windows 上執行 FTP 伺服器，但在登入伺服器時必須輸入 UNIX 樣式的路徑（亦即，使用正斜線），請將此值設為 UNIX，而非 WINDOWS。在 Windows 上執行的伺服器，通常呈現 UNIX 樣式的檔案系統。
fileEncoding	必要項目。定義檔案伺服器所使用的字元編碼。以文字模式傳送檔案時會使用此內容，以於平台之間移動檔案時變更正確的編碼順序。例如 UTF-8。
limitedWrite	選用項目。寫入檔案伺服器時，預設模式是建立暫存檔，然後在傳送完成時重新命名該檔案。若為配置為唯寫的檔案伺服器，則會直接以其最終名稱建立檔案。此內容值可以是 true 或 false。預設值是 false。
controlEncoding	選用項目。傳送至通訊協定檔案服务器的控制訊息的控制編碼值。此內容會影響使用的檔名編碼，且必須與通訊協定檔案服务器的控制編碼相容。預設值是 UTF-8。

僅適用於 FTP 及 FTPS 伺服器的一般屬性：

屬性	說明
timeZone	必要項目。您想要傳送或接收檔案的通訊協定檔案伺服器之時區。例如：美洲/紐約或亞洲/東京。

屬性	說明
locale	必要項目。您想要傳送或接收檔案的通訊協定檔案伺服器所使用之語言。例如：en_US 或 ja_JP
listFormat	選用項目。清單格式，定義從通訊協定檔案伺服器傳回的檔案清單資訊的格式。使用 Windows 或 UNIX。預設值是 UNIX。
listFileRecentDateFormat	選用項目。FTP 伺服器上的 FTP 用戶端目錄清單的最近日期格式（一年內）。此屬性及 listFileOldDateFormat 屬性，可讓您重新定義通訊協定檔案伺服器傳回的預期日期格式。預設值由通訊協定檔案伺服器定義。
listFileOldDateFormat	選用項目。FTP 伺服器上的 FTP 用戶端目錄清單的舊日期格式（一年以上）。此屬性及 listFileRecentDateFormat 屬性，可讓您重新定義通訊協定檔案伺服器傳回的預期日期格式。預設值由通訊協定檔案伺服器定義。
monthShortNames	選用項目。用來解碼從通訊協定檔案伺服器所傳回的日期資訊的月份名稱取代清單。此內容由 12 個以逗點區隔的名稱清單組成，可置換預設語言環境月份值。預設值由通訊協定檔案伺服器定義。

僅適用於 FTP 伺服器的一般屬性：

屬性	說明
passiveMode	選用項目。控制 FTP 伺服器的連線為被動或主動。 如果將此內容值設為 false，連線為主動。如果值設為 true，則連線為被動。預設值是 false。

僅適用於 FTPS 伺服器的一般屬性：

屬性	說明
ftpsType	選用項目。指定使用 FTPS 通訊協定的明確或隱含格式。預設值是 explicit。
trustStore	必要項目。用來判斷是否信任 FTPS 伺服器所提供憑證的信任儲存庫位置。
trustStoreType	選用項目。信任儲存庫檔案的格式。預設值是 JKS。
keyStore	選用項目。在 FTPS 伺服器盤查時，用來提供憑證資訊的金鑰儲存庫位置。預設是讓通訊協定橋接器無法連接至已配置為需要用戶端鑑別的 FTPS 伺服器。
keyStoreType	選用項目。金鑰儲存庫檔案的格式。預設值是 JKS。
ccc	選用項目。選擇完成鑑別時是否使用清除（未加密）指令通道。預設值是 false，表示在整個 FTPS 階段作業期間，指令通道一直保持加密。只有在 ftpsType 設為 explicit 時，此屬性才適用。
protFirst	選用項目。指定是否在 PBSZ/PROT 指令之前或之後向 FTPS 伺服器發出 USER/PASS 指令。預設值是 false，表示先傳送 PBSZ/PROT 指令之後再發出 USER/PASS 指令。只有在 ftpsType 設為 explicit 時，此屬性才適用。
auth	選用項目。指定在 AUTH 指令中指定的通訊協定。首先會嘗試指定的通訊協定，然後依預設嘗試 TLS、SSL、TLS-C 或 TLS-P，直到 FTPS 伺服器不再以 504 回覆碼拒絕為止。只有在 ftpsType 設為 explicit 時，此屬性才適用。

<limits>

所有類型伺服器共同的屬性及其類型伺服器特定的屬性的儲存器元素：

適用於所有類型的通訊協定檔案伺服器的一般限制屬性：

屬性	說明
maxListFileNames	選用項目。在通訊協定檔案伺服器上的目錄中掃描檔名時收集的名稱數目上限。預設值是 999999999。
maxListDirectoryLevels	選用項目。通訊協定伺服器上遞迴掃描檔案名稱的目錄層次數目上限。預設值為 1000。
maxReconnectRetry (此屬性現已淘汰。)	已淘汰。 IBM WebSphere MQ 7.5 或 IBM WebSphere MQ File Transfer Edition 7.0.2 或更新版本不支援此屬性。 選用項目。通訊協定橋接器代理程式停止嘗試之前，通訊協定伺服器嘗試重新連接的次數上限。預設值是 2。
reconnectWaitPeriod (此屬性現已淘汰。)	已淘汰。 IBM WebSphere MQ 7.5 或 IBM WebSphere MQ File Transfer Edition 7.0.2 或更新版本不支援此屬性。 選用項目。嘗試重新連接之前等待的時間間隔（以秒為單位）。預設值是 10 秒。
maxSessions	選用項目。通訊協定伺服器的階段作業數目上限。此數必須大於或等於通訊協定橋接器代理程式的來源及目的地傳送次數上限的總和。預設值為代理程式內容 maxSourceTransfers、maxDestinationTransfers 及 maxCommandHandlerThreads 的值總和加上 1。如果這三個內容使用其預設值 25、25 及 5，則 maxSessions 預設值為 56。
socketTimeout	選用項目。Socket 逾時（以秒為單位）。檔案串流期間會使用此屬性的值。預設值是 30 秒。

僅適用於 **SFTP** 伺服器的限制屬性：

屬性	說明
connectionTimeout	選用項目。等待通訊協定檔案伺服器回應連線要求的時間（以秒為單位）。逾時表示通訊協定檔案伺服器無法使用。預設值是 30 秒。
cipherList	選用項目。指定用於在通訊協定橋接器代理程式與 SFTP 伺服器之間通訊的密碼清單（以逗點區隔）。將會根據此清單中指定的順序來呼叫密碼。此密碼在伺服器及用戶端上必須皆可用，然後才能使用它。 通訊協定橋接器代理程式支援的密碼如下所示： <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 依預設，通訊協定橋接器代理程式使用的密碼清單是 aes128-cbc , aes192-cbc , aes256-cbc。

Connect:Direct 認證檔格式

Managed File Transfer Agent 配置目錄中的 ConnectDirectCredentials.xml 檔案定義 Connect:Direct 代理程式用來向 Connect:Direct 節點進行自我授權的使用者名稱及認證資訊。

ConnectDirectCredentials.xml 檔案必須符合 ConnectDirectCredentials.xsd 綱目。ConnectDirectCredentials.xsd 綱目文件位於 MQMFT 安裝的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目錄中。範例 ConnectDirectCredentials.xml 檔案位於 MQMFT 安裝的 `MQ_INSTALLATION_PATH/mqft/samples/credentials` 目錄中。

代理程式會定期重新載入 ConnectDirectCredentials.xml 檔案，因此對該檔案所做的任何有效變更，都會影響代理程式的行為。預設重新載入間隔是 30 秒。透過在 `agent.properties` 檔案中指定代理程式內容 `xmlConfigReloadInterval`，可以變更此間隔。

綱目

下列綱目說明 ConnectDirectCredentials.xml 檔案中的有效元素。

```
<?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>

```

瞭解 ConnectDirectCredentials.xml 檔案

下列清單中說明了 ConnectDirectCredentials.xml 檔案中使用的元素及屬性。

<credentials>

包含說明認證的元素之群組元素，這些認證可供 Connect:Direct 橋接器代理程式，用來連接至 Connect:Direct 節點。

<agent>

包含指定代理程式之 <pnode> 定義元素的群組元素。

<pnode>

Connect:Direct 傳送中的主要節點 (PNODE)。此節點可起始與次要節點 (SNODE) 的連線。

屬性	說明
名	Connect:Direct 節點的名稱。此屬性的值可以是符合許多節點名稱的型樣。
型樣	指定用於 name 屬性值的型樣類型。pattern 屬性的有效值如下 <ul style="list-style-type: none"> wildcard - 使用萬用字元 regex - 使用 Java 正規表示式

<user>

提交傳送要求的 IBM MQ 使用者。

屬性	說明
名	用於 Managed File Transfer 的使用者名稱。此屬性的值可以是符合許多使用者名稱的型樣。
ignorecase	指定是否不區分名稱大小寫。ignorecase 屬性的有效值如下

屬性	說明
	<ul style="list-style-type: none"> • true - 名稱不區分大小寫 • false - 名稱區分大小寫
型樣	指定用於 name 屬性值的型樣類型。 pattern 屬性的有效值如下 <ul style="list-style-type: none"> • wildcard - 使用萬用字元 • regex - 使用 Java 正規表示式
cdUserId 或 cdUserIdCipher	Connect:Direct 橋接器用來連接其相關聯 Connect:Direct 節點的使用者名稱。 如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。
cdPassword 或 cdPasswordCipher	與 cdUserId 屬性所指定使用者名稱相關聯的密碼。 如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。
pnodeUserId 或 pnodeUserIdCipher	Connect:Direct 主要節點的使用者名稱。 如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。
pnodePassword 或 pnodePasswordCipher	與 pnodeUserId 屬性所指定使用者名稱相關聯的密碼。 如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。

<snode>

在 Connect:Direct 檔案傳送期間，執行次要節點 (SNODE) 角色的 Connect:Direct 節點。

屬性	說明
名	Connect:Direct 節點的名稱。 此屬性的值可以是符合許多節點名稱的型樣。
型樣	指定用於 name 屬性值的型樣類型。 pattern 屬性的有效值如下 <ul style="list-style-type: none"> • wildcard - 使用萬用字元 • regex - 使用 Java 正規表示式
userId 或 userIdCipher	在檔案傳送期間，用來連接至此節點的使用者名稱。 如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。
password 或 passwordCipher	與 userId 屬性所指定使用者名稱相關聯的密碼。 如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。

範例

在此範例中，Connect:Direct 橋接器代理程式會連接至 Connect:Direct 節點 pnode1。 如果 IBM MQ 使用者的使用者名稱是以 fteuser 字首開頭，而且後面接著單一字元（例如 fteuser2），當他要求涉及 Connect:Direct 橋接器的傳送時，Connect:Direct 橋接器代理程式即會使用 cduser 使用者名稱及 passwd0rd 密碼，連接至 Connect:Direct 節點 pnode1。 當 Connect:Direct 節點 pnode1 執行它所負責的傳送作業時，將會使用 pnodeuser 使用者名稱及 passwd0rd1 密碼。

如果 Connect:Direct 傳送中的次要節點，具有以 FISH 字首開頭的名稱，則節點 pnode1 會使用 fishuser 使用者名稱及 passwd0rd2 密碼，連接至該次要節點。 如果 Connect:Direct 傳送中的次要節點，具有以 CHIPS 字首開頭的名稱，則節點 pnode1 會使用 chipsuser 使用者名稱及 passwd0rd3 密碼，連接至該次要節點。

```
<?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="passwd0rd"
        pnodeUserId="pnodeuser" pnodePassword="passwd0rd1">
```

```

<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 節點內容檔格式

Connect:Direct 橋接器代理程式配置目錄中的 ConnectDirectNodeProperties.xml 檔案，會指定檔案傳送中所涉及的遠端 Connect:Direct 節點的相關資訊。

ConnectDirectNodeProperties.xml 檔案必須符合 ConnectDirectNodeProperties.xsd 綱目。ConnectDirectNodeProperties.xsd 綱目文件位於 MFT 安裝的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目錄中。**fteCreateCDAgent** 指令會在代理程式配置目錄中建立範本 ConnectDirectNodeProperties.xml 檔案。

代理程式會定期重新載入 ConnectDirectNodeProperties.xml 檔案，因此對該檔案所做的任何有效變更，都會影響代理程式的行為。預設重新載入間隔是 30 秒。透過指定 agent.properties 檔案中的代理程式內容 `xmlConfigReloadInterval`，即可變更此間隔。

綱目

下列綱目說明 ConnectDirectNodeProperties.xml 檔案中的有效元素。

```

<?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>

```

瞭解 ConnectDirectNodeProperties.xml 檔案

下列清單中說明了 ConnectDirectNodeProperties.xml 檔案中使用的元素及屬性。

nodeProperties


XML 文件的根元素。

credentialsFile

儲存機密性資訊之認證檔案的路徑。若為 IBM WebSphere MQ 7.5 或更新版本，此內容值可包含環境變數。如需相關資訊，請參閱第 452 頁的『[在 MFT 內容中使用環境變數](#)』

node

指定一個以上的 Connect:Direct 節點。

屬性	說明
名	用於識別 Connect:Direct 節點（使用 node 元素指定的定義）名稱的型樣。型樣相符功能不區分大小寫。
型樣	指定用於 name 屬性值的型樣類型。pattern 屬性的有效值如下： <ul style="list-style-type: none"> wildcard - 使用萬用字元 regex - 使用 Java 正規表示式 如需 MFT 使用的正規表示式類型的相關資訊，請參閱 第 554 頁的『MFT 所使用的正規表示式』 。
類型	指定符合 name 屬性所給定型樣的一或多個 Connect:Direct 節點的作業系統類型。type 屬性的有效值如下： <ul style="list-style-type: none"> Windows - 節點在 Windows 上執行 UNIX - 節點在 UNIX 或 Linux 上執行  z/OS、zos、os/390 或 os390 - 節點在 z/OS 上執行 此屬性的值不區分大小寫。

範例

在此範例中，Connect:Direct 認證檔名指定為 ConnectDirectCredentials.xml。此程式碼範例指定下列平台連線：

- 名稱開頭為 "cdnodew" 的所有 Connect:Direct 節點都是在 Windows 平台上執行。
- 名稱開頭為 "cdnodeu" 的所有 Connect:Direct 節點都是在 UNIX 平台上執行。
- 名稱開頭為 "cdnodez" 的所有 Connect:Direct 節點都是在 z/OS 平台上執行。
- 所有其他 Connect:Direct 節點是在 UNIX 平台上執行。

Connect:Direct 橋接器代理程式從檔案開頭到結尾搜尋相符項，並使用它找到的第一個相符項。

```
<?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 程序定義檔案格式

Connect:Direct 橋接器代理程式配置目錄中的 ConnectDirectProcessDefinitions.xml 檔案，會指定要在檔案傳送過程中啟動的使用者定義的 Connect:Direct 程序。

ConnectDirectProcessDefinitions.xml 檔案必須符合 ConnectDirectProcessDefinitions.xsd 綱目。ConnectDirectProcessDefinitions.xsd 綱目文件位於 MFT 安裝的 MQ_INSTALLATION_PATH/mqft/samples/schema 目錄中。

fteCreateCDAgent 指令會在代理程式配置目錄中建立範本 ConnectDirectProcessDefinitions.xml 檔案。

代理程式會定期重新載入 ConnectDirectProcessDefinitions.xml 檔案，因此對該檔案所做的任何有效變更，都會影響代理程式的行為。預設重新載入間隔是 30 秒。透過指定 agent.properties 檔案中的代理程式內容 xmlConfigReloadInterval，即可變更此間隔。

綱目

下列綱目說明 ConnectDirectProcessDefinitions.xml 檔案中的有效元素。

```
<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>
```

瞭解 ConnectDirectProcessDefinitions.xml 檔案

下列清單中說明了 ConnectDirectProcessDefinitions.xml 檔案中使用的元素及屬性。

cdProcess

XML 文件的根元素。

processSet

包含一組使用者定義的程序所有相關資訊的群組元素。

條件 (condition)

包含條件的群組元素，依據這些條件可測試傳送，以決定是否使用 processSet 元素中所含的一組程序。

match

測試變數的值是否符合給定值的條件。

屬性	說明
variable	指定變數。此變數的值會與 value 屬性的值相互比較。此變數為本質符號。如需相關資訊，請參閱第 554 頁的『與使用者定義的 Connect:Direct 程序搭配使用的替代變數』。
value	指定型樣，以符合 variable 屬性所指定之變數的值。
型樣	指定用於 value 屬性值的型樣類型。pattern 屬性的有效值為 <ul style="list-style-type: none">wildcard - 使用萬用字元regex - 使用 Java 正規表示式 此為選用屬性，預設值為 wildcard。

defined

測試是否已定義變數的條件。

屬性	說明
variable	指定變數。如果此變數存在，則表示滿足符合條件。此變數為本質符號。如需相關資訊，請參閱第 554 頁的『與使用者定義的 Connect:Direct 程序搭配使用的替代變數』。

處理程序

包含位置相關資訊的群組元素，在找到相符項時，將在此位置尋找要呼叫的 Connect:Direct 程序。

transfer

在執行傳送要求期間要呼叫的 Connect:Direct 程序。

屬性	說明
處理程序	選用項目。指定檔案名稱，該檔案包含在執行傳送要求期間要呼叫的 Connect:Direct 程序。此檔案路徑相對於 Connect:Direct 橋接器代理程式配置目錄。此為選用屬性，預設值是使用 MFT 產生的程序。若為 IBM WebSphere MQ 7.5 或更新版本，此內容值可包含環境變數。如需相關資訊，請參閱第 452 頁的『在 MFT 內容中使用環境變數』。

範例

在此範例中，有三個 processSet 元素。

第一個 processSet 元素指定如果傳送要求具有 **%FTESNODE** 變數 (其值符合型樣 Client*) 及 **%FTESUSER** 變數 (其值為 Admin)，則 Connect:Direct 橋接器代理程式會在傳送過程中提交位於 *agent_configuration_directory/AdminClient.cdp* 中的 Connect:Direct 程序。

第二個 processSet 元素指定如果傳送要求具有 **%FTESNODE** 變數，且其值符合型樣 Client*，則 Connect:Direct 橋接器代理程式會提交位於 *agent_configuration_directory/Client.cdp* 中的 Connect:Direct 程序，作為傳送的一部分。Connect:Direct 橋接器代理程式會依定義的順序讀取 processSet 元素，如果找到相符項，則會使用第一個相符項，而不會尋找另一個相符項。對於同時符合第一個和第二個 processSet 的條件的傳送要求，Connect:Direct 橋接器代理程式只會呼叫第一個 processSet 指定的處理程序。

第三個 processSet 元素沒有條件，且符合所有傳送。如果傳送要求不符合第一個或第二個 processSet 的條件，則 Connect:Direct 橋接器代理程式會提交由第三個條件指定的 Connect:Direct 處理程序。此程序位於 *agent_configuration_directory/Default.cdp* 中，將在傳送過程中提交。

```

<?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>

```

使用 MFT 使用者沙盤推演

您可以根據要求傳送的 MQMD 使用者名稱，限制可來回傳送檔案的檔案系統區域。

當代理程式是通訊協定橋接器代理程式或 Connect:Direct 橋接器代理程式時，不支援使用者沙盤推演。

若要啟用使用者沙盤推演，請針對您想要限制的代理程式，將下列內容新增至 `agent.properties` 檔案中：

```
userSandboxes=true
```

有了這項內容並設為 `true` 時，代理程式就會使用 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name/UserSandboxes.xml` 檔案中的資訊，來決定要求傳送的使用者可以存取檔案系統的哪些部分。

`UserSandboxes.xml` XML 由包含零個以上 `<sandbox>` 元素的 `<agent>` 元素組成。這些元素說明哪些規則適用於哪些使用者。`<sandbox>` 元素的 `user` 屬性是一種型樣，用來比對要求的 MQMD 使用者。

代理程式會定期重新載入 `UserSandboxes.xml` 檔案，因此對該檔案所做的任何有效變更，都會影響代理程式的行為。預設重新載入間隔是 30 秒。透過指定 `agent.properties` 檔案中的代理程式內容 `xmlConfigReloadInterval`，即可變更此間隔。

如果指定 `userPattern="regex"` 屬性或值，則 `user` 屬性會解譯為 Java 正規表示式。如需相關資訊，請參閱第 554 頁的『MFT 所使用的正規表示式』。

如果您未指定 `userPattern="regex"` 屬性或值，則 `user` 屬性會解譯為具有下列萬用字元的型樣：

- 星號 (*)，代表零個以上字元
- 問號 (?)，正好代表一個字元

會依照該檔案中列出 `<sandbox>` 元素的順序來執行比對。只會使用第一個相符項，而忽略該檔案中所有後續可能的相符項。如果該檔案中指定的 `<sandbox>` 元素全部都不符合與傳送要求訊息相關聯的 MQMD 使用者，則傳送時將無法存取檔案系統。在 MQMD 使用者名稱與 `user` 屬性之間找到相符項之後，此相符項將識別 `<sandbox>` 元素內套用至傳送的一組規則。這組規則用來決定傳送過程中可以讀取或寫入的檔案或資料集。

每一組規則可指定 <read> 元素來識別可讀取的檔案，以及指定 <write> 元素來識別可寫入的檔案。如果您在一組規則中省略了 <read> 或 <write> 元素，即假設與該組規則相關聯的使用者不得執行任何讀取或寫入。

註：在 UserSandboxes.xml 檔中，<read> 元素必須在 <write> 元素之前，<include> 元素必須在 <exclude> 元素之前。

每一個 <read> 或 <write> 元素包含一個以上型樣，用來決定沙盤推演中是否存在檔案以及是否可以進行傳送。請使用 <include> 及 <exclude> 元素來指定這些型樣。<include> 或 <exclude> 元素的 name 屬性指定要比對的型樣。選用的 type 屬性指定名稱值是檔案或佇列型樣。如果未指定 type 屬性，代理程式會將此型樣視為檔案或目錄路徑型樣。例如：

```
<tns:read>
  <tns:include name="/home/user/**"/>
  <tns:include name="USER.**" type="queue"/>
  <tns:exclude name="/home/user/private/**"/>
</tns:read>
```

代理程式使用 <include> 及 <exclude> name 型樣，來決定是否可以讀取或寫入檔案、資料集或佇列。如果標準檔案路徑、資料集或佇列名稱符合至少其中一個包括的型樣，而排除的型樣一個都不符合，則容許作業。使用 <include> 及 <exclude> 元素的 name 屬性指定的型樣，會使用適用於代理程式執行所在平台的路徑分隔字元及使用慣例。如果您指定相對檔案路徑，則解析的路徑將相對於代理程式的 transferRoot 內容。

指定佇列限制時，支援 QUEUE@QUEUEMANAGER 的語法，且規則如下：

- 如果項目中遺漏 at 字元 (@)，則將該型樣視為可以在任何佇列管理程式上存取的佇列名稱。比方說，例如型樣是 name，則將它視為與 name@** 相同。
- 如果 at 字元 (@) 是項目中的第一個字元，則將該型樣視為佇列管理程式名稱，並可存取該佇列管理程式上的所有佇列。比方說，例如型樣是 @name，則將它視為與 **@name 相同。

當您指定下列萬用字元作為 <include> 及 <exclude> 元素的 name 屬性的一部分時，它們具有特殊意義：


在目錄名稱中，或在資料集名稱或佇列名稱的限定元中，單一星號符合零或多個字元。

?

問號符合目錄名稱中的正好一個字元，或符合資料集名稱或佇列名稱的限定元。

兩個星號字元符合零或多個目錄名稱，或資料集名稱或佇列名稱中零或多個限定元。同時，結尾是路徑分隔字元的路徑有隱含的 "***" 新增至路徑結尾。因此，/home/user/ 與 /home/user/** 相同。

例如：

- /**/test/** 符合其路徑中有 test 目錄的任何檔案
- /test/file? 符合 /test 目錄內以字串 file 開頭且後面接著任何單一字元的任何檔案
- c:\test*.txt 符合 c:\test 目錄內具有 .txt 副檔名的任何檔案
- c:\test***.txt 符合 'c:\test 目錄內任何檔案，或其中一個子目錄具有 .txt 副檔名
-  z/OS // 'TEST.*.DATA' 符合其第一個限定元為 TEST、有任何第二個限定元，且第三個限定元為 DATA 的任何資料集。
- *@QM1 符合佇列管理程式 QM1 中具有單一限定元的任何佇列。
- TEST.*.QUEUE@QM1 符合佇列管理程式 QM1 上其第一個限定元為 TEST、有任何第二個限定元，且第三個限定元為 QUEUE 的任何佇列。
- **@QM1 符合佇列管理程式 QM1 上的任何佇列。

符號鏈結

您必須完整解析 UserSandboxes.xml 檔案中檔案路徑所使用的任何符號鏈結，方法是在 <include> 及 <exclude> 元素中指定固定鏈結。例如，如果您具有 /var 對映至 /SYSTEM/var 的符號鏈結，則必須將此路徑指定為 <tns:include name="/SYSTEM/var"/>，否則預期的傳送會因使用者沙盤推演安全錯誤而失敗。

範例

此範例顯示如何容許具有 MQMD 使用者名稱 guest 的使用者，透過將下列 <sandbox> 元素新增至 AGENT_JUPITER 配置目錄中的 UserSandboxes.xml 檔案，從代理程式 AGENT_JUPITER 執行所在系統上的 /home/user/public 目錄或其任何子目錄中傳送任何檔案：

```
<?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>
```

範例

此範例說明如何容許具有 MQMD 使用者名稱 account 且後面跟著單一數字的任何使用者（例如 account4）完成下列動作：

- 傳送來自 /home/account 目錄或其任何子目錄的任何檔案（不包括代理程式 AGENT_SATURN 執行所在系統上的 /home/account/private 目錄）
- 將任何檔案傳送至代理程式 AGENT_SATURN 執行所在系統上的 /home/account/output 目錄或其任何子目錄
- 從本端佇列管理程式上以字首 ACCOUNT. 開頭的佇列中讀取訊息，除非它以 ACCOUNT.PRIVATE. 開頭（即在第二層具有 PRIVATE）。
- 將資料傳送至任何佇列管理程式上開頭為字首 ACCOUNT.OUTPUT. 的佇列上。

若要容許具有 MQMD 使用者名稱 account 的使用者完成這些動作，請將下列 <sandbox> 元素新增至 AGENT_SATURN 配置目錄中的 UserSandboxes.xml 檔案：

```
<?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>
```

相關參考

[第 43 頁的『對萬用字元傳送進行其他檢查』](#)

從 Continuous Delivery 中的 IBM MQ 9.0.1，以及 Long Term Support 中的 IBM MQ 9.0.0 Fix Pack 1，如果已使用使用者或代理程式沙盤推演來配置代理程式，以限制代理程式可以來回傳送檔案的位置，您可以指定對該代理程式的萬用字元傳送進行其他檢查。

第 40 頁的『使用 MFT 使用者沙盤推演』

您可以根據要求傳送的 MQMD 使用者名稱，限制可來回傳送檔案的檔案系統區域。

第 463 頁的『MFT agent.properties 檔案』

每一個 Managed File Transfer Agent 都有自己的內容檔 agent.properties，其中必須包含代理程式用來連接至其佇列管理程式的資訊。agent.properties 檔案也可以包含變更代理程式行為的內容。

MFT 日誌程式配置內容

Managed File Transfer 日誌程式有一組配置內容。您可以在 logger.properties 檔案中指定這些內容，該檔案位於 MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name 目錄中。

若為 IBM WebSphere MQ 7.5 版或更新版本，可在某些表示檔案或目錄位置的 Managed File Transfer 內容中使用環境變數。這可讓執行產品的組件時所使用的檔案或目錄，隨著環境變更（例如執行程序的使用者為何）而改變其所在位置。如需相關資訊，請參閱第 452 頁的『在 MFT 內容中使用環境變數』。

註：在 Windows 上指定檔案路徑時，反斜線 (\) 分隔字元必須呈現為兩條反斜線 (\\)（亦即，跳出 \）。或者，您也可以使用單一正斜線字元 (/) 作為分隔字元。如需 Oracle 中 Java 內容檔字元跳出方式的相關資訊，請參閱 Javadoc 中的 Properties 類別。

連結模式連線內容

內容名稱	說明	預設值
wmqfte.logger.type	使用中的日誌程式類型：檔案或資料庫。請將此值設為 FILE 或 DATABASE。	無預設值
wmqfte.max.transaction.messages	在確定交易之前，於交易中處理的訊息數目上限。在循環式記載模式中，佇列管理程式有固定的空間量可供進行中資料使用。請確保將此內容設定為足夠低的值，以避免耗盡可用空間。	50
wmqfte.max.transaction.time	在兩次交易確定之間經歷的時間長度上限（毫秒）。	5000
wmqfte.max.consecutive.reject	可連續拒絕的訊息數目（亦即，未發現有效訊息）上限。 若超出此數目，則日誌程式推斷問題與訊息本身無關，而與配置有關。例如，若您在資料庫中設定的代理程式名稱直欄不足以容納所有的代理程式名稱，則所有參照代理程式的訊息都會被拒絕。	50
wmqfte.reject.queue.name	日誌程式放置日誌程式無法處理之訊息的佇列名稱。若您有資料庫日誌程式，如需哪些訊息可能會放置到此佇列的詳細資料，請參閱資料庫日誌程式錯誤處理及拒絕。	SYSTEM.FTE.LOG.RJCT.logger_name
wmqfte.command.queue.name	佇列的名稱，日誌程式從中讀取控制其行為的指令訊息。	SYSTEM.FTE.LOG.CMD.logger_name
wmqfte.queue.manager	日誌程式所連接的佇列管理程式。這是必要參數，而且是佇列管理程式的連結模式連線所需要的全部。（如需連接遠端佇列管理程式的內容，請參閱第 507 頁的表 48。）	無預設值

表 47: 連結模式的連線內容 (繼續)		
內容名稱	說明	預設值
wmqfte.message.source.type	<p>下列其中一個值:</p> <p>automatic subscription 預設值。日誌程式會在 SYSTEM.FTE/Log/# 中定義的佇列管理程式上, 建立及使用其專屬的可延續受管理訂閱。此值適用於大多數實務。</p> <p>administrative subscription 如果自動訂閱不適用, 您可以定義不同的訂閱 (例如, 透過使用「IBM MQ Explorer」、MQSC 或 PCF), 並指示日誌程式使用該訂閱。例如, 使用此值分割日誌空間, 以便其中一個日誌程式處理 A-H 的代理程式, 另一個日誌程式處理 I-P 的代理程式, 而第三個日誌程式處理 Q-Z 的代理程式。</p> <p>queue 如果 IBM MQ 拓撲表示不方便為日誌程式建立訂閱, 您可以改用佇列。配置 IBM MQ, 讓佇列接收通常由 SYSTEM.FTE/Log/#。</p>	automatic subscription
wmqfte.message.source.name	<p>如果訊息來源類型為 administrative subscription 或 queue, 則使用訂閱或佇列名稱。如果來源類型為 automatic subscription, 則會忽略此內容。</p>	無預設值
wmqfte.database.credentials.file	<p>包含用來連接資料庫的使用者名稱和密碼的檔案。若為 IBM WebSphere MQ 7.5 或更新版本, 此內容值可包含環境變數。</p> <p>如需相關資訊, 請參閱第 696 頁的『MFT 認證檔格式』。</p>	<p> 如需建立「鑑別認證檔」的詳細資料, 請參閱 建立 MFT 認證檔</p> <p> 如需此檔案的位置及許可權的相關資訊, 請參閱 配置 MQMFTCredentials.xml。</p> <p> 關於建立「鑑別認證檔」的進一步詳細資料位於 MFT 和 IBM MQ 連線鑑別 中</p>
wmqfte.database.driver	<p>資料庫的 JDBC 驅動程式類別位置。這通常是 JAR 檔的路徑及檔名。例如, AIX 系統上 Db2 的第 2 類驅動程式需要 /opt/IBM/db2/V9.5/java/db2jcc.jar 檔案。在 Windows 系統上, 將路徑分隔字元指定為正斜線字元 (/), 例如 C:/Program Files/IBM/SQLLIB/java/db2jcc.jar。</p> <p> 在 z/OS 上, 指定 db2jcc.jar 檔案的完整路徑。例如, wmqfte.database.driver=/db2/db2v10/jdbc/classes/db2jcc.jar。</p> <p> 在 z/OS 系統上, 您必須參照下列所有 JAR 檔:</p> <ul style="list-style-type: none"> • db2jcc.jar • db2jcc_license_cisuz.jar • db2jcc_javax.jar <p>如果資料庫驅動程式包含多個 JAR 檔 (例如, Db2 9.1 需要驅動程式 JAR 檔及授權 JAR 檔), 請在此內容中包含所有這些 JAR 檔。請使用平台適用的類別路徑分隔字元來區隔多個檔名, 亦即: 在 Windows 系統上使用分號字元 (;), 在其他平台上則使用冒號字元 (:).</p>	無預設值

表 47: 連結模式的連線內容 (繼續)		
內容名稱	說明	預設值
wmqfte.database.exclude_duplicate_metadata	<p>控制是否將項目儲存在 meta 資料表格中，該表格所含資訊可在資料庫日誌程式綱目內的其他表格中找到。請將此值設定為 true 或 false。依預設不會再儲存這些 meta 資料項目，因為這會與現有資料重複，從而浪費資料庫儲存區容量。出現相同資料的內容項目及表格如下：</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>將此內容值設定為 false，會導致將這些 meta 資料項目儲存在 meta 資料表格中。</p>	true
wmqfte.database.host	<p>僅限 Db2:</p> <p>若為 IBM WebSphere MQ 7.5 版或更新版本，則是要使用第 4 類 JDBC 驅動程式連接的資料庫伺服器的主機名稱。如果指定了此內容的值，則必須同時指定 wmqfte.database.port 的值。如果兩個內容皆未定義，則資料庫日誌程式會使用預設 Type 2 JDBC 驅動程式進行連接。</p> <p>如果指定了此內容的值，則此日誌程式的認證檔案（由 wmqfte.database.credentials.file 內容定義檔案路徑）必須存在且可存取，以定義連接至資料庫的使用者名稱及密碼，即使資料庫位於本端系統也一樣。</p>	無預設值
wmqfte.database.name	包含 Managed File Transfer 日誌表格的資料庫實例（使用 Db2 for z/OS 時則為子系統）名稱。	無預設值
wmqfte.database.type	使用中的資料庫管理系統：Db2 或 Oracle。將此值設定為 db2 或 oracle。	db2
wmqfte.database.port	<p>僅限 Db2:</p> <p>若為 IBM WebSphere MQ 7.5 版或更新版本，則是要使用第 4 類 JDBC 驅動程式連接的資料庫伺服器的埠號。如果指定了此內容的值，則必須同時指定 wmqfte.database.host 的值。如果兩個內容皆未定義，則資料庫日誌程式會使用預設 Type 2 JDBC 驅動程式進行連接。</p> <p>如果指定了此內容的值，則此日誌程式的認證檔案（由 wmqfte.database.credentials.file 內容定義檔案路徑）必須存在且可存取，以定義連接至資料庫的使用者名稱及密碼，即使資料庫位於本端系統也一樣。</p>	無預設值
wmqfte.database.schema	包含 Managed File Transfer 日誌表格的資料庫綱目。在大部分情況下，預設值即已適用，但可能需要根據自己網站特定的資料庫考量指定其他替代值。	FTELOG

表 47: 連結模式的連線內容 (繼續)		
內容名稱	說明	預設值
wmqfte.database.native.library.path	<p>包含您所選資料庫驅動程式所需原生程式庫的路徑 (如果有的話)。例如, AIX 系統上 Db2 的第 2 類驅動程式需要來自 /opt/IBM/db2/V9.5/lib32/的程式庫。作為此內容的替代方案, 您可以使用其他方法設定 java.library.path 系統內容。</p> <p>在 Solaris 和 HP-UX 系統上, 在執行 fteStartLogger 指令之前, 您還必須設定及匯出 LD_LIBRARY_PATH 環境變數, 以包含此路徑。</p>	無預設值
wmqfte.file.logger.fileDirectory	檔案日誌程式日誌檔所處的目錄。	mqft/logs/coordination_dir/ loggers/logger_name/logs
wmqfte.file.logger.fileSize	<p>容許日誌檔增長到的大小上限。此大小值為大於零的正整數, 後面跟著下列其中一個單位: KB、MB、GB、m (分鐘)、h (小時)、d (日)、w (週)。</p> <p>例如, <code>wmqfte.file.logger.fileSize=5MB</code> 指定檔案大小上限為 5MB。 <code>wmqfte.file.logger.fileSize=2d</code> 指定檔案大小上限為 2 天資料。</p>	10MB
wmqfte.file.logger.fileCount	要建立的日誌檔數上限。當資料量超出可儲存在此數目檔案中的數量上限時, 即會刪除最舊的檔案, 從而使檔案數永不超出指定的值。	3

表 47: 連結模式的連線內容 (繼續)

內容名稱	說明	預設值
wmqfte.file.logger.mode	<p>使用中的日誌程式模式：循環或線性。請將此值設為 CIRCULAR 或 LINEAR。</p> <p>CIRCULAR - 檔案日誌程式會將資訊寫入某個檔案中，直到該檔案達到使用 wmqfte.file.logger.fileSize 內容定義的大小上限為止。達到大小上限時，檔案日誌程式即會啟動新檔案。以此模式寫入的檔案數上限，受制於使用 wmqfte.file.logger.fileCount 內容定義的值。達到此檔案數上限時，檔案日誌程式即會刪除第一個檔案，然後重建該檔案以用作目前作用中的檔案。如果 wmqfte.file.logger.fileSize 內容中定義的值是固定大小位元組單位（例如 KB、MB 或 GB），則此模式下使用的磁碟空間上限將等於 fileSize 乘以 fileCount。如果 wmqfte.file.logger.fileSize 內容中定義的值是時間單位（例如 m、h、d 或 w），則大小上限將取決於系統中這些時段內的日誌訊息傳輸量。以此模式執行時所使用的日誌檔命名慣例是： <i>Logger_namenumber-timestamp.log</i>，其中：</p> <ul style="list-style-type: none"> • <i>logger_name</i> 是在 fteCreateLogger 指令中指定給日誌程式的名稱。 • <i>number</i> 是檔案集內的檔案數目。 • <i>timestamp</i> 是檔案建立時的時間戳記。 <p>例如： LOGGER1-20111216123430147.log</p> <p>LINEAR - 檔案日誌程式會將資訊寫入某個檔案中，直到該檔案達到使用 wmqfte.file.logger.fileSize 內容定義的大小上限為止。達到大小上限時，檔案日誌程式即會啟動新檔案。先前寫入的檔案不會刪除，容許保存作為日誌訊息的歷程記錄。以線性模式執行時不會刪除檔案，因此會忽略 wmqfte.file.logger.fileCount 內容，因為可建立的檔案數並沒有上限。由於以此模式執行時並沒有上限，因此必須追蹤日誌檔所使用的磁碟空間量，以避免磁碟空間不足。以此模式執行時所使用的日誌檔命名慣例是： <i>logger_name-timestamp.log</i>，其中：</p> <ul style="list-style-type: none"> • <i>logger_name</i> 是在 fteCreateLogger 指令中指定給日誌程式的名稱。 • <i>timestamp</i> 是檔案建立時的時間戳記。 <p>例如： LOGGER-20111216123430147.log</p>	無預設值
wmqfte.max.retry.interval	<p>日誌程式遇到持續性錯誤時重試間隔的時間上限（以秒為單位）。</p> <p>部分錯誤狀況（如資料庫連線中斷）會使日誌程式無法繼續執行。發生這種情況時，日誌程式會先回復現行交易，然後在一段時間後重試。日誌程式起初等待的時間非常短，以便臨時錯誤能夠得到快速解決。但是，每次日誌程式重試時，其等待時間都會增加。這可避免在錯誤狀況持續較長時間時（如關閉資料庫以進行維護時）執行太多不必要的工作。</p> <p>使用此內容可設定等待時間長度的限制，以便解決錯誤狀況時在合理的時間內進行重試。</p>	600
loggerQMgrRetryInterval	日誌程式的處理程序控制器檢查佇列管理程式可用性的間隔（以秒為單位）。	30
maxRestartCount	在 maxRestartInterval 內容值所指定的時間間隔內，可進行的重新啟動次數上限。超出此值時，日誌程式的處理程序控制器會停止重新啟動日誌程式，並改為根據 maxRestartDelay 內容的值來執行動作。	4

表 47: 連結模式的連線內容 (繼續)		
內容名稱	說明	預設值
maxRestartInterval	日誌程式的處理程序控制器測量日誌程式重新啟動的間隔 (以秒為單位)。若在此間隔內重新啟動次數超出 maxRestartCount 內容的值, 日誌程式的程序控制器會停止重新啟動日誌程式。日誌程式的處理程序控制器會改為根據 maxRestartDelay 內容的值來執行動作。	120
maxRestartDelay	決定當日誌程式重新啟動率超出 maxRestartCount 和 maxRestartInterval 內容的值時, 日誌程式的處理程序控制器的行為。若您指定的值小於或等於零, 日誌程式的處理程序控制器會停止。若您指定的值大於零, 則必須先等待此秒數之後, 再重設日誌程式的處理程序控制器所保留的重新啟動歷程資訊, 以及重新啟動日誌程式。	-1
wmqfte.oracle.port	日誌程式用來連接至 Oracle 實例的埠。此埠也稱為 TNS 接聽器。	1521
wmqfte.oracle.host	日誌程式用來連接至 Oracle 實例的主機。	localhost
armELEMTYPE	選用內容。如果日誌程式配置為由 Automatic Restart Manager (ARM) 重新啟動, 則將此內容設為相關聯 ARM 原則中指定的 ARM ELEMTYPE 參數值。對於日誌程式, 請將 ELEMTYPE 設為 SYSBFLG。	未設定
armELEMENT	選用內容。如果日誌程式配置為由 Automatic Restart Manager (ARM) 重新啟動, 則將此內容設為相關聯 ARM 原則中指定的 ARM ELEMENT 參數值。您可以將 ELEMENT 值設為對應於日誌程式名稱。	未設定
loggerQMgrAuthenticationCredentialsFile	包含 MQ 連線認證 (用於日誌程式的協調佇列管理程式連線) 的檔案路徑。	 如需建立「鑑別認證檔」的詳細資料, 請參閱 建立 MFT 認證檔  如需此檔案的位置及許可權的相關資訊, 請參閱 配置 MQMFTCredentials.xml 。  關於建立「鑑別認證檔」的進一步詳細資料位於 MFT 和 IBM MQ 連線鑑別 中
trace	選用內容。日誌程式啟動時, 在啟用追蹤的情況下執行日誌程式的追蹤指定。追蹤指定是以逗點區隔的類別清單、等於字元及追蹤層次。 例如, com.ibm.wmqfte.databaselogger、com.ibm.wmqfte.databaselogger.operation=all。 您可以使用以冒號區隔的清單來指定多個追蹤指定。 例如: com.ibm.wmqfte.databaselogger=moderate: com.ibm.wmqfte.databaselogger.operation=all	無
traceFiles	選用內容。要保留的追蹤檔總數。此值會套用於日誌程式的程序控制器及其本身。	5
traceSize	選用內容。每一個追蹤檔在追蹤覆蓋至下一個檔案之前的大小上限 (以 MB 為單位)。此值會套用於日誌程式的程序控制器及其本身。	20

用戶端模式連線內容

V 9.0.4

下表顯示支援日誌程式佇列管理程式的用戶端模式連線所需的其他日誌程式內容。

表 48: 用戶端模式的連線內容		
內容名稱	說明	預設值
wmqfte.queue.manager.host	日誌程式佇列管理程式的主機名稱或 IP 位址。	無預設值
wmqfte.queue.manager.port	日誌程式佇列管理程式正在其中進行接聽的埠。	1414
wmqfte.queue.manager.channel	日誌程式佇列管理程式中伺服器連線通道的名稱。	SYSTEM.DEF.SVRCONN
wmqfte.Ssl.CipherSuite	指定日誌程式與日誌程式佇列管理程式如何交換資料的各個 TLS 層面。 wmqfte.Ssl.CipherSuite 的值是 CipherSuite 名稱。此 CipherSuite 名稱對映至日誌程式佇列管理程式通道上使用的 CipherSpec 名稱。 如需相關資訊，請參閱 CipherSuite 及 CipherSpec 名稱對映。	無預設值
wmqfte.Ssl.PeerName	指定必須符合日誌程式佇列管理程式所提供名稱的識別名稱架構。識別名稱用來檢查佇列管理程式在連線時提供的識別憑證。	無預設值
wmqfte.Ssl.TrustStore	指定日誌程式信任的憑證位置。 wmqfte.Ssl.TrustStore 的值是檔案路徑。如果此檔案路徑是 Windows 檔案路徑，則必須將反斜線字元 (\) 跳出為具有進一步的反斜線字元 (\\)。 請注意，此內容值可包含環境變數。	無預設值
wmqfte.Ssl.TrustStoreCredentialsFile	包含 wmqfte.Ssl.TrustStore 認證的檔案路徑。 請注意，此內容值可包含環境變數。	無預設值
wmqfte.Ssl.TrustStoreType	您想要使用的 SSL 金鑰儲存庫類型。支援 JKS 及 PKCS#12 金鑰儲存庫。此內容值可以是 jks 或 pkcs12。	jks
wmqfte.Ssl.KeyStore	指定日誌程式的私密金鑰位置。 wmqfte.Ssl.KeyStore 的值是檔案路徑。如果此檔案路徑是 Windows 檔案路徑，則必須將反斜線字元 (\) 跳出為具有進一步的反斜線字元 (\\)。 請注意，此內容值可包含環境變數。	無預設值
wmqfte.Ssl.KeyStore.CredentialsFile	包含 wmqfte.Ssl.KeyStore 認證的檔案路徑。 請注意，此內容值可包含環境變數。	無預設值
wmqfte.Ssl.KeyStoreType	您想要使用的 SSL 金鑰儲存庫類型。支援 JKS 及 PKCS#12 金鑰儲存庫。此內容值可以是 jks 或 pkcs12。	jks
wmqfte.Ssl.FipsRequired	指定您想要在日誌程式層次上啟用 FIPS 支援。此內容值可以是 true 或 false。如需相關資訊，請參閱第 560 頁的『MFT 中的 FIPS 支援』。	false

相關參考

第 452 頁的『在 MFT 內容中使用環境變數』

從 IBM WebSphere MQ 7.5 開始，可能會在代表檔案或目錄位置的 Managed File Transfer 內容中使用環境變數。這可讓執行產品的某部分時所使用的檔案或目錄，隨著環境變更而改變其所在位置。例如，執行程序的使用者為何。

第 508 頁的『MFT 的 SSL 內容』

對 IBM MQ 及 Managed File Transfer 使用 SSL 或 TLS，可避免代理程式與佇列管理程式之間未獲授權的連線，以及加密代理程式與佇列管理程式之間的訊息資料流量。




MFT 的 Java 系統內容

有一些 Managed File Transfer 指令及代理程式內容必須定義為 Java 系統內容，因為它們為無法使用指令或代理程式內容機制的早期功能定義配置。

定義環境變數 BFG_JVM_PROPERTIES，以對要執行 Managed File Transfer 指令的 JVM 定義系統內容及其他 JVM 選項。例如，若要在 UNIX 類型平台上設定 com.ibm.wmqfte.maxConsoleLineLength 內容，請如下所示定義變數：

```
export BFG_JVM_PROPERTIES="-Dcom.ibm.wmqfte.maxConsoleLineLength=132"
```

如果您將代理程式作為 Windows 服務執行，則可以透過在 **fteModifyAgent** 指令上指定 **-sj** 參數來修改代理程式的 Java 系統內容。

內容名稱	說明	值
com.ibm.wmqfte.maxConsoleLineLength	可寫入主控台的字行長度上限。超過此長度的字行會自動換行。此值以位元組數表示（而非字元）。	 IBM i 的預設長度為 132 個位元組。   若為 UNIX、Linux、Windows 及 z/OS，則此長度無限制。
com.ibm.wmqfte.daemon.windows.windowsServiceLogFilesm	（僅限 Windows。）指定要保留的 Windows 服務日誌檔數目上限。Windows 服務日誌檔會建立在代理程式及資料庫日誌程式日誌目錄中（如果這些應用程式是以 Windows 服務方式執行）。Windows 服務日誌檔以字首 <i>service</i> 命名，它包含啟動及停止服務的相關訊息。	5

相關概念

第 49 頁的『多平台上的 MFT 配置選項』

Managed File Transfer 提供了一組內容檔，這些檔案包含有關您的設定的金鑰資訊，是作業所必要的。這些內容檔位於安裝產品時所定義的配置目錄中。


第 229 頁的『使用 MFT 的提示及要訣』

這裡提供了一些建議，可協助您充分使用 Managed File Transfer。

MFT 的 SSL 內容

對 IBM MQ 及 Managed File Transfer 使用 SSL 或 TLS，可避免代理程式與佇列管理程式之間未獲授權的連線，以及加密代理程式與佇列管理程式之間的訊息資料流量。

下列 MFT 內容檔包括 SSL 內容：

- [agent.properties](#) 檔
- [coordination.properties](#) 檔
- [command.properties](#) 檔
-  [logger.properties](#) 檔

如需搭配使用 SSL 與 Managed File Transfer 的相關資訊，請參閱第 44 頁的『配置 MFT 的 SSL 或 TLS 加密』。

若為 IBM WebSphere MQ 7.5 或更新版本，可在某些代表檔案或目錄位置的 Managed File Transfer 內容中使用環境變數。這可讓執行產品的組件時所使用的檔案或目錄，隨著環境變更（例如執行程序的使用者為何）而改變其所在位置。如需相關資訊，請參閱第 452 頁的『在 MFT 內容中使用環境變數』。

agent.properties 檔的 SSL 內容

代理程式的 `agent.properties` 檔位於 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` 目錄中。它包含的內容包括下列 SSL 內容：

內容名稱	說明	預設值
agentSslCipherSpec	<p>指定在代理程式與代理程式佇列管理程式之間交換資料時所使用的通訊協定、雜湊演算法及加密演算法，以及加密金鑰中使用的位元數。</p> <p>agentSslCipherSpec 的值是 CipherSpec 名稱。此 CipherSpec 名稱與代理程式佇列管理程式通道上使用的 CipherSpec 名稱相同。有效 CipherSpec 名稱的清單內含在 IBM MQ classes for Java 中的 SSL/TLS CipherSpec 及 CipherSuite 與 IBM MQ classes for JMS 中的 SSL/TLS CipherSpec 及 CipherSuite 中。</p> <p>agentSslCipherSpec 與 agentSslCipherSuite 類似。如果同時指定 agentSslCipherSuite 及 agentSslCipherSpec，則會使用 agentSslCipherSpec 的值。</p>	無
agentSslCipherSuite	<p>指定關於代理程式與代理程式佇列管理程式如何交換資料的各種 SSL 觀點。</p> <p>agentSslCipherSuite 的值是 CipherSuite 名稱。此 CipherSuite 名稱對映至代理程式佇列管理程式通道上使用的 CipherSpec 名稱。如需相關資訊，請參閱 CipherSuite 及 CipherSpec 名稱對映。</p> <p>agentSslCipherSuite 與 agentSslCipherSpec 類似。如果同時指定 agentSslCipherSuite 及 agentSslCipherSpec，則會使用 agentSslCipherSpec 的值。</p>	無
agentSslPeerName	<p>指定必須符合代理程式佇列管理程式所提供名稱的識別名稱架構。識別名稱用來檢查佇列管理程式在連線時提供的識別憑證。</p>	無
agentSslTrustStore	<p>指定代理程式信任的憑證位置。agentSslTrustStore 的值是檔案路徑。如果它是 Windows 檔案路徑，則必須跳出 (\) 反斜線字元 (\)。</p> <p>若為 IBM WebSphere MQ 7.5 或更新版本，此內容值可包含環境變數。</p>	無
agentSslKeyStore	<p>指定代理程式的私密金鑰位置。agentSslKeyStore 的值是檔案路徑。如果它是 Windows 檔案路徑，則反斜線字元 (\) 必須跳出 (\)。只有在代理程式佇列管理程式需要用戶端鑑別時，才需要此內容。</p> <p>若為 IBM WebSphere MQ 7.5 或更新版本，此內容值可包含環境變數。</p>	無
agentSslFipsRequired	<p>指定您想要在代理程式層次上啟用 FIPS 支援。此內容值可以是 true 或 false。如需相關資訊，請參閱第 560 頁的『MFT 中的 FIPS 支援』。</p>	false
agentSslKeyStoreType	<p>您想要使用的 SSL 金鑰儲存庫類型。支援 JKS 及 PKCS#12 金鑰儲存庫。此內容值可以是 jks 或 pkcs12。</p>	jks
agentSslKeyStoreCredentialsFile	<p>包含 agentSslKeyStore 認證的檔案路徑。</p> <p>若為 IBM WebSphere MQ 7.5 或更新版本，此內容值可包含環境變數。</p>	在 Windows 上，此內容的預設值為 %USERPROFILE%\MQMFTCredentials.xml，在其他平台上，則為 \$HOME/MQMFTCredentials.xml。

表 50: <i>agent.properties</i> 檔的 SSL 內容 (繼續)		
內容名稱	說明	預設值
agentSslTrustStoreType	您想要使用的 SSL 金鑰儲存庫類型。支援 JKS 及 PKCS#12 金鑰儲存庫。此內容值可以是 jks 或 pkcs12。	jks
agentSslTrustStoreCredentialsFile	包含 agentSslTrustStore 認證的檔案路徑。 若為 IBM WebSphere MQ 7.5 或更新版本，此內容值可包含環境變數。	在 Windows 上，此內容的預設值為 %USERPROFILE%\MQMFTCredentials.xml，在其他平台上，則為 \$HOME/MQMFTCredentials.xml。

coordination.properties 檔的 SSL 內容

coordination.properties 檔位於 *MQ_DATA_PATH/mqft/config/coordination_qmgr_name* 目錄中。它包含的內容包括下列 SSL 內容：

表 51: <i>coordination.properties</i> 檔的 SSL 內容		
內容名稱	說明	預設值
coordinationSslCipherSpec	指定在指令與協調佇列管理程式之間交換資料時所使用的通訊協定、雜湊演算法及加密演算法，以及加密金鑰中使用的位元數。 coordinationSslCipherSpec 的值是 CipherSpec 名稱。此 CipherSpec 名稱與協調佇列管理程式通道上使用的 CipherSpec 名稱相同。有效 CipherSpec 名稱的清單內含在 IBM MQ classes for Java 中的 SSL/TLS CipherSpec 及 CipherSuite 與 IBM MQ classes for JMS 中的 SSL/TLS CipherSpec 及 CipherSuite 中。 coordinationSslCipherSpec 與 coordinationSslCipherSuite 類似。如果同時指定 coordinationSslCipherSuite 及 coordinationSslCipherSpec，則會使用 coordinationSslCipherSpec 的值。	無
coordinationSslCipherSuite	指定關於指令與協調佇列管理程式如何交換資料的各種 SSL 觀點。 coordinationSslCipherSuite 的值是 CipherSuite 名稱。此 CipherSuite 名稱對映至代理程式佇列管理程式通道上使用的 CipherSpec 名稱。如需相關資訊，請參閱 CipherSuite 及 CipherSpec 名稱對映。 coordinationSslCipherSuite 與 coordinationSslCipherSpec 類似。如果同時指定 coordinationSslCipherSuite 及 coordinationSslCipherSpec，則會使用 coordinationSslCipherSpec 的值。	無
coordinationSslPeerName	指定必須符合協調佇列管理程式所提供名稱的識別名稱架構。識別名稱用來檢查協調佇列管理程式在連線時提供的識別憑證。	無
coordinationSslTrustStore	指定指令信任的憑證位置。coordinationSslTrustStore 的值是檔案路徑。如果它是 Windows 檔案路徑，則必須跳出 (\) 反斜線字元 (\)。 若為 IBM WebSphere MQ 7.5 或更新版本，此內容值可包含環境變數。	無
coordinationSslTrustStoreType	您想要使用的 SSL 金鑰儲存庫類型。支援 JKS 及 PKCS#12 金鑰儲存庫。此內容值可以是 jks 或 pkcs12。	jks
coordinationSslTrustStoreCredentialsFile	包含 coordinationSslTrustStore 認證的檔案路徑。 若為 IBM WebSphere MQ 7.5 或更新版本，此內容值可包含環境變數。	在 Windows 上，此內容的預設值為 %USERPROFILE%\MQMFTCredentials.xml，在其他平台上，則為 \$HOME/MQMFTCredentials.xml。

內容名稱	說明	預設值
coordinationSslKeyStore	指定指令的私密金鑰位置。coordinationSslKeyStore 的值是檔案路徑。如果它是 Windows 檔案路徑，則反斜線字元 (\) 必須跳出 (\\)。只有在協調佇列管理程式需要用戶端鑑別時，才需要此內容。 若為 IBM WebSphere MQ 7.5 或更新版本，此內容值可包含環境變數。	無
coordinationSslKeyStoreType	您想要使用的 SSL 金鑰儲存庫類型。支援 JKS 及 PKCS#12 金鑰儲存庫。此內容值可以是 jks 或 pkcs12。	jks
coordinationSslKeyStoreCredentialsFile	包含 coordinationSslKeyStore 認證的檔案路徑。 若為 IBM WebSphere MQ 7.5 或更新版本，此內容值可包含環境變數。	在 Windows 上，此內容的預設值為 %USERPROFILE%\MQMFTCredentials.xml，在其他平台上，則為 \$HOME/MQMFTCredentials.xml。
coordinationSslFipsRequired	指定您想要在協調佇列管理程式層次上啟用 FIPS 支援。此內容值可以是 true 或 false。如需相關資訊，請參閱第 560 頁的『MFT 中的 FIPS 支援』。	false

command.properties 檔的 SSL 內容

command.properties 檔位於 MQ_DATA_PATH/mqft/config/coordination_qmgr_name 目錄中。它包含的內容包括下列 SSL 內容：

內容名稱	說明	預設值
connectionSslCipherSpec	指定在指令與指令佇列管理程式之間交換資料時所使用的通訊協定、雜湊演算法及加密演算法，以及加密金鑰中使用的位元數。 connectionSslCipherSpec 的值是 CipherSpec 名稱。此 CipherSpec 名稱與指令佇列管理程式通道上使用的 CipherSpec 名稱相同。有效 CipherSpec 名稱的清單包括在 Java 的 IBM MQ 類別中的 SSL/TLS CipherSpecs 和 CipherSuites 和 JMS 的 IBM MQ 類別中的 SSL/TLS CipherSpecs 和 CipherSuites 中。 connectionSslCipherSpec 與 connectionSslCipherSuite 類似。如果同時指定 connectionSslCipherSuite 及 connectionSslCipherSpec，則會使用 connectionSslCipherSpec 的值。	無
connectionSslCipherSuite	指定關於指令與指令佇列管理程式如何交換資料的各種 SSL 觀點。 connectionSslCipherSuite 的值是 CipherSuite 名稱。此 CipherSuite 名稱對映至代理程式佇列管理程式通道上使用的 CipherSpec 名稱。如需相關資訊，請參閱 CipherSuite 及 CipherSpec 名稱對映。 connectionSslCipherSuite 與 connectionSslCipherSpec 類似。如果同時指定 connectionSslCipherSuite 及 connectionSslCipherSpec，則會使用 connectionSslCipherSpec 的值。	無
connectionSslPeerName	指定必須符合指令佇列管理程式所提供名稱的識別名稱架構。識別名稱用來檢查指令佇列管理程式在連線時提供的識別憑證。	無
connectionSslTrustStore	指定指令信任的憑證位置。connectionSslTrustStore 的值是檔案路徑。如果它是 Windows 檔案路徑，則必須跳出 (\\) 反斜線字元 (\)。 若為 IBM WebSphere MQ 7.5 或更新版本，此內容值可包含環境變數。	無
connectionSslTrustStoreType	您想要使用的 SSL 信任儲存庫類型。支援 JKS 及 PKCS#12 金鑰儲存庫。此內容值可以是 jks 或 pkcs12。	jks

內容名稱	說明	預設值
connectionSslTrustStoreCredentialsFile	包含 connectionSslTrustStore 認證的檔案路徑。 若為 IBM WebSphere MQ 7.5 或更新版本, 此內容值可包含環境變數。	在 Windows 上, 此內容的預設值為 %USERPROFILE%\MQMFTCredential s.xml, 在其他平台上, 則為 \$HOME/MQMFTCredentials.xml。
connectionSslKeyStore	指定指令的私密金鑰位置。connectionSslKeyStore 的值是檔案路徑。如果它是 Windows 檔案路徑, 則反斜線字元 (\) 必須跳出 (\\)。只有在指令佇列管理程式需要用戶端鑑別時, 才需要此內容。 若為 IBM WebSphere MQ 7.5 或更新版本, 此內容值可包含環境變數。	無
connectionSslKeyStoreType	您想要使用的 SSL 金鑰儲存庫類型。支援 JKS 及 PKCS#12 金鑰儲存庫。此內容值可以是 jks 或 pkcs12。 若為 IBM WebSphere MQ 7.5 或更新版本, 此內容值可包含環境變數。	jks
connectionSslKeyStoreCredentialsFile	包含 connectionSslKeyStore 認證的檔案路徑。 若為 IBM WebSphere MQ 7.5 或更新版本, 此內容值可包含環境變數。	在 Windows 上, 此內容的預設值為 %USERPROFILE%\MQMFTCredential s.xml, 在其他平台上, 則為 \$HOME/MQMFTCredentials.xml。
connectionSslFipsRequired	指定您想要在指令佇列管理程式層次上啟用 FIPS 支援。此內容值可以是 true 或 false。如需相關資訊, 請參閱第 560 頁的『MFT 中的 FIPS 支援』。	false

logger.properties 檔的 SSL 內容

V 9.0.4

logger.properties 檔案位於 *MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name* 目錄中。支援日誌程式佇列管理程式的用戶端模式連線所需的內容包括下列 SSL 內容:

內容名稱	說明	預設值
wmqfte.Ssl.CipherSuite	指定日誌程式與日誌程式佇列管理程式如何交換資料的各個 TLS 層面。 wmqfte.Ssl.CipherSuite 的值是 CipherSuite 名稱。此 CipherSuite 名稱對映至日誌程式佇列管理程式通道上使用的 CipherSpec 名稱。 如需相關資訊, 請參閱 CipherSuite 及 CipherSpec 名稱對映。	無預設值
wmqfte.Ssl.PeerName	指定必須符合日誌程式佇列管理程式所提供名稱的識別名稱架構。識別名稱用來檢查佇列管理程式在連線時提供的識別憑證。	無預設值
wmqfte.Ssl.TrustStore	指定日誌程式信任的憑證位置。 wmqfte.Ssl.TrustStore 的值是檔案路徑。如果此檔案路徑是 Windows 檔案路徑, 則必須將反斜線字元 (\) 跳出為具有進一步的反斜線字元 (\\)。 請注意, 此內容值可包含環境變數。	無預設值

表 53: <i>logger.properties</i> 檔的 SSL 內容 (繼續)		
內容名稱	說明	預設值
wmqfte.Ssl.TrustStoreCredentialsFile	包含 wmqfte.Ssl.TrustStore 認證的檔案路徑。 請注意，此內容值可包含環境變數。	無預設值
wmqfte.Ssl.TrustStoreType	您想要使用的 SSL 金鑰儲存庫類型。支援 JKS 及 PKCS#12 金鑰儲存庫。此內容值可以是 jks 或 pkcs12。	jks
wmqfte.Ssl.KeyStore	指定日誌程式的私密金鑰位置。 wmqfte.Ssl.KeyStore 的值是檔案路徑。如果此檔案路徑是 Windows 檔案路徑，則必須將反斜線字元 (\) 跳出為具有進一步的反斜線字元 (\\)。 請注意，此內容值可包含環境變數。	無預設值
wmqfte.Ssl.KeyStore.CredentialsFile	包含 wmqfte.Ssl.KeyStore 認證的檔案路徑。 請注意，此內容值可包含環境變數。	無預設值
wmqfte.Ssl.KeyStoreType	您想要使用的 SSL 金鑰儲存庫類型。支援 JKS 及 PKCS#12 金鑰儲存庫。此內容值可以是 jks 或 pkcs12。	jks
wmqfte.Ssl.FipsRequired	指定您想要在日誌程式層次上啟用 FIPS 支援。此內容值可以是 true 或 false。如需相關資訊，請參閱第 560 頁的『MFT 中的 FIPS 支援』。	false

MFT 的 SHA-2 CipherSpec 及 CipherSuite

Managed File Transfer 支援 SHA-2 CipherSpec 及 CipherSuite。

在 IBM MQ 8.0 中，若要在代理程式與 IBM MQ 佇列管理程式之間的連線上啟用 SHA-2 CipherSpec 和 CipherSuite，您必須使用 IBM JRE 6.0 SR13 FP2、7.0 SR4 FP2 或更新版本。

若要在 Managed File Transfer 第 8 版中啟用 SHA-2 CipherSpec 和 CipherSuite，以在 FTPS 模式下使用通訊協定橋接器連接到 FTPS 伺服器，您必須使用 IBM JRE 6.0 SR13 FP2、7.0 SR4 FP2 或更新版本。

如需適用於代理程式與 IBM MQ 佇列管理程式連線之 CipherSpec 及 CipherSuite 的相關資訊，請參閱 [SSL CipherSpec 及 CipherSuite](#)。

如需如何配置 CipherSpec 及 CipherSuite，以與通訊協定橋接器代理程式及 FTPS 伺服器搭配使用的相關資訊，請參閱第 558 頁的『通訊協定橋接器支援的 FTPS 伺服器』及第 484 頁的『通訊協定橋接器內容檔格式』。

IBM i JVM 不支援在 MQ 8.0 中指定 CipherSpecs 中詳述的較新密碼。因此，IBM i 平台的 SHA-2 支援只涵蓋在 MQ 7.5 中指定 CipherSpec 所詳述的那些密碼。

如果希望符合 SP 800-131A，則必須滿足下列需求：

- 必須使用已適當配置的 FTPS；不支援 SFTP。
- 遠端伺服器必須只傳送符合 SP 800-131A 標準的密碼組合。

相關參考

第 508 頁的『MFT 的 SSL 內容』

對 IBM MQ 及 Managed File Transfer 使用 SSL 或 TLS，可避免代理程式與佇列管理程式之間未獲授權的連線，以及加密代理程式與佇列管理程式之間的訊息資料流量。

MFT 檔案日誌程式配置檔

除了 *logger.properties* 檔案之外，Managed File Transfer 獨立式檔案日誌程式也在其配置目錄中具有 XML 配置檔。此配置檔稱為 *FileLoggerFormat.xml*，可定義檔案日誌程式將訊息寫入至日誌檔時所使用的格式。此檔案內容必須符合 *FileLoggerFormat.xsd* 檔中定義的 XML 綱目。

MFT 獨立式檔案日誌程式的預設日誌格式

Managed File Transfer 獨立式檔案日誌程式的預設日誌檔格式定義。

```
<?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>

```

相關參考

第 90 頁的『MFT 獨立式檔案日誌程式格式』

檔案日誌程式寫入的訊息資訊格式可定義於 FileLoggerFormat.xml 檔中。

第 518 頁的『獨立式檔案日誌程式格式 XSD』

獨立式檔案格式的綱目。

獨立式檔案日誌程式格式 XSD

獨立式檔案格式的綱目。

綱目

```

<?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>

```

相關參考

第 90 頁的『MFT 獨立式檔案日誌程式格式』

檔案日誌程式寫入的訊息資訊格式可定義於 `FileLoggerFormat.xml` 檔中。

第 514 頁的『MFT 獨立式檔案日誌程式的預設日誌格式』

Managed File Transfer 獨立式檔案日誌程式的預設日誌檔格式定義。

SYSTEM.FTE 主題

SYSTEM.FTE 主題是協調佇列管理程式上的主題，供 Managed File Transfer 用來記載傳送，以及儲存代理程式、監視器、排程及範本的相關資訊。

主題結構

```
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

本主題包含的保留發佈資訊說明 Managed File Transfer 網路中的代理程式及其內容。本主題上的訊息會隨著代理程式狀態而定期更新。如需相關資訊，請參閱 [第 625 頁的『MFT 代理程式狀態訊息格式』](#)。

SYSTEM.FTE/monitors/agent_name

本主題包含的保留發佈資訊說明與代理程式 *agent_name* 相關聯的資源監視器。保留發佈資訊的 XML 符合 MonitorList.xsd 綱目。如需相關資訊，請參閱 [第 626 頁的『MFT 監視器清單訊息格式』](#)。

SYSTEM.FTE/Scheduler/agent_name

本主題包含的保留發佈資訊說明與代理程式 *agent_name* 相關聯的所有作用中排程。保留發佈資訊的 XML 符合 ScheduleList.xsd 綱目。如需相關資訊，請參閱 [第 630 頁的『MFT 排程清單訊息格式』](#)。

SYSTEM.FTE/Templates

本主題包含的保留發佈資訊說明 Managed File Transfer 拓撲中定義的所有範本。

- 與每一個範本相關聯的發佈資訊會發佈至名稱為 SYSTEM.FTE/Templates/*template_ID* 的子主題。

如需此保留發佈資訊的內容範例，請參閱 [第 634 頁的『MFT 範本 XML 訊息範例』](#)。

SYSTEM.FTE/Transfers/agent_name

本主題包含的發佈資訊說明源自於代理程式 *agent_name* 的傳送狀態。與每一個傳送相關聯的發佈資訊會發佈至名稱為 SYSTEM.FTE/Transfers/*agent_name/transfer_ID* 的子主題。「IBM MQ Explorer」外掛程式使用這些發佈資訊，以提供個別傳送的進度資訊。發佈資訊的 XML 符合 TransferStatus.xsd 綱目。如需相關資訊，請參閱 [第 635 頁的『檔案傳送狀態訊息格式』](#)。

SYSTEM.FTE/Log/agent_name

本主題包含的發佈資訊記載源自於代理程式 *agent_name* 的傳送、監視器及排程的相關資訊。資料庫日誌程式可記載這些發佈資源，以提供 Managed File Transfer 網路中發生的事件審核記錄。

- 與每一個傳送相關聯的發佈資訊會發佈至名稱為 SYSTEM.FTE/Log/*agent_name/transfer_ID* 的子主題，且發佈資訊的 XML 符合 TransferLog.xsd 綱目。如需相關資訊，請參閱 [第 637 頁的『檔案傳送日誌訊息格式』](#)。
- 與每一個排定傳送相關聯的發佈資訊會發佈至名稱為 SYSTEM.FTE/Log/*agent_name/schedule_ID* 的子主題，且發佈資訊的 XML 符合 ScheduleLog.xsd 綱目。如需相關資訊，請參閱 [第 658 頁的『排定的檔案傳送日誌訊息格式』](#)。
- 與每一個監視器相關聯的發佈資訊會發佈至名稱為 SYSTEM.FTE/Log/*agent_name/Monitors/monitor_name/monitor_ID* 的子主題，且發佈資訊的 XML 符合 MonitorLog.xsd 綱目。如需相關資訊，請參閱 [第 663 頁的『MFT 監視器日誌訊息格式』](#)。

MFT 代理程式佇列設定

fteCreateAgent 指令產生的 MQSC 指令 Script 可建立代理程式佇列，且參數設定為下列值。如果您未使用提供的 MQSC Script 來建立佇列，而是手動建立佇列，請確保將下列參數設定為給定的值。

代理程式作業佇列

代理程式的作業佇列具有下列名稱：

- 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*

參數	值 (如果適用)
DEFPRTY	0
DEFSOPT	共用
GET	ENABLED
MAXDEPTH	5000
MAXMSGL	4194304
MSGDLVSQ	PRIORITY
PUT	ENABLED
RETINTVL	999999999
SHARE	
NOTRIGGER	
USAGE	正常
REPLACE	

代理程式權限佇列

代理程式的權限佇列具有下列名稱：

- 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*

參數	值 (如果適用)
DEFPRTY	0
DEFSOPT	共用

表 55: 代理程式權限佇列參數 (繼續)

參數	值 (如果適用)
GET	ENABLED
MAXDEPTH	0
MAXMSGL	0
MSGDLVSQ	PRIORITY
PUT	ENABLED
RETINTVL	999999999
SHARE	
NOTRIGGER	
USAGE	正常
REPLACE	

相關參考

第 326 頁的『[fteCreateAgent \(建立 MFT 代理程式\)](#)』

fteCreateAgent 指令會建立 Managed File Transfer Agent 及其相關聯的配置。

MFT 系統佇列及系統主題

Managed File Transfer 有許多僅供內部使用的系統佇列及一個系統主題。

名稱以 SYSTEM.FTE 是 Managed File Transfer (MFT) 的內部系統佇列。請勿刪除這些佇列，因為這樣做會使 IBM MQ MFT 無法正確運作。第 523 頁的表 56 顯示每一個佇列上的訊息類型：

表 56: 佇列名稱、類型及用法

佇列名稱	佇列類型	使用情形
SYSTEM.FTE.AUTHAGT1.agent_name	權限管理中心	佇列，用於配置傳送及接收傳送要求的權限。
SYSTEM.FTE.AUTHTRN1.agent_name	權限管理中心	用於配置權限以啟動及取消受管理傳送的佇列。同時啟動受管理呼叫。
SYSTEM.FTE.AUTHMON1.agent_name	權限管理中心	佇列，用於配置權限以容許使用者建立或刪除由相同使用者建立的資源監視器。
SYSTEM.FTE.AUTHOPS1.agent_name	權限管理中心	佇列，用於配置權限以刪除由另一個使用者建立的資源監視器及排程傳送。
SYSTEM.FTE.AUTHSCH1.agent_name	權限管理中心	佇列，用於配置權限以建立或刪除由相同使用者建立的排程傳送。
SYSTEM.FTE.AUTHADM1.agent_name	權限管理中心	使用 fteStopAgent 指令上的 -m 選項，配置權限以關閉代理程式的佇列。
SYSTEM.FTE.COMMAND.agent_name	作業	將指令要求傳送至代理程式的佇列。

表 56: 佇列名稱、類型及用法 (繼續)

佇列名稱	佇列類型	使用情形
SYSTEM.FTE.DATA.agent_name	作業	目的地代理程式用於保留來源代理程式所傳送資料的佇列。
SYSTEM.FTE.REPLY.agent_name	作業	用於接收來自目的地代理程式的回覆的佇列。
SYSTEM.FTE.STATE.agent_name	作業	用於保留傳送要求狀態的佇列。
SYSTEM.FTE.EVENT.agent_name	作業	用於保留資源監視器歷程的佇列。

如果代理程式正在參與「訊息轉為檔案」或「檔案轉為訊息」傳送，則可能需要修改 `SYSTEM.FTE.STATE.agent_name` 佇列的定義，以容許進行這些受管理傳送。如需此相關資訊，請參閱設定與訊息大小相關聯的 MQ 屬性及 MFT 內容的指引。



小心: 您不應變更其他系統佇列的定義。

此外，請勿修改或刪除 `SYSTEM.FTE` 主題，因為這也僅供內部使用。

暫時佇列

Managed File Transfer 建立的暫時佇列有許多用途。每一個佇列的名稱以 `WMQFTE` 開頭。依預設。（句點是預設字首的一部分）。如果您想要變更此字首，可以在 `command.properties` 檔及（或）`coordination.properties` 檔中使用 `dynamicQueuePrefix` 內容。`command.properties` 檔中的內容可用來設定暫時佇列的字首，建立這些暫時佇列是為了回應需要代理程式回應的指令。`coordination.properties` 檔案中的內容用來設定為了其他目的而建立的暫時佇列字首；例如 `WMQFTE.FTE.TIMECHCK.QUEUE`，其中 `WMQFTE` 是 `dynamicQueuePrefix` 內容所定義的值。

相關參考

第 301 頁的『限制 MFT 代理程式動作的使用者權限』

除了使用群組管理對資源的存取外，您還可以啟用額外的一層安全來限制使用者可以採取的 Managed File Transfer 代理程式動作。將代理程式權限佇列的權限授與使用者，使使用者有權執行特定的代理程式動作。

MFT 物件命名慣例

請對您的 Managed File Transfer 物件使用下列命名慣例：

- 代理程式及日誌程式名稱：
 - 長度上限為 28 個字元，且不區分大小寫。
 - 以小寫或大小寫混合格式輸入會轉換為大寫
 - 必須符合標準 IBM MQ 物件命名慣例。
 這些慣例詳述如下：[IBM MQ 物件的命名規則](#)。
- 除了 IBM MQ 物件命名慣例之外，還包括：
 - 代理程式名稱或日誌程式名稱中不能使用正斜線 (/) 字元
 - 代理程式名稱或日誌程式名稱中不能使用百分比 (%) 字元。
- 內容檔中的內容名稱須區分大小寫。
- 佇列管理程式名稱須區分大小寫。
- 檔案名稱在某些平台上須區分大小寫。
- 資源監視器及傳送範本名稱：
 - 不區分大小寫
 - 以小寫或大小寫混合格式輸入會轉換為大寫
 - 不得包含星號 (*)、百分比 (%) 或問號 (?) 字元
- 通訊協定檔案伺服器名稱必須：

- 長度下限為 2 個字元，但沒有長度上限
 - 不區分大小寫
 - 必須符合標準 IBM MQ 物件命名慣例。
- 這些慣例詳述如下：[IBM MQ 物件的命名規則](#)。

IBM i 整合檔案系統 (IFS) 中的檔案

IBM i IFS 中的檔案名稱不可包含下列任何字元：

- 反斜線 (\)
- 正斜線 (/)
- 冒號 (:)
- 星號 (*)
- 問號 (?)
- 雙引號 (")
- 小於符號 (<)
- 大於符號 (>)
- 垂直線 (|)

如果您嘗試將名稱中包含前述任何字元的檔案傳送至 IBM i IFS，這些檔案的傳送將會失敗。

資料集名稱

z/OS 資料集具有命名限制，會影響資料集名稱所能使用的名稱長度上限及可用字元數。PDS 資料集成員名稱的長度上限為 8 個字元，而且不能包含點 (.) 字元。在傳送至資料集時，您必須明確指定名稱，以確保這些命名限制不會造成問題。但當您從檔案傳送至 PDS 成員時，檔案路徑可能不會對映至 PDS 成員名稱。當您傳送至 PDS 資料集時，每一個來源檔案都會成為 PDS 成員，且每一個成員名稱都會從來源的名稱產生。

PDS 成員名稱是 z/OS 非限定名稱，由下列正規表示式定義：

```
[a-zA-Z$#@][a-zA-Z0-9$#@]{0-7}
```

下列方法可讓您將來源資料集或來源檔案名稱轉換成有效的 PDS 成員名稱。下列考量依其列出順序套用：

1. 只會使用名稱中位於最後一個正斜線 (/)、最後一個反斜線 (\) 或最後一個冒號 (:) 字元後面的字元。也就是說，只會使用檔案路徑的名稱部分。
2. 若為來源檔案（不是資料集或 PDS 成員），在最後一個點 (.) 字元後面（包含點字元）的字元，都會被忽略。
3. 若為長度超過八個字元的名稱，只會使用最後八個字元。
4. 點字元會取代為 at 符號 (@) 字元。
5. 無效字元會取代為 at 符號 (@) 字元。
6. 如果轉換後未產生任何字元，則 PDS 成員名稱會是 @。

MFT 管理參照

MFT 代理程式狀態值

fteListAgents 及 **fteShowAgentDetails** 指令可產生代理程式狀態資訊。此狀態有數個可能的值。

ACTIVE

代理程式正在執行，且正在傳送或接收檔案。代理程式正在定期發佈其狀態。在預期時段內收到前次更新。

備妥

代理程式正在執行，但未傳送或接收檔案。代理程式正在定期發佈其狀態。在預期時段內收到前次更新。

STARTING

正在啟動代理程式，但尚未準備好執行傳送。

UNREACHABLE

未在預期的時間間隔收到代理程式狀態更新。代理程式可能由於錯誤而停止執行，或突然關閉，或雖然在執行但發生通訊問題。如需相關資訊，請參閱 [第 255 頁的『代理程式顯示為處於 UNKNOWN 狀態時要執行的動作』](#)。

STOPPED

代理程式已停止。它已經以受管制的方式關閉。

ENDED UNEXPECTEDLY

代理程式非預期地結束。除非在 `maxRestartInterval` 時段內重新啟動 `maxRestartCount` 次以上，且 `maxRestartDelay` 值小於或等於 0，否則會自動重新啟動代理程式。如需這些內容的相關資訊，請參閱 [第 463 頁的『MFT agent.properties 檔案』](#)。

NO_INFORMATION

代理程式版本可能是 IBM WebSphere MQ File Transfer Edition 7.0.2 或更早版本。代理程式未以此指令可以處理的形式發佈更新項目。

不明

無法判斷代理程式的狀態。發佈的狀態可能無法被此工具辨識。如果您已在您的網路上混用產品版本，升級此工具的安裝版本可以解決此問題。

V 9.0.2 在 IBM MQ 9.0.2 中，當您執行指令或查看連接至協調管理程式的代理程式清單及其個別內容時，您可以看到代理程式的新 **Status Age** 值，其中顯示其前次報告狀態的經歷時間。如需相關資訊，請參閱 [第 255 頁的『代理程式顯示為處於 UNKNOWN 狀態時要執行的動作』](#)。

PROBLEM

代理程式指令處理程式可能無法運作。代理程式正在發佈狀態訊息，但這些狀態訊息已過期。

相關參考

[第 398 頁的『fteListAgents: 列出協調佇列管理程式的 MFT 代理程式』](#)

使用 **fteListAgents** 指令，以列出向特定協調佇列管理程式登錄的所有 Managed File Transfer 代理程式。

[第 436 頁的『fteShowAgentDetails』](#)

使用 **fteShowAgentDetails** 指令顯示特定 Managed File Transfer Agent 的詳細資料。這些是代理程式的 Managed File Transfer 協調佇列管理程式儲存的詳細資料。

[第 264 頁的『認為檔案傳送停滯時應該如何處理』](#)

在負載沈重的系統上，或來源與目的地代理程式之間發生網路問題時，傳送有時會呈現停滯在佇列或回復狀態。有許多因素可能會造成此狀況。

[第 255 頁的『代理程式顯示為處於 UNKNOWN 狀態時要執行的動作』](#)

您的代理程式正在執行中並順利回應 **ftePingAgent** 指令，且項目正在正常傳送。不過，**fteListAgents** 和 **fteShowAgentDetails** 指令以及 IBM MQ Explorer Managed File Transfer 外掛程式會將代理程式報告為處於 UNKNOWN 狀態。

Multi MFT 處理程序控制器概觀

IBM MQ Managed File Transfer (MFT) 處理程序控制器負責啟動 MFT 代理程式，並在該處理程序因任何原因而結束時重新啟動該處理程序。每個代理程式處理程序都有一個處理程序控制器。

註: 程序控制器僅適用於 IBM MQ for Multiplatforms。

z/OS 在 IBM MQ for z/OS 上，「自動重新啟動管理程式 (ARM)」會重新啟動代理程式處理程序。如需此作業的相關資訊，請參閱 [配置 MFT 以使用 z/OS Automatic Restart Manager \(ARM\)](#)

處理程序控制器的運作方式

執行 **fteStartAgent** 指令時，它會啟動該代理程式的處理程序控制器實例，然後處理程序控制器會啟動代理程式處理程序。

當 **fteStopAgent** 指令執行時，它會連接至該代理程式的處理程序控制器，並傳送停止要求給它。處理程序控制器會接收要求，停止代理程式處理程序，然後自行關閉。

處理程序控制器會監視代理程式處理程序。如果代理程式處理程序非預期地停止，處理程序控制器會重新啟動它。

依預設，如果代理程式處理程序在兩分鐘內停止五次，則處理程序控制器會自行關閉，且不會再次嘗試重新啟動代理程式。在此狀況下，您需要使用 **fteStartAgent** 指令手動重新啟動代理程式。

您可以透過修改下列代理程式內容來變更此行為：

- **maxRestartCount**
- **maxRestartDelay**
- **maxRestartInterval**

如果您已配置代理程式使用 BINDINGS 傳輸來連接至其代理程式佇列管理程式，則處理程序控制器會在啟動時建立與此佇列管理程式的連線。然後，處理程序控制器會監視此連線。

如果因為佇列管理程式變成無法使用而導致連線中斷，則處理程序控制器會停止代理程式，然後定期嘗試重新建立連線。

嘗試重新連線之間的時段由代理程式內容 **agentQMGrRetryInterval** 決定。一旦佇列管理程式再次可用，且處理程序控制器已能夠連接它，處理程序控制器即會重新啟動代理程式處理程序。

註：當代理程式配置為使用 CLIENT 傳輸連接至其代理程式佇列管理程式時，如果代理程式處理程序與佇列管理程式中斷連線，則代理程式處理程序會保持作用中狀態。在此狀況下，代理程式處理程序會定期嘗試重新連接自己。

如需本節所提及四個內容的相關資訊，請參閱 *MFT agent.properties* 檔案主題中的 [進階代理程式內容：程序控制器](#) 一節。

處理程序控制器日誌檔

程序控制器會將參考訊息寫入其事件日誌。這是稱為 `pceventN.log` 的檔案，其中 N 是一個數字，可在下列目錄中找到：`MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/`

每一個處理程序控制器事件日誌檔的大小，以及歷程檔案的數目，由代理程式內容 **outputLogSize** 及 **outputLogFiles** 決定。

如需本節中所提及內容的相關資訊，請參閱 *MFT agent.properties* 檔案主題的 [進階代理程式內容：追蹤和記載](#) 一節。

註：這些內容也用來決定代理程式日誌檔（稱為 `outputN.log`）以及處理程序控制器日誌檔的大小和數目。

寫入處理程序控制器事件日誌的訊息包括處理程序控制器的處理程序 ID，以及代理程式處理程序的處理程序 ID。以下顯示這些訊息的部分範例：

```
[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
```

```
ProcessControl I BFGPC0007I: IBM MQ Managed File Transfer process controller with process identifier 18736 started AGENT1@QM1 with process identifier 1304.
```

在這裡，與代理程式 AGENT1 相關聯的處理程序控制器以處理程序 ID 18736 執行。

一開始，它會啟動代理程式處理程序-此處理程序的處理程序 ID 為 1748。

在代理程式啟動之後不久，處理程序控制器偵測到它已非預期地停止，因此重新啟動它。重新啟動之後，代理程式處理程序的處理程序 ID 為 1304。

相關參考

第 529 頁的『MFT 程序控制器結束碼』

如果 Managed File Transfer 程序控制器結束，會產生 BFGPC0004I 訊息及結束碼，其提供程序控制器結束的原因。

MFT 代理程式程序控制器狀態值

fteShowAgentDetails 指令會產生代理程式程序控制器狀態資訊。此狀態有數個可能的值。

WAITING

代理程式處理程序控制器正在等待佇列管理程式變成可用，再啟動代理程式。

已啟動

代理程式處理程序控制器已啟動代理程式處理程序。

STOPPED

代理程式處理程序控制器已停止，可能是因為停止代理程式的要求，或是因為在重新啟動間隔內有太多代理程式處理程序重新啟動。

RECOVERING

代理程式處理程序非預期地停止，而程序控制器將嘗試重新啟動它。

ISTOPPING

代理程式處理程序收到立即關閉的要求。當代理程式處理程序停止時，程序控制器會停止。

CSTOPPING

代理程式處理程序收到以控制方式關閉的要求。當代理程式處理程序停止時，程序控制器會停止。

不明

無法判斷代理程式處理程序控制器的狀態。可能是代理程式處理程序控制器未在執行中，或是其在不同於執行 **fteShowAgentDetails** 指令的系統上執行。

相關參考

第 436 頁的『fteShowAgentDetails』

使用 **fteShowAgentDetails** 指令顯示特定 Managed File Transfer Agent 的詳細資料。這些是代理程式的 Managed File Transfer 協調佇列管理程式儲存的詳細資料。

MFT 日誌程式狀態值

fteShowLoggerDetails 指令會產生日誌程式狀態資訊。此狀態有數個可能的值。

ACTIVE

日誌程式正在執行，且正在傳送或接收檔案。日誌程式正在定期發佈其狀態。在預期時段內收到前次更新。

備妥

日誌程式正在執行，但未傳送或接收檔案。日誌程式正在定期發佈其狀態。在預期時段內收到前次更新。

STARTING

日誌程式正在啟動，但尚未備妥以執行傳送。

UNREACHABLE

未在預期的時間間隔收到日誌程式狀態更新項目。日誌程式可能由於錯誤已停止執行，或已突然關閉，或者雖然正在執行，但發生通訊問題。

STOPPED

已停止日誌程式。它已經以受管制的方式關閉。

ENDED UNEXPECTEDLY

日誌程式非預期地結束。日誌程式將自動重新啟動，除非在 `maxRestartInterval` 時段內重新啟動超過 `maxRestartCount` 次，且 `maxRestartDelay` 值小於或等於 0。如需這些內容的相關資訊，請參閱第 99 頁的『MFT 日誌程式配置內容』。

針對 **fteShowLoggerDetails** 指令，這個狀態的詳細資料會包括狀態碼，也就是日誌程式程序結束碼。請參閱「程序結束碼」以取得已知的結束碼清單。

NO_INFORMATION

日誌程式版本可能是 IBM WebSphere MQ File Transfer Edition 7.0.2 或更舊版本。日誌程式未以此指令可以處理的形式發佈更新項目。

不明

無法判斷日誌程式的狀態。發佈的狀態可能無法被此工具辨識。如果您已在您的網路上混用產品版本，升級此工具的安裝版本可以解決此問題。

PROBLEM

日誌程式指令處理程式可能無法運作。日誌程式正在發佈狀態訊息，但這些狀態訊息已過期。

相關參考

第 443 頁的『fteShowLoggerDetails』

使用 **fteShowLoggerDetails** 指令可顯示特定 Managed File Transfer 日誌程式的詳細資料。

MFT 日誌程式程序控制器狀態值

fteShowLoggerDetails 指令會產生日誌程式程序控制器狀態資訊。此狀態有數個可能的值。

WAITING

日誌程式程序控制器正在等待佇列管理程式變成可用，然後再啟動日誌程式。

已啟動

日誌程式程序控制器已啟動日誌程式程序。

STOPPED

日誌程式程序控制器已停止，可能是因為停止日誌程式的要求，或是因為在重新啟動間隔內有太多日誌程式程序重新啟動。

RECOVERING

日誌程式程序非預期地停止，程序控制器將嘗試重新啟動它。

ISTOPPING

日誌程式程序收到立即關閉的要求。當日誌程式程序停止時，程序控制器即會停止。

CSTOPPING

日誌程式程序收到以控制方式關閉的要求。當日誌程式程序停止時，程序控制器即會停止。

不明

無法判斷日誌程式程序控制器的狀態。可能是日誌程式程序控制器未在執行中，或是其在不同於執行 **fteShowLoggerDetails** 指令的系統上執行。

相關參考

第 443 頁的『fteShowLoggerDetails』

使用 **fteShowLoggerDetails** 指令可顯示特定 Managed File Transfer 日誌程式的詳細資料。

MFT 程序控制器結束碼

如果 Managed File Transfer 程序控制器結束，會產生 BFGPC0004I 訊息及結束碼，其提供程序控制器結束的原因。

出現下列訊息指出程序控制器已結束：

BFGPC0004I IBM MQ Managed File Transfer 程序控制器已結束，結束碼為 `reason_code`。

其中 `reason_code` 顯示程序控制器結束的原因。

註: 來自程序控制器的結束碼多半反映標準作業系統結束碼, 但部分結束碼是針對特定用途而定義的, 它們在程序控制器日誌檔中一律隨附有特定訊息。

表 57: 原因碼值	
原因碼	說明
RC_SUCCESS = 0	程序控制器順利結束。
RC_FAILURE = 1	一般程序控制器失敗回覆碼 (一般應不會傳回)。
RC_EXIT = 2	程序控制器被迫結束 (例如, 診斷系統要求程序控制器終止)。
RC_ABEND = 70	程序控制器發生無法復原的問題, 強制終止中。
RC_QMUNAVAIL = 75	程序控制器無法繼續, 因為程序控制器的佇列管理程式無法使用。
RC_CONFIG = 78	程序控制器無法繼續, 因為啟動配置資料有問題。

這些結束碼會寫入 pceventX.log, 其中 X 可以是任何數字, 例如, 日誌檔名稱可以是 pcevent0.log。

相關資訊

[Managed File Transfer 診斷訊息: BFGPC0001 - BFGPC9999](#)

傳送檔案的準則

視傳送的來源及目的地作業系統以及是以二進位還是以文字模式進行傳送而定, 對於預期發生的行為有一些準則可循。

如需進一步資訊, 請閱讀相關主題。

相關參考

[第 531 頁的『在 z/OS 與分散式系統之間傳送檔案及資料集』](#)

您可以使用 Managed File Transfer, 在 z/OS 與分散式檔案系統之間傳送檔案及支援的資料集類型。請仔細檢閱下列行為, 視您在其中來回傳送的系統類型而定。

[第 532 頁的『在 z/OS 上的資料集之間傳送』](#)

您可以使用 Managed File Transfer 在 z/OS 資料集之間進行傳送。請仔細檢閱下列行為, 以確保正確傳送資料集。

[第 534 頁的『從 Connect:Direct 節點來回傳送資料集』](#)

您可以使用 Connect:Direct 橋接器, 在 Managed File Transfer 代理程式與 IBM Sterling Connect:Direct 節點之間傳送資料集。您可以將資料集指定為傳送來源及/或傳送目的地。

[第 536 頁的『Connect:Direct 程序陳述式參數與 BPXWDYN 鍵之間的對映』](#)

當您針對來源或目的地為 Connect:Direct 節點的資料集提交傳送要求時, 您提供的任何受支援的 BPXWDYN 鍵, 均會轉換為 Connect:Direct 程序接受的格式。

[第 540 頁的『不得與 MFT 一起使用的 BPXWDYN 內容』](#)

使用 **fteCreateTemplate** 指令、**fteCreateTransfer** 指令或 `agent.properties` 檔中的 **bpxwdynAllocAdditionalOptions** 內容時, 不得指定某些 BPXWDYN 選項。

[第 540 頁的『使用 MFT 傳送文字檔』](#)

文字檔傳送涉及將檔案的字碼頁從一種字碼頁轉換成另一種字碼頁。文字檔傳送也涉及在系統之間轉換 CRLF (歸位-換行) 字元。本主題彙總 Managed File Transfer 的文字檔傳送行為。

[第 542 頁的『在 Connect:Direct 與 MFT 之間傳送文字檔』](#)

文字傳送涉及將檔案的字碼頁從一種字碼頁轉換為另一種字碼頁。文字檔傳送還會涉及在系統之間轉換 CRLF (換行, carriage return line feed) 字元。本主題彙總 Managed File Transfer Agent 與 Connect:Direct 節點之間傳送時的文字檔傳送行為。

[第 543 頁的『與通訊協定橋接器代理程式來回傳送檔案』](#)

您可以使用通訊協定橋接器代理程式, 從 Managed File Transfer 網路外部的 FTP 或 SFTP 檔案伺服器來回傳送檔案。

[第 543 頁的『與 IBM i 系統來回傳送檔案』](#)

如果您在文字模式下使用 Managed File Transfer 與 IBM i 系統來回傳送檔案，並且想要轉換檔案中的資料，則請考量本主題中的資訊。

第 547 頁的『傳送位於 IBM i 上 QSYS.LIB 中的儲存檔』

Managed File Transfer 支援在兩個 IBM i 系統之間傳送位於 QSYS.LIB 檔案系統中的儲存檔。要求對儲存檔進行檔案傳送時，請考量下列資訊。

第 548 頁的『傳送世代資料群組 (GDG)』

Managed File Transfer 支援對於 z/OS 上的來源及目的地資料集使用世代資料群組 (GDG)。支援絕對及相對 GDG 名稱。在您寫入新世代時，基本 GDG 必須存在。

第 549 頁的『搭配使用萬用字元與 MFT』

當您指定用於檔案傳送的來源檔案名稱及來源檔路徑時，可以使用萬用字元。這可讓您同步選取多個檔案。

在 z/OS 與分散式系統之間傳送檔案及資料集

您可以使用 Managed File Transfer，在 z/OS 與分散式檔案系統之間傳送檔案及支援的資料集類型。請仔細檢閱下列行為，視您在其中來回傳送的系統類型而定。

Managed File Transfer 支援對於 z/OS 上的來源及目的地資料集使用世代資料群組 (GDG)。支援絕對及相對 GDG 名稱。在您寫入新世代時，基本 GDG 必須存在。

當您將檔案或資料集傳送至磁帶時，任何已在磁帶上的現有資料集都會被取代。新資料集的屬性將根據傳送定義中所傳遞的屬性進行設定。如果未指定任何屬性，則會將屬性設定為與來源資料集的那些屬性相同；如果來源是檔案時，則會設為預設值。現有磁帶資料集的屬性會被忽略。

檔案至資料集的傳送 - 二進位傳送

目的地資料集的格式決定目的地記錄長度。請確定資料集存在於目的地系統上，或以正確屬性指定目的地資料集，以適當地建立資料集。如果未指定屬性，則系統會指定下列預設值：具有未定義的記錄格式的實體循序資料集，以及裝置的區塊大小上限 (BLKSIZE) (如 DEVTYPE 巨集所傳回)。例如，若為 DASD，此大小為 6144；若為磁帶，此大小為 32760。如果您想要將分散式系統上的檔案以二進位模式傳送至 z/OS 資料集，請注意下列行為：

實體循序 (PS) 目的地資料集：

- 分散式系統上的來源檔案會以循序讀取方式填入每一個記錄或區塊。
- 在可變格式資料集上，將每一個記錄填入容量中。

分割的資料集 (PDS) 目的地資料集：

- 每一個來源檔案會複製到具有相同或相等名稱的 PDS 成員。如果檔案名稱比成員名稱容許的長度上限還要長，則檔案名稱會轉換成有效的成員名稱。如需成員名稱的相關資訊，請參閱物件命名慣例。如果來源檔案是目錄，該目錄中的每一個檔案都會變成 PDS 的成員。
- 如果 PDS 成員已存在，且您已指定要改寫用於傳送的現有目的地檔案，則會改寫該成員。如果未指定改寫，則傳送會失敗。
- 分散式系統上的來源檔案會以循序讀取方式填入該成員的每一個記錄或區塊。
- 在可變格式 PDS 成員上，將每一個記錄填入容量中。

檔案至資料集的傳送 - 文字傳送

目的地資料集的格式決定目的地記錄長度。請確定資料集存在於目的地系統上，或以正確屬性指定目的地資料集，以適當地建立資料集。如果您想要將分散式系統上的檔案以文字模式傳送至 z/OS 資料集，請注意下列行為：

實體循序 (PS) 目的地資料集：

- 每一行文字都會變成記錄 (或未定義記錄格式 (RECFM=U) 資料集的區塊)。行尾字元不會出現在資料集記錄中 (僅限非 ASA 資料集)。
- 當 ASA 格式控制字元用於目的地資料集時，行尾字元會有效地轉換成相等的 ASA 格式控制碼。
- 當字行長度超過記錄時，該行會在記錄界限處分割並流向下一個記錄。

PDS 目的地資料集：

- 每一個來源檔案會複製到具有相同或相等名稱的 PDS 成員。如果檔案名稱比成員名稱容許的長度上限還要長，則檔案名稱會轉換成有效的成員名稱。如需成員名稱的相關資訊，請參閱物件命名慣例。如果來源檔案是目錄，該目錄中的每一個檔案都會變成 PDS 的成員。
- 如果 PDS 成員已存在，且您已指定要改寫用於傳送的現有目的地檔案，則會改寫該成員。如果未指定改寫，則傳送會失敗。
- 每一行文字都會變成記錄（或未定義記錄格式 (RECFM=U) 資料集的區塊）。行尾字元不會出現在成員記錄中（僅限非 ASA 資料集）。
- 當 ASA 格式控制字元用於目的地資料集時，行尾字元會有效地轉換成相等的 ASA 格式控制碼。
- 當字行長度超過記錄時，該行會在記錄界限處分割並流向下一個記錄。

資料集至檔案的傳送 - 二進位及文字傳送

如果您想要以二進位或文字模式進行「資料集至檔案」傳送，請注意下列行為：

- 以二進位形式將每一個記錄的內容傳送至檔案；不會傳送記錄、區塊格式資訊或 ASA 格式控制字元。
- 針對文字傳送，每一個資料集記錄都會變成一行轉換成目的地代理程式字碼頁的文字。亦即，若為 Windows 目的地系統，會附加歸位換行 (CRLF) 字元，若為 UNIX 目的地系統，會附加換行 (CR) 字元。
- **非 VSAM 及 PS 來源資料集。** 來源資料集的記錄會傳送至目的地檔案並連結在一起。如果目的地檔案存在，視您對檔案傳送指定的目的地檔案行為選項而定，會改寫該檔案。如果將目的地指定為目錄而非檔案，則目的地檔名是不包括高階限定元 (HLQ) 的資料集名稱。
- **PDS 來源資料集。** 每一個指定的成員或所有成員（如果未指定成員）會解壓縮至目的地。如果目的地指定目錄，便會將成員解壓縮至個別檔案。否則，每一個指定的成員會寫入目的地檔案，而導致實際上只傳送一個成員。如果成員的目的地檔案存在，視您對檔案傳送指定的目的地檔案行為選項而定，會改寫該檔案。

相關參考

第 530 頁的『傳送檔案的準則』

視傳送的來源及目的地作業系統以及是以二進位還是以文字模式進行傳送而定，對於預期發生的行為有一些準則可循。

第 532 頁的『在 z/OS 上的資料集之間傳送』

您可以使用 Managed File Transfer 在 z/OS 資料集之間進行傳送。請仔細檢閱下列行為，以確保正確傳送資料集。

第 367 頁的『fteCreateTransfer: 啟動新的檔案傳送』

fteCreateTransfer 指令從指令行建立並啟動新的檔案傳送。此指令可立即啟動檔案傳送、將檔案傳送排定在未來的時間和日期、將排定的傳送重複一次以上，以及根據特定條件觸發檔案傳送。

在 z/OS 上的資料集之間傳送

您可以使用 Managed File Transfer 在 z/OS 資料集之間進行傳送。請仔細檢閱下列行為，以確保正確傳送資料集。

Managed File Transfer 不支援磁碟或磁帶上未編定型錄的資料集。必須對現有資料集編定型錄，並且會對新資料集編定型錄。

請考量下列情況：

如果您在 z/OS 系統之間複製或移動資料集，且目的地不存在。

依預設，會建立目的地資料集且性質與來源相同。您可以為目的地資料集指定屬性，以置換預設性質。如果這樣指定，則會執行相容性檢查，以確保可以傳送。

如果您在 z/OS 系統之間複製或移動資料集，且目的地已存在。

- 如果為目的地資料集指定屬性以置換預設性質，則會執行相容性檢查，以確保能夠以想要的方式來存取目的地資料集。但是，不能置換下列屬性：
 - 基本資料集組織及類型
 - 邏輯記錄長度 (LRECL)

- 區塊大小 (BLKSIZE)

將資料集傳送至磁帶。

將資料集傳送至磁帶時，會取代磁帶上已經存在的任何現有資料集。新資料集的屬性將根據傳送定義中所傳遞的屬性進行設定。如果您沒有指定任何屬性，會將屬性設定為與來源資料集相同，或者，如果來源為檔案，則設定為預設值。現有磁帶資料集的屬性會被忽略。

此外，目的地代理程式執行的使用者 ID 需要具有裝載磁帶的正確權限。如需如何執行此動作的相關資訊，請參閱您企業所使用外部安全管理程式的說明文件。

如果您是從磁帶傳送至資料集。

為了存取磁帶上的資料集，來源代理程式執行的使用者 ID 必須具有裝載磁帶的適當權限。如需如何執行此動作的相關資訊，請參閱您企業所使用外部安全管理程式的說明文件。

資料集相容性

請檢閱資料集相容性的下列行為及限制：

記錄格式及長度差異：

- 可變格式記錄在記錄資料中使用 4 位元組記錄長度欄位。因此對於從固定記錄到可變記錄資料集的傳送，可變記錄長度必須大於或等於固定記錄長度加上 4。對於從可變格式記錄資料集到固定格式記錄資料集的傳送，固定格式記錄資料集記錄長度必須大於或等於可變記錄長度減 4。

區塊大小差異：

- 對於固定及可變格式記錄資料，區塊大小差異會使來源及目的地資料集佈置有所不同。
- 對於未定義的格式記錄，提供的目的地區塊大小會大於或等於來源資料集區塊大小，您可以傳送資料集。
- 對於未定義的格式資料集，如果來源區塊大小大於目的地區塊大小，則無法進行傳送。

分割的資料集 (PDS) 及分割的資料集延伸 (PDSE) 資料集

下列行為及限制也同樣適用於 PDS 及 PDSE：

- 如果將 PDS 或 PDSE 成員傳送至目的地 PDS 或 PDSE，則會建立目的地 PDS 或 PDSE 的成員。如果目的地 PDS 或 PDSE 成員已經存在，則會改寫成員。如果將 PDS 或 PDSE 成員傳送至非 PDS 或非 PDSE 成員目的地資料集，則會建立目的地資料集以包含成員資料。如果目的地資料集已經存在，則會改寫資料集。
- 如果嘗試將 PDS 或 PDSE 傳送至非 PDS 或非 PDSE 目的地，這會導致將 PDS 或 PDSE 的所有成員寫入非 PDSE 目的地。根據傳送選項，每個後續的成員傳送都會改寫非 PDSE 目的地先前的內容，或者傳送失敗。
- 將 PDS 或 PDSE 傳送至目的地 PDS 或 PDSE 時，會在目的地建立整個 PDS 或 PDSE 的副本。如果目的地 PDS 或 PDSE 成員已經存在，則會新增來自來源的成員。如果目的地已經存在 PDS 或 PDSE 成員，則會改寫成員。
- 將非 PDS 或非 PDSE 傳送至目的地 PDS 或 PDSE 時，會將非 PDS 或非 PDSE 的內容新增為 PDS 或 PDSE 的新成員。如果 PDS 成員已經存在，則會改寫成員。如果不為新成員指定名稱，則會從來源資料集產生名稱，或使用 DD 名稱。
- 在磁碟空間有限的系統上，傳送至 PDS 及 PDSE 資料集具有已知限制。如需詳細資料，請參閱 [一般 MFT 問題](#) 中的 z/OS 一節。
- **註：**將 PDS 或 PDSE 傳送至目的地 PDS 或 PDSE 時，不會保留成員資訊及統計資料。比方說，例如傳送儲存為 PDS 的載入程式庫，目的地 PDS 無法用作載入程式庫。

二進位及文字傳送

資料集的二進位傳送定義為採用其二進位格式的記錄資料，如同使用預設記錄格式（類型=記錄）從資料集進行讀取一樣。根據每筆記錄來讀取及寫入資料。系統服務會執行必要的記錄及區塊轉換（其中資料集具有不同的記錄及區域設定），以及必要的 ASA 與機器控制碼轉換。如果已為 ASA 格式控制字元定義一個資料集，但另一個不適用，則會使用 C/C++ 系統程式庫函數行為執行到一般控制碼的轉換。

世代資料群組 (GDG)

Managed File Transfer 支援對於 z/OS 上的來源及目的地資料集使用世代資料群組 (GDG)。支援絕對及相對 GDG 名稱。在您寫入新世代時，基本 GDG 必須已經存在。

相關參考

第 530 頁的『傳送檔案的準則』

視傳送的來源及目的地作業系統以及是以二進位還是以文字模式進行傳送而定，對於預期發生的行為有一些準則可循。

第 548 頁的『傳送世代資料群組 (GDG)』

Managed File Transfer 支援對於 z/OS 上的來源及目的地資料集使用世代資料群組 (GDG)。支援絕對及相對 GDG 名稱。在您寫入新世代時，基本 GDG 必須存在。

第 534 頁的『從 Connect:Direct 節點來回傳送資料集』

您可以使用 Connect:Direct 橋接器，在 Managed File Transfer 代理程式與 IBM Sterling Connect:Direct 節點之間傳送資料集。您可以將資料集指定為傳送來源及/或傳送目的地。

第 531 頁的『在 z/OS 與分散式系統之間傳送檔案及資料集』

您可以使用 Managed File Transfer，在 z/OS 與分散式檔案系統之間傳送檔案及支援的資料集類型。請仔細檢閱下列行為，視您在其中來回傳送的系統類型而定。

從 Connect:Direct 節點來回傳送資料集

您可以使用 Connect:Direct 橋接器，在 Managed File Transfer 代理程式與 IBM Sterling Connect:Direct 節點之間傳送資料集。您可以將資料集指定為傳送來源及/或傳送目的地。

指定資料集名稱

若要在傳送要求中指定 Connect:Direct 節點上的資料集，請使用在 Managed File Transfer 代理程式之間進行資料集傳送所用的語法，但要進行下列兩項變更：

- 您必須以 Connect:Direct 節點名稱和冒號 (:) 作為資料集名稱的字首。語法如下：

```
cdNode:data_set_name{;attrib1;...;attribN}
```

例如，若要於 Connect:Direct 節點 CD_NODE1 所在系統上，指定稱為 OBJECT.LIB 的已分割資料集，請使用下列語法：

```
CD_NODE1://'OBJECT.LIB';RECFM(F,B);BLKSIZE(800);LRECL(80)
```

在此範例中，文字 RECFM(F,B);BLKSIZE(800);LRECL(80) 指定了三個選用屬性。

- 指定的資料集名稱不論是否以單引號字元括住，都會解譯為完整資料集名稱。系統永不新增任何字首。如果要指定字首（例如，執行代理程式的使用者 ID），您必須將其指定為資料集名稱的一部分。這不同於僅涉及 Managed File Transfer 代理程式的資料集傳送行為，在後者中，如果指定的資料集名稱未以單引號字元括住，系統會將目的地代理程式的預設高階限定元新增為字首。

除了這兩項變更以外，請使用與在 Managed File Transfer 代理程式之間進行資料集傳送所用的相同語法，指定資料集名稱及任何選用屬性。該語法具有下列規則：

- 您必須以兩個正斜線字元 (//) 作為資料集名稱的字首。
- 如果要指定資料集屬性，請在資料集名稱之後提供這些屬性（以分號區隔）。必須以適合 BPXWDYN 的 *key(value)* 格式提供屬性。

如需在傳送要求中指定資料集的相關資訊，請參閱第 367 頁的『fteCreateTransfer: 啟動新的檔案傳送』及第 354 頁的『fteCreateTemplate: 建立新的檔案傳送範本』。

要在傳送要求中使用的參數

對於涉及 Connect:Direct 節點上資料集的大部分傳送要求，您可以按僅涉及 Managed File Transfer 代理程式的資料集傳送所用的相同方式，指定來源及目的地資料集。請將 **source_specification**、**-ds** 及 **-dp** 參數，與 **fteCreateTransfer** 或 **fteCreateTemplate** 指令搭配使用。下列實務支援此語法：

- 傳送中涉及的所有代理程式均為 IBM WebSphere MQ File Transfer Edition 7.0.4 或更新版本
- 來源代理程式為 Connect:Direct 橋接器代理程式，因此為 IBM WebSphere MQ File Transfer Edition 7.0.4 或更新版本，目的地代理程式為 IBM WebSphere MQ File Transfer Edition 7.0.3 或更舊版本

如果目的地代理程式為 Connect:Direct 橋接器代理程式，且來源代理程式為 IBM WebSphere MQ File Transfer Edition 7.0.3 或更舊版本，您必須對傳送要求進行下列變更：

- 若要將循序資料集或分割的資料集 (PDS) 成員指定為傳送目的地，請使用 **-df** 參數。
- 若要將 PDS 指定為傳送目的地，請使用 **-dd** 參數。

對於來源代理程式為 IBM WebSphere MQ File Transfer Edition 7.0.4 或更新版本的傳送，您也可以使用此語法作為一般 **-ds** 及 **-dp** 參數的替代方案。例如，如要在所有實務和涉及 IBM WebSphere MQ File Transfer Edition 7.0.3 或更舊版本之來源代理程式的部分實務中使用一致的語法，請使用 **-df** 及 **-dd** 參數。

註：如果傳送目的地為 PDS，且目的地代理程式為 Connect:Direct 橋接器代理程式，您必須指定值為 overwrite 的 **-de** 參數。

指定資料集屬性

某些資料集屬性是由 Managed File Transfer 設定，並作為參數傳遞至 Connect:Direct **COPY** 程序。您也可以透過指定適當的 BPXWDYN 索引鍵，在傳送要求中提供某些屬性。Connect:Direct 橋接器會將具有對等 Connect:Direct 內容的索引鍵，轉換為 Connect:Direct 所需的格式。例如，在資料集規格 `CD_NODE1: // 'OBJECT.LIB'; RECFM(F,B); BLKSIZE(800); LRECL(80)` 中，會將屬性 `RECFM(F,B); BLKSIZE(800); LRECL(80)` 轉換為 `DCB=(RECFM=FB, BLKSIZE=800, LRECL=80)`。

如需這兩種類型參數之間對映的詳細資料，包括支援與 Connect:Direct 傳送搭配使用的 BPXWDYN 索引鍵之詳細資料，請參閱第 536 頁的『Connect:Direct 程序陳述式參數與 BPXWDYN 鍵之間的對映』。並非所有 BPXWDYN 索引鍵都有對等的 Connect:Direct 程序參數，也並非所有 Connect:Direct 程序參數都有對等的 BPXWDYN 索引鍵。

其他考量


- 如果傳送目的地為 Connect:Direct 節點上分割的資料集，您必須在傳送之前建立分割的資料集，因為 Connect:Direct 節點不會為您建立該資料集。

相關概念

第 192 頁的『Connect:Direct 橋接器』

您可以從現有 IBM Sterling Connect:Direct 網路來回傳送檔案。使用 Connect:Direct 橋接器 (Managed File Transfer 元件)，可在 MFT 與 IBM Sterling Connect:Direct 之間傳送檔案。

相關工作

 將資料集傳送至 z/OS 上的 Connect:Direct 節點

您可以使用位於 Windows 或 Linux 系統上的 Connect:Direct 橋接器，從 z/OS 上的 Managed File Transfer 代理程式，將資料集傳送至 z/OS 上的 Connect:Direct 節點。

相關參考

第 532 頁的『在 z/OS 上的資料集之間傳送』

您可以使用 Managed File Transfer 在 z/OS 資料集之間進行傳送。請仔細檢閱下列行為，以確保正確傳送資料集。

第 367 頁的『fteCreateTransfer: 啟動新的檔案傳送』

fteCreateTransfer 指令從指令行建立並啟動新的檔案傳送。此指令可立即啟動檔案傳送、將檔案傳送排定在未來的時間和日期、將排定的傳送重複一次以上，以及根據特定條件觸發檔案傳送。

第 354 頁的『fteCreateTemplate: 建立新的檔案傳送範本』

fteCreateTemplate 指令可建立檔案傳送範本，以備將來使用。唯一必要的參數是 **-tn** (*template_name*) 參數。其他所有參數是選用的，雖然如果指定來源檔案規格，您還必須提供目的地檔案。同樣地，如果指定目的地檔案，您還必須提供來源檔案規格。

第 291 頁的『指定有雙正斜線的 Connect:Direct 檔案路徑』

在檔案傳送過程中，如果您使用以雙正斜線 (//) 開頭的檔案路徑，來指定位於 Connect:Direct 節點上的檔案，則會將該檔案視為資料集。

Connect:Direct 程序陳述式參數與 BPXWDYN 鍵之間的對映

當您針對來源或目的地為 Connect:Direct 節點的資料集提交傳送要求時，您提供的任何受支援的 BPXWDYN 鍵，均會轉換為 Connect:Direct 程序接受的格式。

如需 IBM Sterling Connect:Direct 處理程序陳述式的相關資訊，請參閱 [Connect:Direct Process Language Reference Guide](#)。

Connect:Direct COPY 陳述式的參數	BPXWDYN 鍵
DSN	DSN（適用於從資料集來回傳送）指定此鍵會置換 Managed File Transfer 指派之參數值，該參數值是以傳送要求中提供的來源檔案或目的地檔案規格為基礎。
檔案	無對映（針對資料集）。
PNODE	無對映。傳送的主要節點由 Managed File Transfer 識別。如果您嘗試為此參數提供值，則會產生錯誤。
SNODE	無對映。傳送的次要節點由 Managed File Transfer 識別。如果您嘗試為此參數提供值，則會產生錯誤。
DCB	請參閱 DCB 子參數的對映
DISP	請參閱 COPY From 陳述式 DISP 子參數的對映 ，以及 COPY To 陳述式 DISP 子參數的對映
RESGDG	無對映
LABEL	請參閱 LABEL 子參數的對映
MSVGP	無對映
UNIT	UNIT
VOL	請參閱 VOL 子參數的對映
ALIAS	無對映
EXCLUDE	無對映
PDS.DIR	無對映。Managed File Transfer 將此處理程序參數的值設為 N，因此不會傳送目錄中的任何使用者相關資訊。
REPLACE NOREPLACE	無 BPXWDYN 對等項目。目的地系統中已經存在目的地資料集時的行為，是由傳送要求中 -de (destination_file_behavior) 參數的值來定義。如需目的地資料集已經存在時的 Managed File Transfer 預設行為相關資訊，請參閱第 532 頁的『 在 z/OS 上的資料集之間傳送 』。
SELECT	無 BPXWDYN 對等項目。所選要複製的資料集成員，是由傳送要求中的來源檔案規格定義。
BUFND	無對映
IOEXIT	無對映
DATAEXIT	無對映
SYSOPTS	請參閱 SYSOPTS 子參數的對映
類型	無對映
AVGREC	無對映

表 58: Connect:Direct COPY 陳述式的參數，以及 Managed File Transfer 使用的對等 BPXWDYN 鍵 (繼續)	
Connect:Direct COPY 陳述式的參數	BPXWDYN 鍵
DATACLAS	DATACLAS
DSNTYPE	DSNTYPE。為此鍵指定 PDS 值，會置換 Managed File Transfer 指派的參數值（即 LIBRARY）。所有其他值（EXTPREF、EXTREQ、BASIC 或 LARGE）皆沒有對映。指定這些不受支援的任何值，皆會產生錯誤。為循序資料集指定 PDS 或 LIBRARY，會產生錯誤。
KEYLEN	無對映
KEYOFF	無對映
類似	類似
LRECL	無對映
MGMTCLAS	MGMTCLAS
RECORG	無對映
SECMODEL	無對映
STORCLAS	STORCLAS
SPACE	請參閱 SPACE 子參數的對映
SYSOUT	無對映
CKPT	無對映
COMPRESS	無對映
SECURE	無對映

表 59: Connect:Direct COPY 陳述式 DCB 參數的子參數，以及 Managed File Transfer 使用的對等 BPXWDYN 鍵	
DCB 參數的子參數	BPXWDYN 鍵
model-file-name	無對映
BLKSIZE	BLKSIZE
NCP	BUFNO
DEN	無對映
DSORG	DSORG
KEYLEN	無對映
LIMCT	無對映
LRECL	LRECL
OPTCD	無對映
RECFM	RECFM
RKP	無對映
TRTCH	TRTCH

表 60: Connect:Direct **COPY From** 陳述式 **DISP** 參數的子參數，以及 Managed File Transfer 使用的對等 **BPXWDYN** 鍵

COPY From 陳述式之 DISP 參數的子參數	BPXWDYN 鍵	詳細資料
[OLD SHR]	[OLD SHR]	指定資料集在傳送之前的狀態。Managed File Transfer 會將此子參數設為 SHR 。
[KEEP DELETE]	[KEEP DELETE] 或 PATHDISP	指定在順利完成傳送之後的資料集狀態。Managed File Transfer 所設定的值，取決於 -sd 參數所定義的來源檔案處置方式。
[KEEP DELETE]	[KEEP DELETE] 或 PATHDISP	指定在異常完成傳送之後的資料集狀態。Managed File Transfer 會將此子參數設為 KEEP 。

表 61: Connect:Direct **COPY To** 陳述式 **DISP** 參數的子參數，以及 Managed File Transfer 使用的對等 **BPXWDYN** 鍵

COPY To 陳述式之 DISP 參數的子參數	BPXWDYN 鍵	詳細資料
[NEW OLD MOD RPL SHR]	[NEW OLD MOD SHR]	指定資料集在傳送之前的狀態。Managed File Transfer 所設定的值，取決於傳送要求中 -de (destination_file_behavior) 參數的值。如果目的地資料集尚不存在，此子參數值將為 NEW 。如果該資料集已經存在，此子參數值將為 RPL 。Managed File Transfer 不支援在傳送要求中提供 RPL 鍵。
[KEEP CATLG]	[KEEP CATLOG] 或 PATHDISP	指定在順利完成傳送之後的資料集狀態。Managed File Transfer 會將此子參數設定為 CATLOG 。
[KEEP CATLG DELETE]	[KEEP DELETE] 或 PATHDISP	指定在異常完成傳送之後的資料集狀態。Managed File Transfer 會將此子參數設定為 DELETE 。

表 62: Connect:Direct **COPY** 陳述式 **LABEL** 參數的子參數，以及 Managed File Transfer 使用的對等 **BPXWDYN** 鍵

COPY 陳述式之 LABEL 參數的子參數	BPXWDYN 鍵	詳細資料
file-sequence-number	SEQUENCE	
[SL AL BLP LTM NL]	LABEL (type)	可能的 type 值為 NL、SL、NSL、SUL、BLP、LTM、AL 及 AUL。Connect:Direct 接受這些值的子集。如果您指定 Connect:Direct 不支援的值，Connect:Direct 會產生錯誤訊息。
[PASSWORD NOPWREAD]	無對映	
[IN OUT]	無對映	
[RETPD EXPDT]	RETPD	不支援 EXPDT

COPY 陳述式之 VOL 參數的子參數	BPXWDYN 鍵
私人	無對映
RETAIN	無對映
volume-sequence-no	無對映
volume-count	MAXVOL
SER	VOL
REF	無對映

COPY 陳述式之 SYSOPTS 參數的子參數	BPXWDYN 鍵
DBCS	無對映
CODEPAGE	此值取決於 Managed File Transfer 傳送選項。如需相關資訊，請參閱第 540 頁的『使用 MFT 傳送文字檔』。
資料類型	無對映. Managed File Transfer 會針對往返於資料集的文字傳送，將此值設為 TEXT，否則會將此值設為 BINARY。
XLATE	無對映. Managed File Transfer 當 DATATYPE 的值為 TEXT 時，會將此值設為 NO。
STRIP.BLANKS	無對映. Managed File Transfer 當 DATATYPE 的值為 TEXT 時，會將此值設為 YES。
PERMISS	無對映
PRECOMP	無對映
UNIQUE	無對映
SYSOUT	無對映

COPY 陳述式之 SPACE 參數的子參數	BPXWDYN 鍵
CYL	CYL
TRK	TRACKS
blk	BLOCKS
av-rec-len	無對映
prim、[sec]、[dir]	SPACE(prim[,sec])、DIR
RLSE	RELEASE
CONTIG	無對映
ROUND	無對映

相關概念

第 192 頁的『Connect:Direct 橋接器』

您可以從現有 IBM Sterling Connect:Direct 網路來回傳送檔案。使用 Connect:Direct 橋接器（Managed File Transfer 元件），可在 MFT 與 IBM Sterling Connect:Direct 之間傳送檔案。

相關工作

z/OS 將資料集傳送至 z/OS 上的 Connect:Direct 節點

您可以使用位於 Windows 或 Linux 系統上的 Connect:Direct 橋接器，從 z/OS 上的 Managed File Transfer 代理程式，將資料集傳送至 z/OS 上的 Connect:Direct 節點。

相關參考

與 Connect:Direct 節點來回傳送資料集

您可以使用 Connect:Direct 橋接器，在 Managed File Transfer 代理程式與 IBM Sterling Connect:Direct 節點之間傳送資料集。您可以將資料集指定為傳送來源及/或傳送目的地。

z/OS 不得與 MFT 一起使用的 BPXWDYN 內容

使用 `fteCreateTemplate` 指令、`fteCreateTransfer` 指令或 `agent.properties` 檔中的 `bpxwdynAllocAdditionalOptions` 內容時，不得指定某些 BPXWDYN 選項。

有許多不得與 Managed File Transfer 一起指定的 BPXWDYN 選項，因為這些選項由代理程式使用或不受支援。如果您使用這些選項，它們可能會導致無法預期的行為；下表列出了這些選項。

BPXWDYN 選項	說明
DA DSN	指定要配置的資料集名稱。
FI DD	指定要配置的 ddname。
FILEDATA	對循序存取方法服務指定資料是視為文字還是二進位。
OLD SHR MOD NEW SYSOUT	指定資料集狀態。
REUSE	指定在執行功能之前，釋放具名資料集。
HOLD	指定將保留輸出資料集，直至使用者或操作員釋放資料集。
KEEP DELETE CATALOG UNCATALOG	指定在釋放資料集之後資料集的處置方式。
RECORG(LS)	建立 VSAM 線性資料集。
MSG	導向配置訊息。 註：您可以使用此選項，但由於 Managed File Transfer 使用此選項來將錯誤資訊導向傳送日誌，因此使用此選項可能會導致無法預期的行為。

使用 MFT 傳送文字檔

文字檔傳送涉及將檔案的字碼頁從一種字碼頁轉換成另一種字碼頁。文字檔傳送也涉及在系統之間轉換 CRLF（歸位-換行）字元。本主題彙總 Managed File Transfer 的文字檔傳送行為。

除非您另有指定，否則會從檔案的來源系統的預設字碼頁轉換成其目的地系統的預設字碼頁。此外，文字檔傳送會執行換行轉換，這表示目的地檔案的換行字元是其目的地平台的原生換行字元。您可以指定用來讀取來源檔案及寫入目的地檔案的字碼頁，來置換系統上使用的預設字碼頁。您也可以指定用於目的地檔案的行尾字元序列。如需相關資訊，請參閱第 367 頁的『`fteCreateTransfer`：啟動新的檔案傳送』和第 133 頁的『使用傳送定義檔』主題。

文字檔傳輸會在字碼頁之間執行簡單的字碼點替換。文字檔傳輸不會執行複雜的傳輸或資料轉換，例如在雙向資料或文字造形的顯示與邏輯格式之間轉換。

區域	預設行為	您能否變更此行為?
來源檔案編碼	來源平台編碼	是 當您指定來源檔案編碼且來源是資料集時，編碼必須是 EBCDIC 字碼頁，否則傳送失敗。同樣地，如果目的地是資料集，目的地編碼必須是 EBCDIC 字碼頁。
來源檔案行尾字元序列	將單一 (LF) 或 (CRLF) 序列轉換成目的地的行尾字元序列	否
目的地檔案編碼	目的地平台編碼	是 當您指定來源檔案編碼且來源是資料集時，編碼必須是 EBCDIC 字碼頁，否則傳送失敗。同樣地，如果目的地是資料集，目的地編碼必須是 EBCDIC 字碼頁。
目的地檔案行尾字元序列	目的地平台 EOL	是
來源或目的地中未對映的或形態異常的字元的文字取代字元序列	空白，表示如果有未對映的字元或形態異常的字元，則傳送失敗。您可以使用 <code>textReplacementCharacterSequence</code> 內容來指定取代文字，其說明於第 463 頁的『MFT <code>agent.properties</code> 檔案』中。	是

z/OS 資料集

z/OS

以文字模式存取資料集記錄時，每一個記錄代表一行。記錄中不會有換行字元，但以 ASA 格式資料集而言，設定了 ASA 格式控制碼字元來代表換行字元（或其他控制字元）。當含有終止的換行字元的一行文字寫入記錄時，會自動移除換行字元，或適時設定適當的 ASA 控制碼。讀取記錄時，換行字元會自動附加到傳回的資料。以 ASA 格式資料集而言，如果適合記錄的 ASA 控制碼，此字元可以是多次換行或換頁。

此外，以固定格式資料集而言，在讀取記錄時，會在記錄最後一個字元（但非空格字元）之後附加換行字元，使固定格式資料集適合儲存文字。

區域	預設行為	您能否變更此行為?
最大行長度	適當的目的地資料集 LRECL 或 BLKSIZE 設定	否
折返長度行	折返。如有需要，此行可分割成多個記錄和區塊。	否

當 Managed File Transfer 代理程式執行時，環境變數 `_EDC_ZERO_RECLLEN` 一律設為 "Y"。此設定使 Managed File Transfer 文字傳送行為與 FTP 對變數和固定區塊資料集的傳送行為相同。不過，以未定義的格式資料集而言，Managed File Transfer 會將單一空格行轉換成空白行，並保留空白行。FTP 將空白行轉換成單一空格行，並保留單一空格行。表 3 說明 Managed File Transfer 行為，以及 FTP 行為與其不同之處。

資料集的格式也決定每一文字行如何寫入記錄中。以非 ASA 格式資料集而言，換行及歸位字元不會寫入記錄中。以 ASA 格式資料集而言，每一個記錄的第一個位元組是代表行尾或換頁的 ASA 控制碼，以及其他適當的代碼。因為 ASA 控制碼位於每一個記錄的開頭，如果來源文字檔的開頭不是換行字元序列，則會插入空白 (' ') ASA 控制字元序列（相等於新增一行）。這表示如果 ASA 資料集傳送到檔案，空白行會出現在檔案開頭。

資料集格式	檔案中的原始文字行	資料集記錄	讀取資料集記錄	FTP 讀取行為
已修正區塊	空白行	填入空格的記錄	空白行	與 MFT 相同
已修正區塊	單一空格	填入空格的記錄	空白行	與 MFT 相同
可變區塊	空白行	空白記錄	空白行	與 MFT 相同
可變區塊	單一空格	單一空格記錄	單一空格	與 MFT 相同
未定義	空白行	單一空格記錄	空白行	單一空格
未定義	單一空格	單一空格記錄	空白行	單一空格

相關參考

第 530 頁的『傳送檔案的準則』

視傳送的來源及目的地作業系統以及是以二進位還是以文字模式進行傳送而定，對於預期發生的行為有一些準則可循。

第 542 頁的『在 Connect:Direct 與 MFT 之間傳送文字檔』

文字傳送涉及將檔案的字碼頁從一種字碼頁轉換為另一種字碼頁。文字檔傳送還會涉及在系統之間轉換 CRLF（換行，carriage return line feed）字元。本主題彙總 Managed File Transfer Agent 與 Connect:Direct 節點之間傳送時的文字檔傳送行為。


第 575 頁的『MFT 的可用字碼頁』

本參考主題列出了適用於在 Managed File Transfer 支援的各種平台上，進行文字檔轉換的所有字元編碼格式。

在 Connect:Direct 與 MFT 之間傳送文字檔

文字傳送涉及將檔案的字碼頁從一種字碼頁轉換為另一種字碼頁。文字檔傳送還會涉及在系統之間轉換 CRLF（換行，carriage return line feed）字元。本主題彙總 Managed File Transfer Agent 與 Connect:Direct 節點之間傳送時的文字檔傳送行為。

如需 Managed File Transfer 中文字傳送行為的相關資訊，請參閱第 540 頁的『使用 MFT 傳送文字檔』。

- 確保 Connect:Direct 橋接器節點的網路對映，以及用作傳送目的地之任何 Connect:Direct 節點的網路對映，都包含正確的平台說明。
 - 如果 Connect:Direct 橋接器節點是在 Windows 系統上，請務必針對網路對映中的每個遠端節點，從**作業系統**清單中選擇正確的值。
 - 如果遠端節點是在 Windows 系統上，請選取 Windows。
 - 如果遠端節點是在 UNIX 或 Linux 系統上，請選取 UNIX。
 -  如果遠端節點是在 z/OS 系統上，請選取 OS/390。

Connect:Direct 橋接器不支援傳送至其他作業系統上的遠端節點。

- 請確定將檔案傳送至或從中的每一個遠端節點，在 Connect:Direct 橋接器代理程式配置目錄中，指定 ConnectDirectNodeProperties.xml 檔案中遠端 Connect:Direct 節點的作業系統類型。如需相關資訊，請參閱第 120 頁的『配置 ConnectDirectNodeProperties.xml 檔案以包含遠端 Connect:Direct 節點的相關資訊』及第 494 頁的『Connect:Direct 節點內容檔格式』。

Connect:Direct 會使用此網路對映資訊來決定要使用的行尾結束符號。

- 如果傳送目的地為 Managed File Transfer Agent，此代理程式會執行行尾結束字元轉換。

- 如果傳送目的地為 Connect:Direct 節點，Connect:Direct 橋接器代理程式會執行行尾結束字元轉換。

相關參考

第 540 頁的『使用 MFT 傳送文字檔』

文字檔傳送涉及將檔案的字碼頁從一種字碼頁轉換成另一種字碼頁。文字檔傳送也涉及在系統之間轉換 CRLF（歸位-換行）字元。本主題彙總 Managed File Transfer 的文字檔傳送行為。

第 290 頁的『傳入或傳出 Connect:Direct 節點的文字傳送未正確轉換資料時要執行的作業』

當您以文字模式在 MFT 代理程式與 Connect:Direct 節點之間傳送檔案時，將會執行字碼頁及行尾字元轉換。該傳送會使用 Connect:Direct 橋接器節點的網路對映中的作業系統資訊，來判斷遠端節點的行尾字元。如果網路對映中的資訊不正確，可能不會正確執行行尾字元轉換。

與通訊協定橋接器代理程式來回傳送檔案

您可以使用通訊協定橋接器代理程式，從 Managed File Transfer 網路外部的 FTP 或 SFTP 檔案伺服器來回傳送檔案。

使用通訊協定橋接器傳送檔案時，橋接器必須有權讀取您要傳送的檔案所在的來源或目的地目錄。比方說，例如您要從僅具有執行權限 (d--x--x--x) 的目錄 /home/fte/bridge 傳送檔案，您嘗試從此目錄進行的任何傳送都會失敗，且會顯示下列錯誤訊息：

```
BFGBR0032E: Attempt to read filename from the protocol file server has failed with server error 550  
Failed to open file.
```

在檔案傳送期間，檔案通常以暫存檔形式寫入目的地，然後在傳送完成時重新命名。但是，如果傳送目的地是配置為限制寫入的通訊協定檔案伺服器（使用者可以將檔案上傳至通訊協定檔案伺服器，但無法以任何方式變更這些已上傳的檔案；實際上，使用者只能寫入一次），已傳送的檔案將直接寫入目的地。這意味著，如果在傳送期間發生問題，已寫入一部分的檔案仍然在目的地通訊協定檔案伺服器上，Managed File Transfer 無法刪除或編輯這些檔案。在此狀況下，傳送會失敗。

除了通訊協定橋接器代理程式之外，請確定您在 Managed File Transfer 網路中還有另一個代理程式。通訊協定橋接器代理程式只是 FTP 或 SFTP 伺服器的橋接器，不會將已傳送的檔案寫入本端磁碟。如果要將檔案傳送至 FTP 或 SFTP 伺服器或從這些伺服器傳送檔案，您必須使用通訊協定橋接器代理程式作為檔案傳送的目的地或來源（代表 FTP 或 SFTP 伺服器），及將其他標準代理程式作為對應的來源或目的地。

需要在 SFTP 檔案伺服器上建立新目錄的受管理傳送要求

Managed File Transfer 通訊協定橋接器代理程式使用協力廠商 JSch 程式庫，透過 SFTP 通訊協定與檔案伺服器進行通訊。如果通訊協定橋接器代理程式嘗試將檔案傳送至檔案伺服器上不存在的目錄，且 JSch 無法執行所要求的 SFTP 作業來建立該目錄，因為通訊協定橋接器代理程式登入檔案伺服器時所使用的使用者無權這樣做，JSch 會將異常狀況傳回給通訊協定橋接器代理程式。然後，通訊協定橋接器代理程式會將受管理傳送標示為「失敗」，並產生增補訊息。如果 JSch 已提供失敗的相關資訊，則通訊協定橋接器代理程式會在增補訊息中包含此資訊：

```
BFGTR0072E: 傳送無法完成，因為發生異常狀況：
```

```
BFGBR0119E: Bridge 代理程式無法建立目錄 directory name，因為來自 JSch 異常狀況的訊息
```

V 9.0.0.9 從 IBM MQ 9.0.0 Fix Pack 9 開始，如果 JSch 異常狀況不包含任何失敗的相關資訊，則通訊協定橋接器代理程式會產生下列增補訊息：

```
BFGTR0072E: 傳送無法完成，因為發生異常狀況：
```

```
BFGBR0209E: 橋接器代理程式無法建立目錄 directory name
```

相關概念

第 178 頁的『通訊協定橋接器』

通訊協定橋接器可供 Managed File Transfer (MFT) 網路存取儲存在 MFT 網路之外的檔案伺服器上的檔案（在本端網域或遠端位置）。此檔案伺服器可使用 FTP、FTPS 或 SFTP 網路通訊協定。每一個檔案伺服器需要至少一個專用的代理程式。專用代理程式即所謂的通訊協定橋接器代理程式。橋接器代理程式可以與多個檔案伺服器互動。

IBM i 與 IBM i 系統來回傳送檔案

如果您在文字模式下使用 Managed File Transfer 與 IBM i 系統來回傳送檔案，並且想要轉換檔案中的資料，則請考量本主題中的資訊。

IBM i 系統上的每個檔案都標有編碼字集 ID (CCSID) 值標籤，該值用於識別檔案的資料編碼。例如，包含 EBCDIC 資料的檔案可能具有 CCSID 值 037，包含 ASCII 資料的檔案可能具有 CCSID 值 819。

對於文字模式傳送，Managed File Transfer 會在來源及目的地檔案的檔案編碼不同時轉換資料。但是，Managed File Transfer 目前會忽略與 IBM i 系統上檔案相關聯的 CCSID 標籤。相反的，它會使用執行來源代理程式及目的地代理程式的 JVM 的 JVM 檔案編碼內容。此內容的預設值基於語言環境 (但您可以使用下列小節中說明的 `SystemDefault.properties` 檔案來置換 IBM i 系統上的此預設值:第 544 頁的『變更 `SystemDefault.properties` 檔中的 `file.encoding` 記錄』)。藉由此預設實作，以文字模式傳送檔案的代理程式的功能會限制為處理具有不同檔案編碼的文字檔。例如，在不停止代理程式並以適當的 (即 EBCDIC 或 ASCII) 檔案編碼就地置換後重新啟動的情況下，您不能使用同一代理程式既傳送包含 EBCDIC 文字的檔案，又傳送包含 ASCII 文字的檔案。在 IBM i V6R1 系統上，您可以使用 `WRKJVMJOB` 選項 7「顯示現行 Java 系統內容」，來查看執行此代理程式工作的 JVM 的檔案編碼值。(WRKJVMJOB 指令不存在於 IBM i V5R4 系統上。)

如果您計劃使用 Managed File Transfer 來傳送具有不同檔案編碼的文字檔，則請考量建立多個代理程式及啟動這些代理程式的多位使用者，以便針對每種唯一的編碼，備妥並啟用相應的代理程式來傳送該資料類型。

比方說，例如您想要將包含 EBCDIC 文字且 CCSID 值為 037 的檔案，從 IBM i 系統 (來源) 傳送至另一個 IBM i V6R1 系統 (目的地)，並且想要將目的地的檔案內容轉換為 ASCII 文字且 CCSID 值為 819，請完成下列步驟：

1. 選取 JVM 檔案編碼為 Cp037 的來源代理程式。
2. 選取 JVM 檔案編碼為 ISO8859_1 的目的地代理程式。
3. 選取文字模式傳送，並依需要選取其他規格。

變更 `SystemDefault.properties` 檔中的 `file.encoding` 記錄

若要針對特定編碼啟用執行代理程式的 JVM，請完成下列步驟：

1. 決定由哪位使用者啟動在 IBM i 系統上執行的代理程式。此代理程式用於處理 Managed File Transfer 檔案傳送要求。

依需要在該使用者的起始目錄中建立 `SystemDefault.properties` 檔案。比方說，例如您要啟動代理程式，請使用 Qshell 執行以下指令：

```
touch -C 819 /home/your_userID/SystemDefault.properties
```

2. 使用 Qshell，視需要執行 `/qibm/proddata/mqm/bin/fteStopAgent` 指令來停止代理程式。
3. 更新步驟 1 中所說明的 `SystemDefault.properties` 檔案，確保該檔案中包含如下所示的記錄：

```
file.encoding=java_encoding
```

其中 `java encoding` 對應於檔案中所包含的資料類型，並符合下表中的 `file.encoding` 值：
[File.encoding values and System i5 CCSID](#)。

4. 步驟 1 中所識別的使用者必須完成下列步驟：

- a. 僅限在 IBM i V5R4 上：將 `QIBM_PASE_DESCRIPTOR_STDIO` 環境變數 (*JOB 範圍) 新增至 'B' (如果使用 EBCDIC 檔案編碼) 或 'T' (如果使用 ASCII 編碼)。例如：

```
ADDENVVAR ENVVAR('QIBM_PASE_DESCRIPTOR_STDIO') VALUE('B') REPLACE(*YES)
```

- b. 如果 Qshell 處於作用中狀態，請按 **F3=Exit** 以結束 Qshell。
- c. 啟動 Qshell 並適當地執行 `/qibm/proddata/mqm/bin/fteStartAgent` 指令，以重新啟動代理程式。

當執行代理程式的 JVM 的檔案編碼變更後，就會使用該編碼寫入代理程式日誌。如果您想要閱讀該代理程式日誌的內容，必須使用已針對該編碼啟用的檢視器。

針對資料轉換使用傳送定義

在傳送檔案時轉換資料的替代方式是建立指定檔案編碼的傳送定義，或者使用 `fteCreateTransfer` 指令的 `-sce` 及 `-dce` 參數。當目的地為 IBM i 系統時，如果您使用這些參數，則會導致檔案所具有的 CCSID 標籤不正確。因此，在檔案位於 IBM i 系統的情況下，控制資料轉換的建議方法是使用前述部分所說明的 `SystemDefault.properties`。

通訊協定橋接器限制

在 IBM i 上，您不能使用通訊協定橋接器代理程式對 SFTP 伺服器來回傳送 EBCDIC 檔案。

相關參考

第 530 頁的『傳送檔案的準則』

視傳送的來源及目的地作業系統以及是以二進位還是以文字模式進行傳送而定，對於預期發生的行為有一些準則可循。

第 547 頁的『傳送位於 IBM i 上 QSYS.LIB 中的儲存檔』

Managed File Transfer 支援在兩個 IBM i 系統之間傳送位於 QSYS.LIB 檔案系統中的儲存檔。要求對儲存檔進行檔案傳送時，請考量下列資訊。

相關資訊

在 IBM i 上安裝 IBM MQ 伺服器

IBM i 傳送位於 IBM i 上 QSYS.LIB 中的實體檔

Managed File Transfer 支援在兩個 IBM i 系統之間傳送位於 QSYS.LIB 檔案系統中的實體檔案成員。當您需要對實體檔案成員進行檔案傳送時，請考量下列資訊。

IBM i 上的實體檔案成員位於實體檔中，而該檔案位於 IBM i 的程式庫中。程式庫可以是作業系統隨附的其中一個標準程式庫（例如，QSYS 或 QGPL），也可以是您建立的程式庫。

IBM i 上有兩種不同方式可識別 QSYS.LIB 檔案系統中的實體檔。當您在 IBM i 指令行上執行 CL 指令時，請使用下列命名語法：

```
FILE(library name/file name) MBR(member name)
```

例如，位於 SOMELIB 程式庫之 MYFILE 檔案中名為 MYMBR 的實體檔案成員，將被識別為 FILE(SOMELIB/MYFILE) MBR(MYMBR)。您也可以指定遵循「整合式檔案系統 (IFS)」命名慣例的 UNIX 形式路徑名稱，以識別相同的實體檔案成員。使用 IFS 命名慣例時，SOMELIB 中 MYFILE 的 MYMBR 具有下列路徑名稱：

```
/QSYS.LIB/SOMELIB.LIB/MYFILE.FILE/MYMBR.MBR
```

如需相關資訊，請參閱 [QSYS.LIB 檔案系統中的路徑名稱](#)。

IBM i 上的 Managed File Transfer 可辨識 IFS 命名慣例，但不支援 CL 指令使用的語法。下列範例說明 MFT 的有效及無效路徑名稱。下列範例是實體檔案成員的有效路徑名稱：

```
/QSYS.LIB/SOMELIB.LIB/MYFILE.FILE/MYMBR.MBR
```

此範例假設 MYFILE 是程式庫 SOMELIB 中的實體檔，而且含有名為 MYMBR 的成員。

下列範例是實體檔案成員傳送的無效路徑名稱：

- /QSYS.LIB/SOMELIB.LIB/MYFILE.FILE (.FILE 假設為 SAVF，而不是實體檔。如果 MYFILE 是實體檔，則傳送會失敗，並產生無效檔案類型錯誤)
- /QSYS.LIB/MYLIB.LIB/ (需要實體檔及成員名稱)
- /QSYS.LIB/SOMELIB.LIB/MYFILE.FILE/MYMBR (成員名稱必須包含副檔名 .MBR)
- /QSYS.LIB/SOMELIB.LIB/MYFILE/MYMBR.MBR (實體檔名稱副檔名必須是 .FILE)

在單一傳送要求中從實體檔傳送多個實體檔案成員

IBM i 上的 Managed File Transfer 支援以單一傳送要求從單一實體檔傳送多個實體檔案成員。您可以指定包括萬用字元的適當路徑名稱，如下列範例所示：

- ABCLIB 包含具有多個成員的實體檔 MYFILE。若要在單一要求中傳送所有成員，請指定下列路徑名稱：/QSYS.LIB/ABCLIB.LIB/MYFILE.FILE/*.*
- XYZLIB 包含實體檔 MYFILE，其成員名稱以單一字元不同，即：TEST1.MBR，TEST2.MBR。TEST3.MBR，依此類推。若要在單一要求中傳送所有成員，請指定下列路徑名稱：/QSYS.LIB/XYZLIB.LIB/MYFILE.FILE/TEST?.MBR。

下列類型的傳送要求不支援傳送多個實體檔案成員，而會導致錯誤：

- /QSYS.LIB/MYLIB.LIB/*.*
- /QSYS.LIB/MYLIB.LIB/*
- /QSYS.LIB/MYLIB.LIB/*.FILE/MYMBR.MBR
- /QSYS.LIB/MYLIB.LIB/MYFILE*.FILE/*.* (檔案名稱不支援使用萬用字元，只有成員名稱支援)
- /QSYS.LIB/MYLIB.LIB/*.FILE/*.*
- /QSYS.LIB/MYLIB.LIB/MYFILE.FILE (.FILE 假設為 SAVF，而不是實體檔。因此，如果 MYFILE 是實體檔，則傳送會失敗，並產生無效檔案類型錯誤)

與非 IBM i 系統來回傳送實體檔案成員

MFT 支援在非 IBM i 系統（例如 UNIX, Linux, and Windows）中來回傳送實體檔案成員。所有傳送都必須在文字模式中完成。下列範例說明使用非 IBM i 系統時支援的一些 **fteCreateTransfer** 要求：

- 此指令會將 IBM i 上的實體檔案成員 FILE(FROMIBMI/FILE1) MBR(FILE1) 傳送到 Linux 上的文字檔 /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
```

- 此指令會將 IBM i 上的實體檔案成員 FILE(FROMIBMI/FILE1) MBR(FILE1) 傳送到 Windows 上的文字檔 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
```

- 此指令會將 Windows 上的文字檔 C:\FTE\toibmi\file.txt 傳送到 IBM i 上的實體檔案成員 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
```

下列指令是對非 IBM i 系統的無效實體檔案成員傳送的範例：

- 因為 Windows 上的來源檔具有 .txt 副檔名，但是目的地目錄已指定 .file，所以此指令失敗。使用目的地目錄參數來指定目的地實體檔進行傳送時，來源檔副檔名必須是 .mbr 檔案（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
```

- 預設傳送模式為二進位，但是傳送實體檔案成員時必須指定文字模式。

```
fteCreateTransfer -da windows -dm QM1 -sa ibmi -sm QM1 -df C:\FTE\fromibmi\file.bin /qsys.lib/
fromibmi.lib/file1.file/file1.mbr
```

MFT 支援傳送位於 QSYS.LIB 檔案系統中的實體檔案成員，但不支援傳送位於 QSYS.LIB 檔案系統中的來源實體檔案成員。在 QDLS 檔案系統中，支援使用提供的範例使用者結束程式來進行檔案傳送。您可以將 MFT 中提供的範例使用者結束程式用於下列作業：

- 在 QDLS 檔案系統中傳送檔案。
- 使用與 MFT 檔案監視器相同的方式，自動從 IBM i 程式庫傳送實體檔案成員。
- 如果在傳送過程中刪除來源檔案成員，則刪除空檔案物件。

如需相關資訊，請參閱 [第 219 頁的『IBM i 上的 MFT 範例使用者結束程式』](#)。

相關參考

[第 530 頁的『傳送檔案的準則』](#)

視傳送的來源及目的地作業系統以及是以二進位還是以文字模式進行傳送而定，對於預期發生的行為有一些準則可循。

[第 543 頁的『與 IBM i 系統來回傳送檔案』](#)

如果您在文字模式下使用 Managed File Transfer 與 IBM i 系統來回傳送檔案，並且想要轉換檔案中的資料，則請考量本主題中的資訊。

IBM i 傳送位於 IBM i 上 QSYS.LIB 中的儲存檔

Managed File Transfer 支援在兩個 IBM i 系統之間傳送位於 QSYS.LIB 檔案系統中的儲存檔。要求對儲存檔進行檔案傳送時，請考量下列資訊。

IBM i 上的儲存檔位於 IBM i 上的程式庫。程式庫可以是作業系統隨附的其中一個標準程式庫，例如 QSYS 或 QGPL，也可以是使用者建立的程式庫。IBM i 上有兩種不同方式可識別 QSYS.LIB 檔案系統中的儲存檔。在 IBM i 指令行上使用 CL 指令時，使用的命名語法如下所示：

```
FILE(library name/file name)
```

例如，稱為 MYSAVF 的儲存檔位於稱為 SOMELIB 的程式庫中，以 FILE(SOMELIB/MYSAVF) 表示。

您也可以指定遵循「整合式檔案系統 (IFS)」命名慣例的 UNIX 形式路徑名稱，以識別相同的儲存檔。如需相關資訊，請參閱 [QSYS.LIB 檔案系統中的路徑名稱](#)。使用 IFS 命名慣例時，SOMELIB 中的 MYSAVF 具有下列路徑名稱：

```
/QSYS.LIB/SOMELIB.LIB/MYSAVF.FILE
```

IBM i 上的 Managed File Transfer 可辨識 IFS 命名慣例，但不支援 CL 指令使用的語法。下列範例說明 Managed File Transfer 的有效及無效路徑名稱。

儲存檔傳送的一些有效路徑名稱範例，如下所示：

- /QSYS.LIB/SOMELIB.LIB/MYSAVF.FILE (假設 MYSAVF 儲存檔位於程式庫 SOMELIB)
- /QSYS.LIB/MYSAVF.FILE (假設 MYSAVF 位於程式庫 QSYS)

儲存檔傳送的一些無效路徑名稱範例，如下所示：

- SOMELIB.LIB/MYSAVF.FILE (路徑名稱必須以 /QSYS.LIB 為開頭)
- /QSYS.LIB/MYLIB.LIB (路徑必須以儲存檔名稱為結尾，而非程式庫名稱)
- /QSYS.LIB/MYLIB.LIB/ (需要儲存檔名稱)
- /QSYS.LIB/SOMELIB.LIB/MYSAVF (儲存檔的名稱中必須有 .FILE 副檔名)
- /QSYS.LIB/SOMELIB.LIB/MYSAVF.SAVF (儲存檔名稱副檔名必須為 .FILE)

在單一傳送要求中從程式庫傳送多個儲存檔

IBM i 上的 Managed File Transfer 支援以單一傳送要求從程式庫傳送多個儲存檔。您可以指定包括萬用字元的適當路徑名稱，如下列範例所示：

- ABCLIB 包含許多儲存檔。若要在單一要求中傳送所有檔案，請指定下列路徑名稱：

```
/QSYS.LIB/ABCLIB.LIB/*.FILE
```

- XYZLIB 包含數個儲存檔，其名稱中只有一個字元不同，亦即：TEST1.FILE、TEST2.FILE、TEST3.FILE 等。若要在單一要求中傳送所有檔案，請指定下列路徑名稱：

```
/QSYS.LIB/XYZLIB.LIB/TEST?.FILE
```

下列類型的傳送要求不支援傳送多個儲存檔，會導致錯誤：

- ```
/QSYS.LIB/MYLIB.LIB/*.*
```
- ```
/QSYS.LIB/MYLIB.LIB/*
```

Managed File Transfer 支援傳送位於 QSYS.LIB 檔案系統中的儲存檔，但不支援傳送位於 QSYS.LIB 檔案系統中的其他類型檔案。不過，Managed File Transfer 提供的範例使用儲存檔支援及使用預先定義的 fteAnt 作業，示範如何在兩個 IBM i 系統之間傳送完整程式庫、來源實體檔案或資料庫檔案。如需如何自訂及使用這些範例的詳細資料，請參閱第 211 頁的『將 Ant Script 與 MFT 搭配使用入門』。

相關參考

第 530 頁的『傳送檔案的準則』

視傳送的來源及目的地作業系統以及是以二進位還是以文字模式進行傳送而定，對於預期發生的行為有一些準則可循。

第 543 頁的『與 IBM i 系統來回傳送檔案』

如果您在文字模式下使用 Managed File Transfer 與 IBM i 系統來回傳送檔案，並且想要轉換檔案中的資料，則請考量本主題中的資訊。

傳送世代資料群組 (GDG)

Managed File Transfer 支援對於 z/OS 上的來源及目的地資料集使用世代資料群組 (GDG)。支援絕對及相對 GDG 名稱。在您寫入新世代時，基本 GDG 必須存在。

註：在批次環境中使用 BASEGDG (+n) 建立 GDG 項目時，稍後在相同工作中無法使用相同的正世代號碼來參照該 GDG 項目。在工作步驟之間保留相同的 GDG 項目號碼是 JCL 的職責，並非由使用動態配置來更新 GDG 的公用程式函數實現。因此，一旦傳送順利完成，使用 BASEGDG(+1) 建立新世代的工作就會發現 GDG 已更新，並且隨後需要參照與 BASEGDG(0) 相同的資料集。

GDG 範例

下列範例顯示使用 GDG 的 **fteCreateTransfer** 指令。在這些範例中，名稱 BASEGDG 指現有的基本 GDG 名稱。名稱 DSET 指要建立的循序資料集。名稱 /u/user/file.dat 指來源資料檔案的名稱。

以下指令會將 file.dat 複製到 BASEGDG 中的新世代。新世代的絕對名稱會列在傳送日誌報告中：

```
fteCreateTransfer -sa A1 -da A2 -ds "//BASEGDG(+1)" /u/user/file.dat
```

以下指令會將 file.dat 複製到具有 BASEGDG 中所指定絕對名稱的世代。

```
fteCreateTransfer -sa A1 -da A2 -ds "//BASEGDG.G0009V00" /u/user/file.dat
```

以下指令會將 BASEGDG 中的最新世代複製到 DSET。該世代的絕對名稱會列在傳送日誌報告中：

```
fteCreateTransfer -sa A1 -da A2 -ds "//DSET" "//BASEGDG(0)"
```

以下指令會將 BASEGDG 中下一個最新世代複製到 DSET。該世代的絕對名稱會列在傳送日誌報告中：

```
fteCreateTransfer -sa A1 -da A2 -ds "//DSET" "//BASEGDG(-1)"
```

相關參考

第 530 頁的『傳送檔案的準則』

視傳送的來源及目的地作業系統以及是以二進位還是以文字模式進行傳送而定，對於預期發生的行為有一些準則可循。

第 367 頁的『fteCreateTransfer: 啟動新的檔案傳送』

fteCreateTransfer 指令從指令行建立並啟動新的檔案傳送。此指令可立即啟動檔案傳送、將檔案傳送排定在未來的時間和日期、將排定的傳送重複一次以上，以及根據特定條件觸發檔案傳送。

第 532 頁的『在 z/OS 上的資料集之間傳送』

您可以使用 Managed File Transfer 在 z/OS 資料集之間進行傳送。請仔細檢閱下列行為，以確保正確傳送資料集。

搭配使用萬用字元與 MFT

當您指定用於檔案傳送的來源檔案名稱及來源檔路徑時，可以使用萬用字元。這可讓您同步選取多個檔案。

多平台

您可以在 [Multiplatforms](#) 上使用下列萬用字元：

?

請使用問號 (?) 來代表正好一個字元。在比對檔案名稱時，所有其他指定字元必須相符。

例如，ab?d.jpg 符合 abcd.jpg、abed.jpg 及 abfd.jpg 檔。


*

使用星號字元 (*) 代表零個以上字元。

例如，*.txt 符合檔案 abc.txt 和 x.txt，但不符合 newtxt，因為檔名中的句點 (.) 是必要字元。

型樣 *txt 符合檔案 abc.txt、x.txt 及 newtxt。

您必須使用雙引號含括星號字元 (*)。如果沒有這麼做，該字元會由指令 Shell 解譯，而可能導致指令失敗。

 在 UNIX and Linux 上，使用星號字元 (*) 將不會包括虛擬隱藏式檔案，例如 .bashrc。

如果作業系統在檔案及路徑名稱上不區分大小寫，例如，Windows，則型樣相符不區分大小寫。您只能使用萬用字元來指定檔案名稱：您不能在目錄名稱中使用萬用字元。

通訊協定橋接器代理程式

如果要使用通訊協定橋接器代理程式來傳送 FTP、FTPS 或 SFTP 檔案伺服器中的檔案，則萬用字元比對要區分大小寫，這與實際執行檔案伺服器的平台無關。

Connect:Direct 橋接器

當傳送的來源是 Connect:Direct 橋接器代理程式，而此代理程式要求 Connect:Direct 節點中的檔案時，不支援萬用字元。

IBM i

 IBM i

您可以在 IBM i 平台上使用下列萬用字元：

?

請使用問號 (?) 來代表正好一個字元。在比對檔案名稱時，所有其他指定字元必須相符。

例如，ab?d.jpg 符合 abcd.jpg、abed.jpg 及 abfd.jpg 檔。

*

使用星號字元 (*) 代表零個以上字元。

例如，*.txt 符合 abc.txt 及 x.txt 檔。

型樣 *txt 符合 abc.txt、x.txt 和 newtxt 檔案，因為型樣中的句點 (.) 是必要字元。

如需關於在儲存檔案傳送中使用萬用字元的其他考量，請參閱在 [IBM i 系統上傳送位於 QSYS.LIB 檔案系統中的儲存檔](#)。

z/OS

z/OS 若為 z/OS 系統，Managed File Transfer 的萬用字元規則一般會遵循標準 ISPF 萬用字元使用慣例。循序及分割的資料集有一些特定的規則，如下所示：

循序資料集

z/OS 當您參照循序資料集時，可使用包含星號 (*) 及百分比符號 (%) 的資料集名稱限定元，如下所示：

*

使用單一星號 (*) 代表至少一個限定元。限定元內的單一星號代表零個以上字元。

**

使用兩個星號 (**) 代表零個以上限定元。您不能在限定元內使用兩個星號。

%

使用單一百分比符號 (%) 代表單一英數字元或國家語言字元。

%%

使用 1 到 8 個百分比符號代表零個以上字元。

已分割的資料集

z/OS 當您參照分割的資料集時，只能對成員名稱指定萬用字元。您可以使用包含星號 (*)、底線 () 及問號 (?) 的資料集名稱限定元，如下所示：

*

使用星號 (*) 字元代表零個以上字元。

_

使用底線 () 字元代表正好一個字元。

?

請使用問號 (?) 字元來代表正好一個字元。問號是底線字元的替代字元，提供作為 ISPF 使用慣例的額外項目。

目錄

依預設，如果您建立檔案傳送時使用的萬用字元型樣符合子目錄，不會傳送子目錄。您可以在 `fteCreateTransfer` 指令上指定 `-r` 參數，以包含符合萬用字元型樣的子目錄。當您傳送子目錄時，會傳送子目錄的整個內容及結構：包括其所有檔案、子目錄及隱藏式檔案。

比方說，例如您有一個稱為 `abc` 的目錄，則指定來源檔案路徑 `/opt/abc` 與 `/opt/abc/*` 會造成行為上的差異。如果是 `/opt/abc`，因為傳送了目錄，所以會在目的地建立一個稱為 `abc` 的目錄，並且會傳送所有檔案內容。如果是 `/opt/abc/*`，則會將 `abc` 的內容傳送至目的地路徑。

隱藏式檔案

萬用字元不符合隱藏式檔案，除非在 UNIX 類型平台上，萬用字元型樣以點字元 (.) 開頭。例如: /opt/.***** 會傳送 opt 目錄中的所有隱藏式檔案。

在 Windows 上，如果您想要傳送隱藏式檔案，請確切指定檔名，或傳送包含隱藏式檔案的目錄。

符號鏈結

符號鏈結是一種檔案類型，它包含另一個檔案或目錄的指標，這在 Windows 上稱為捷徑。您可以比對符號鏈結檔案與萬用字元。不過，當目的地檔案是從符號鏈結的來源建立時，目的地檔案會變成固定鏈結（亦即，一般檔案）。您無法順利將符號鏈結傳送至目錄，因為這可能會建立遞迴路徑。

傳送其檔案名稱中有萬用字元的檔案

您可以傳送檔案名稱本身包含萬用字元的檔案。如果您確切指定該檔案名稱，則只會傳送該檔案，而不會傳送符合該萬用字元的那一組檔案。

比方說，例如您有一個稱為 /opt/abc*.txt 的檔案，且建立 /opt/abc*.txt 的檔案傳送，則唯一傳送的檔案是 /opt/abc*.txt。但是，如果您建立 /opt/ab*.txt 的檔案傳送，則會傳送符合 /opt/ab*.txt 型樣的所有檔案，包括 /opt/abc*.txt 檔。

傳送包含萬用字元的目錄路徑

以雙引號 (") 或單引號 (') 括住任何其中包括萬用字元的目錄路徑，以避免被 Shell 展開。發生 Shell 展開時，作業系統會在萬用字元傳遞至 Managed File Transfer 指令之前，展開萬用字元，而如此可能會造成非预期的行為。

例如，如果您在 UNIX 上執行下列 **fteCreateTransfer** 指令時搭配 **-gt** 參數，其中 `${...}` 是資源監視器中的變數替代：

```
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 剖析 `${FilePath}` 且不會將其傳遞給指令。暫行解決方法是以雙引號括住 `${FilePath}`，即 `"${FilePath}"`。

傳送回報成功，但是萬用字元不符合任何檔案

如果您試圖傳送不存在的檔案，Managed File Transfer 會將此試圖視為傳送失敗。如果已明確指定檔名（例如 /a/missing/filename.txt），且 MFT 找不到該檔案，則會在日誌中回報下列錯誤訊息：

```
BFGI00001E: File "/a/missing/filename.txt" does not exist
```

在此過程中，找不到檔案的來源代理程式，會通知目的地代理程式此檔案傳送已取消（因為來源代理程式找不到要讀取的原始檔）。如果您此時計劃在傳送後觸發結束程式，目的地代理程式會觸發其 `DestinationTransferEndExit`，且該檔名的 `FileExitResultCode` 為 `CANCEL_FILE`。

但是，如果您試圖傳送萬用字元（例如 /a/missing/*.txt）且來源代理程式找不到符合該萬用字元的任何檔案，MFT 會回報此傳送成功。因為就技術上來說，系統要求來源代理程式傳送 0 個檔案。日誌中回報下列錯誤訊息：

```
The transfer request has successfully completed, although no files were transferred.
```

在此範例中，因為目的地代理程式從未參與傳送，所以不會呼叫其結束程式。

相關參考

第 530 頁的『傳送檔案的準則』

視傳送的來源及目的地作業系統以及是以二進位還是以文字模式進行傳送而定，對於預期發生的行為有一些準則可循。

第 367 頁的『fteCreateTransfer: 啟動新的檔案傳送』

fteCreateTransfer 指令從指令行建立並啟動新的檔案傳送。此指令可立即啟動檔案傳送、將檔案傳送排定在未來的時間和日期、將排定的傳送重複一次以上，以及根據特定條件觸發檔案傳送。

在兩個不同的 MFT 拓撲之間傳送

Managed File Transfer (MFT) 代理程式只能在相同拓撲中的其他代理程式之間執行受管理傳送。不過，如果您有多個拓撲，則在它們之間傳送資料會很有用。下列文字提供如何執行此動作的一些高階指引。

以下圖表顯示兩種不同的拓撲：

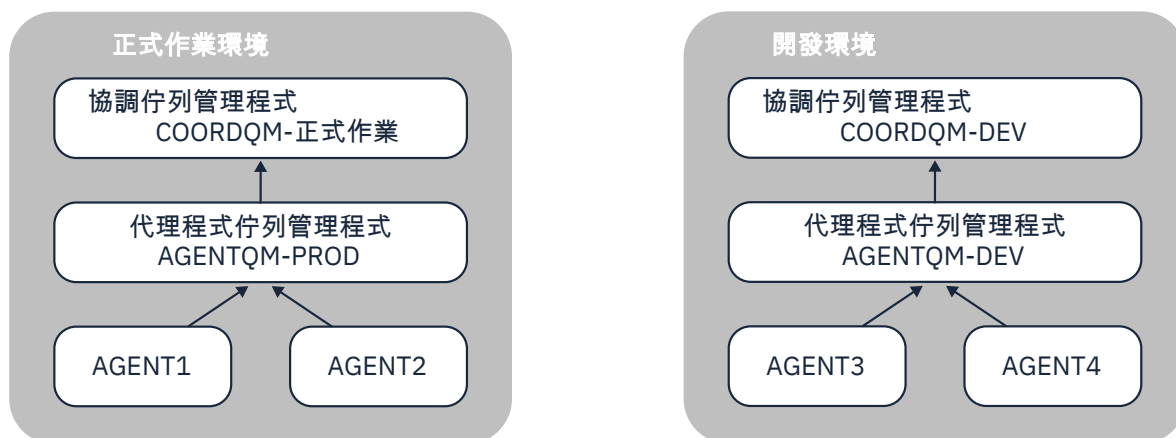


圖 14: AGENT1 和 AGENT2 是正式作業環境中拓撲的一部分，AGENT3 和 AGENT4 是開發環境拓撲的一部分。

「正式作業」拓撲與「開發」拓撲是分開的。這表示「正式作業」中的代理程式無法直接與「開發」環境中的代理程式一起參與受管理傳送 (例如，AGENT2 無法對 AGENT3 執行受管理傳送)。若要在環境之間傳送資料，您可以使用共用檔案系統，或「檔案轉為訊息」及「訊息轉為檔案」傳送。

使用共用檔案系統傳送資料

在此解決方案中，兩個拓撲中的代理程式都可以存取相同的共用檔案系統。

一個拓撲中的代理程式充當受管理傳送的目的地代理程式，並將檔案寫入檔案系統上的已知位置。第二個拓撲中的另一個代理程式使用資源監視器或排定的傳送來偵測檔案何時出現在該位置，然後處理它。

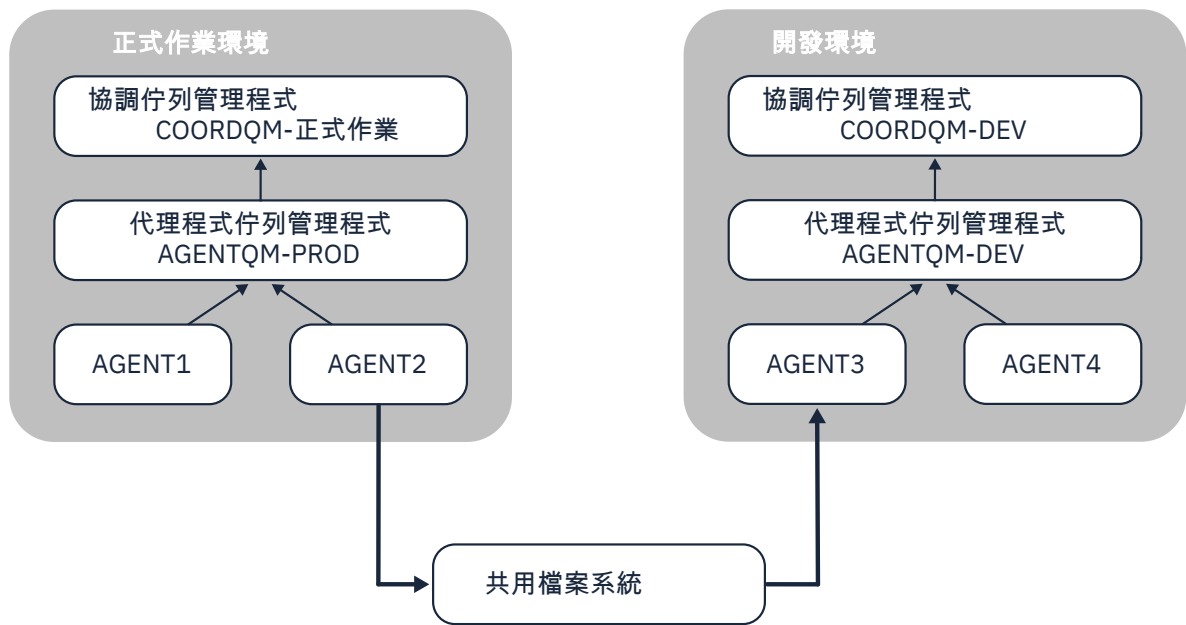


圖 15: 在這裡，AGENT2 是在「正式作業環境」拓撲中執行之受管理傳送的目的地代理程式，並將檔案寫入共用檔案系統上的某個位置。在 AGENT3 內執行的資源監視器會輪詢該位置。當它偵測到 AGENT2 所寫入的檔案時，它會向 AGENT3 提交受管理傳送要求來處理它，並將它帶入「開發環境」拓撲中。

請注意，共用檔案系統應該是可靠的，以確保資料不會遺失。

使用「訊息轉為檔案」及「檔案轉為訊息」傳送來傳送資料

替代方法是在兩個拓撲之間使用閘道佇列管理程式。此佇列管理程式會使用傳送端及接收端通道連接至拓撲中的代理程式佇列管理程式，以容許在兩者之間傳遞資料。

其中一個拓撲中的代理程式會執行檔案至訊息傳送，以將資料寫入遠端佇列。然後，該訊息會透過閘道佇列管理程式遞送至另一個拓撲中佇列管理程式上的本端佇列。然後，該拓撲中的代理程式會執行「訊息轉為檔案」傳送，以取得訊息並處理該訊息。



圖 16: 在這裡，AGENT2 會連接至其代理程式佇列管理程式 AGENTQM-PROD，並執行「檔案至訊息」傳送，以將訊息寫入稱為 Q1 的佇列。Q1 是遠端佇列，因此訊息會透過「閘道佇列管理程式」及傳送端/接收端通道，遞送至佇列管理程式 AGENTQM-DEV 上的本端佇列 Q1。然後，AGENT3 會執行「訊息轉為檔案」傳送，以取得訊息，並將它帶入「開發環境」拓撲。

此解決方案使用標準 IBM MQ 網路，透過閘道佇列管理程式將訊息從一個拓撲傳送至另一個拓撲。這表示如果閘道佇列管理程式與其中一個代理程式佇列管理程式之間的通道由於某些原因而無法使用，則訊息可能會停滯且不會到達目的地佇列。在此狀況下，您應該檢查通道以確保它們都在執行中。

相關參考

第 530 頁的『傳送檔案的準則』

視傳送的來源及目的地作業系統以及是以二進位還是以文字模式進行傳送而定，對於預期發生的行為有一些準則可循。

MFT 所使用的正規表示式

Managed File Transfer 在許多情況下會使用正規表示式。例如，正規表示式可用來比對 Connect:Direct 安全認證的使用者 ID，或將檔案分割成多則訊息（每次符合正規表示式時建立一則新訊息）。Managed File Transfer 所使用的正規表示式語法是 `java.util.regex` API 所支援的語法。此正規表示式類似（但不同於）Perl 語言所使用的正規表示式語法。

如需 Java 正規表示式的相關資訊，請參閱 [Java 指導教學正規表示式](#)。

範例

若要比對所有型樣，請使用下列正規表示式：

```
.*
```

若要比對所有以 `fte` 字串為開頭的型樣，請使用下列正規表示式：

```
fte.*
```

若要比對所有以 `accounts` 字串為開頭、後面跟著一個數字且以 `.txt` 結尾的型樣，請使用下列正規表示式：

```
accounts[0-9]\.txt
```

與使用者定義的 Connect:Direct 程序搭配使用的替代變數

您可以使用 Managed File Transfer 專用的本質符號變數，定義要代入使用者定義的 Connect:Direct 程序的值。

為了遵循 Connect:Direct 命名慣例，Managed File Transfer 使用的所有本質符號變數均會具有下列格式：`%FTE` 後接五個大寫英數字元。如需本質符號變數的相關資訊，請參閱 Connect:Direct 產品文件。


建立程序以將檔案從 Connect:Direct 節點傳送至 Connect:Direct 橋接器系統時，您必須在 Connect:Direct 程序中，將本質變數 `%FTETFILE` 用作 TO FILE 的值。建立程序以將檔案從 Connect:Direct 橋接器系統傳送至 Connect:Direct 節點時，您必須在 Connect:Direct 程序中，將本質變數 `%FTEFFILE` 用作 FROM FILE 的值。這些變數包含 Connect:Direct 橋接器代理程式在 Managed File Transfer 網路中進行傳入及傳出時，所用的暫存檔路徑。

變數名稱	說明
<code>%FTESAGNT</code>	Managed File Transfer 來源代理程式的名稱。僅針對從 Managed File Transfer Agent 至 Connect:Direct 節點的傳送，設定此變數。
<code>%FTEDAGNT</code>	Managed File Transfer 目的地代理程式的名稱。僅針對從 Connect:Direct 節點至 Managed File Transfer Agent 的傳送，設定此變數。
<code>%FTEPNODE</code>	Connect:Direct 主要節點名稱。此值一律是屬於 Connect:Direct 橋接器的 Connect:Direct 節點名稱。
<code>%FTEPPLAT</code>	執行 Connect:Direct 主要節點的平台。此變數的可能值為 UNIX 及 WINDOWS。此資訊由 Connect:Direct 橋接器代理程式提供。
<code>%FTEPUSER</code>	要在 Connect:Direct 程序中使用的 Connect:Direct 主要節點使用者 ID。此資訊將從 <code>ConnectDirectCredentials.xml</code> 檔案取得。

表 69: Managed File Transfer 及 Connect:Direct 使用的本質符號變數 (繼續)

變數名稱	說明
%FTEPPASS	要與 %FTEPUSER 變數所定義的使用者名稱搭配使用的密碼。此資訊將從 ConnectDirectCredentials.xml 檔案取得。
%FTESNODE	Connect:Direct 次要節點名稱。此值一律是將檔案傳送至其中或從中傳送檔案的 Connect:Direct 節點名稱。
%FTESPLAT	執行 Connect:Direct 次要節點的平台。此變數的可能值為 UNIX、WINDOWS 及 ZOS。此資訊將從 ConnectDirectNodeProperties.xml 檔案取得。
%FTESUSER	要在 Connect:Direct 程序中使用的 Connect:Direct 次要節點使用者 ID。此資訊將從 ConnectDirectCredentials.xml 檔案取得。
%FTESPASS	要與 %FTESUSER 變數所定義的使用者名稱搭配使用的密碼。此資訊將從 ConnectDirectCredentials.xml 檔案取得。
%FTEFFILE	來源檔案名稱。請僅針對在每個檔案範圍內提交的 Connect:Direct 程序，設定此變數。 將檔案從 Managed File Transfer Agent 傳送至 Connect:Direct 節點時，此值為與 Connect:Direct 橋接器位於相同系統上的檔案的完整位置。 將檔案從 Connect:Direct 節點傳送至 Managed File Transfer Agent 時，此值為在 Managed File Transfer 傳送要求中指定為來源檔案的檔案名稱。
%FTEFDISP	程序完成時，來源檔案的處置方式。此變數的值與平台相關，且與 MFT 傳送要求的值相等。請僅針對在每個檔案範圍內提交的 Connect:Direct 程序，設定此變數。 將檔案從 Managed File Transfer Agent 傳送至 Connect:Direct 節點時，將由 Managed File Transfer 橋接器代理程式執行刪除或不刪除來源檔案的動作。 將檔案從 Connect:Direct 節點傳送至 Managed File Transfer Agent 時，必須由 Connect:Direct 程序執行刪除或不刪除來源檔案的動作。
%FTEFCP	用於來源檔案的字碼頁。請僅針對在每個檔案範圍內提交的 Connect:Direct 程序，設定此變數。 將檔案從 Managed File Transfer Agent 傳送至 Connect:Direct 節點時，此值為 UTF-8；如果是二進位傳送，則不設定此值。 將檔案從 Connect:Direct 節點傳送至 Managed File Transfer Agent 時，此值由 Connect:Direct 指定；如果是二進位傳送，則不設定此值。
%FTEFSYSO	傳送來源的 Connect:Direct SYSOPTS。如果遠端 Connect:Direct 節點是在 Linux、UNIX 或 Windows 上，此值將包含傳送來源的字碼頁和資料類型相關資訊。  如果遠端節點是在 z/OS 上，則此值還包含其他資訊。
%FTEFNODE	識別來源檔案所處的 Connect:Direct 節點。此變數將設定為值 PNODE 或 SNODE。
%FTETFILE	目的地檔案名稱。請僅針對在每個檔案範圍內提交的 Connect:Direct 程序，設定此變數。 將檔案從 Managed File Transfer Agent 傳送至 Connect:Direct 節點時，此值為在 Managed File Transfer 傳送要求中指定為目的地檔案的檔案名稱。 將檔案從 Connect:Direct 節點傳送至 Managed File Transfer Agent 時，此值為將檔案寫入與 Connect:Direct 橋接器位於相同系統上的位置的完整名稱。

表 69: Managed File Transfer 及 Connect:Direct 使用的本質符號變數 (繼續)

變數名稱	說明
%FTETDISP	目的地檔案的處置方式。此變數的值與平台相關，且與 Connect:Direct 傳送要求的值相等。請僅針對在每個檔案範圍內提交的 Connect:Direct 程序，設定此變數。 將檔案從 Managed File Transfer Agent 傳送至 Connect:Direct 節點時，必須由 Connect:Direct 程序執行建立檔案或取代現有檔案的動作。 將檔案從 Connect:Direct 節點傳送至 Managed File Transfer Agent 時，將由 Managed File Transfer 橋接器代理程式執行建立檔案或取代現有檔案的動作。
%FTETCP	用於目的地檔案的字碼頁。請僅針對在每個檔案範圍內提交的 Connect:Direct 程序，設定此變數。 將檔案從 Managed File Transfer Agent 傳送至 Connect:Direct 節點時，此值由 Connect:Direct 指定；如果是二進位傳送，則不設定此值。 將檔案從 Connect:Direct 節點傳送至 Managed File Transfer Agent 時，此值為 UTF-8；如果是二進位傳送，則不會設定此值。
%FTETSYSO	傳送目的地的 Connect:Direct SYSOPTS。如果遠端 Connect:Direct 節點是在 UNIX、Connect:Direct 或 Windows 上，此值將包含傳送目的地的字碼頁和資料類型相關資訊。  如果遠端節點是在 Windows 上，則此值還包含其他資訊。
%FTETNODE	識別目的地檔案所處的 Connect:Direct 節點。此變數將設定為值 PNODE 或 SNODE。
%FTEDTYPE	傳送的資料類型或模式。此變數可能具有的值為 text 或 binary。請僅針對在每個檔案範圍內提交的 Connect:Direct 程序，設定此變數。
%FTETRID	來自 Managed File Transfer 傳送的 48 字元十六進位傳送 ID。
%FTEJOB	來自 Managed File Transfer 傳送要求的工作名稱。此變數的值將會截斷為 256 個字元，可用於統計資料的程序。
%FTEPNAME	Managed File Transfer 橋接器代理程式產生的 Connect:Direct 程序名稱。此變數的值為 8 個英數字元。此值一律以英文字母開頭。
%FTEMETA(key)	來自 Managed File Transfer 傳送要求的 meta 資料。key 的值為 meta 資料的索引鍵。key 的值不區分大小寫。索引鍵 ABC 將會視為與索引鍵 abc 相同。如果將 ABC 及 abc 都定義為 meta 資料索引鍵，則所定義的第二個 meta 資料的值，將會改寫所定義的第一個 meta 資料的值。

 下表包含傳送中的遠端 Connect:Direct 節點在 z/OS 平台上時使用的其他本質符號變數等相關資訊。

表 70:

變數名稱	說明
%FTEFDCB	傳送來源中的 DCB 參數值。
%FTEFSPACE	傳送來源中的 SPACE 參數值。
%FTEFLBEL	傳送來源中的 LABEL 參數值。
%FTEFUNIT	傳送來源中的 UNIT 參數值。
%FTEFVOL	傳送來源中的 VOL 參數值。
%FTEFDAACL	傳送來源中的 DATACLAS 參數值。

表 70: (繼續)	
變數名稱	說明
%FTETDCB	傳送目的地中的 DCB 參數值。
%FTETSPACE	傳送目的地中的 SPACE 參數值。
%FTETLABEL	傳送目的地中的 LABEL 參數值。
%FTETUNIT	傳送目的地中的 UNIT 參數值。
%FTETVOL	傳送目的地中的 VOL 參數值。
%FTETDACL	傳送目的地中的 DATACLAS 參數值。
%FTETDSTY	傳送目的地中的 DSNTYPE 參數值。
%FTETLIKE	傳送目的地中的 LIKE 參數值。
%FTETMGCL	傳送目的地中的 MGMTCLAS 參數值。
%FTETSTCL	傳送目的地中的 STORCLAS 參數值。

範例：呼叫 MFT 指令的 Connect:Direct 程序檔案

呼叫 Managed File Transfer **ftetag** 指令及 **ftecxfer** 指令的 Connect:Direct 程序檔案範例。

在此範例中，會發生下列動作：

1. Connect:Direct COPY 陳述式會將檔案從執行次要節點之系統上的 C:\test\from\sent.txt 傳送至執行主要節點之系統上的 C:\test\tmp\midpoint.txt。
2. Connect:Direct 程序呼叫 **ftetag** 指令，以在 MFT 中建立審核資訊。
3. Connect:Direct 程序呼叫 **ftecxfer** 指令。
4. **ftecxfer** 指令將執行主要節點及代理程式 CD_BRIDGE 的系統上的檔案 C:\test\tmp\midpoint.txt，傳送至 LINUX_AGENT 代理程式所在系統上的 /test/to/arrived.txt。

```

/*BEGIN_REQUESTER_COMMENTS
  $PNODE$="cd_win01" $PNODE_OS$="Windows"
  $SNODE$="CD_WIN01" $SNODE_OS$="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:\wmgfte\bin\ftetag) args(C:\test\tmp\midpoint.txt)"

RUN TASK PNODE
  SYSOPTS="pgm(C:\wmgfte\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

```

相關概念

第 204 頁的『[使用 Connect:Direct 程序提交 Managed File Transfer 傳送要求](#)』

您可以透過 Connect:Direct 程序，將傳送要求提交至 Connect:Direct 橋接器代理程式。Managed File Transfer 提供可在 Connect:Direct 處理程序中從 **RUN TASK** 陳述式呼叫的指令。

相關工作

第 204 頁的『[使用「Connect:Direct 要求端」建立及提交呼叫 Managed File Transfer 的 Connect:Direct 程序](#)』

「Connect:Direct 要求端」是一種圖形使用者介面，您可以用來建立及提交呼叫 Managed File Transfer 的 Connect:Direct 程序。

Connect:Direct 橋接器代理程式的限制

Connect:Direct 橋接器代理程式配置為從 Connect:Direct 節點來回傳送檔案。但 Connect:Direct 橋接器代理程式無法執行某些功能。

- Connect:Direct 橋接器代理程式無法從佇列讀取訊息，或無法將訊息寫入佇列。它無法充當「檔案轉為訊息」傳送中的目的地代理程式，或無法充當「訊息轉為檔案」傳送中的來源代理程式。
- 您無法在 Connect:Direct 橋接器代理程式中定義資源監視器。
- 您無法將 Connect:Direct 橋接器代理程式同時作為傳送的來源及目的地。您無法透過 Connect:Direct 橋接器，從 Connect:Direct 節點傳送至 Connect:Direct 節點。
- Connect:Direct 橋接器代理程式不支援在傳送之前或之後呼叫的使用者結束程式。Connect:Direct 橋接器代理程式不支援認證對映結束程式。如需相關資訊，請參閱第 123 頁的『[使用結束程式類別對映 Connect:Direct 的認證](#)』。
- 對於將 Connect:Direct 橋接器代理程式作為來源代理程式的傳送，您無法定義 presrc 或 postsrc 程式呼叫。如需相關資訊，請參閱第 749 頁的『[程式呼叫巢狀元素](#)』。
- 對於將 Connect:Direct 橋接器代理程式作為目的地代理程式的傳送，您無法定義 predst 或 postdst 程式呼叫。如需相關資訊，請參閱第 749 頁的『[程式呼叫巢狀元素](#)』。
- 如果來源代理程式為 Connect:Direct 橋接器代理程式，您無法在來源規格中指定萬用字元。
- 在從 Connect:Direct 節點傳送檔案 **z/OS** 或資料集時，如果您將來源處置 (**-sd**) 指定為 delete，則其行為不同於常見的來源處置行為。發生下列其中一種情況：
 - 如果 Connect:Direct 使用 Managed File Transfer 產生的程序，從來源中移動檔案或資料集，則指定 delete 選項會造成傳送失敗。若要指定刪除來源檔案，您必須提交使用者定義的 Connect:Direct 程序。如需相關資訊，請參閱第 200 頁的『[透過檔案傳送要求提交使用者定義的 Connect:Direct 程序](#)』。
 - 如果 Connect:Direct 利用使用者定義的程序從來源中移出檔案或資料集，則會透過 **%FTEFDISP** 內建符號變數將此參數傳遞給程序。使用者定義的程序決定是否刪除來源。傳送所傳回的結果，取決於使用者定義的程序所傳回的結果。

相關概念

第 192 頁的『[Connect:Direct 橋接器](#)』

您可以從現有 IBM Sterling Connect:Direct 網路來回傳送檔案。使用 Connect:Direct 橋接器 (Managed File Transfer 元件)，可在 MFT 與 IBM Sterling Connect:Direct 之間傳送檔案。

通訊協定橋接器支援的 FTPS 伺服器

通訊協定橋接器可支援 RFC-2228、RFC-4217 及名稱為 *Secure FTP over SSL* 的網際網路草案所定義的 FTPS 通訊協定子集。



如需通訊協定橋接器代理程式與 FTPS 伺服器之間連線的有效密碼組合值清單，請參閱 IBM SDK 及執行時期環境 Java Technology Edition 7 產品說明文件中的 [密碼組合](#)。

支援 FTPS 通訊協定的下列特性：

- 作業的隱含及明確模式。
- 驗證伺服器憑證。

- 使用用戶端憑證檢查進行交互鑑別（選用）。
- 在為資料通道選取起始鑑別及保護層次後使用明確的控制通道（選用）。
- 支援 SHA-2 密碼組合及 FIPS 140-2 相符性。必須具備下列版本的 Java: IBM JRE 6.0 SR13 FP2、7.0 SR4 FP2 或更新版本。

不支援 FTPS 通訊協定的下列特性及執行時期環境:

- 針對其他安全資料交換使用 **ADAT** 指令。
- 僅針對伺服器憑證未經驗證的通道加密使用 FTPS。
- 使用 **PROT** 指令選取 **清除**、**安全** 或 **機密** 保護層次。
- 使用 **MIC**、**CONF** 和 **ENC** 指令來加密每一個指令。
- 如果伺服器不支援明確 FTPS，則使用 FTP 通訊協定。使用通訊協定橋接器提供的 FTP 支援，以使用這類伺服器。
- 使用 **FEAT** 指令來決定 FTPS 伺服器的可用功能。
- 針對 DN 欄位使用型樣相符來驗證憑證。
- 憑證撤銷檢查。
- 藉由簽發授信憑證管理中心驗證憑證。
- 明確選取適用於建立階段作業之 SSL 協議階段的密碼組合。
-  使用專屬於 z/OS  或 IBM i 的延伸，以將加密法與作業系統整合。具體而言，使用 z/OS 金鑰環或非階層式檔案系統來儲存金鑰及信任資訊（例如，資料集）。如果這些功能是由 JVM 以透過方式加以管理且不需要明確的應用程式碼，則會使用加密硬體及卸載引擎。

相關概念

第 178 頁的『通訊協定橋接器』

通訊協定橋接器可供 Managed File Transfer (MFT) 網路存取儲存在 MFT 網路之外的檔案伺服器上的檔案（在本端網域或遠端位置）。此檔案伺服器可使用 FTP、FTPS 或 SFTP 網路通訊協定。每一個檔案伺服器需要至少一個專用的代理程式。專用代理程式即所謂的通訊協定橋接器代理程式。橋接器代理程式可以與多個檔案伺服器互動。

通訊協定橋接器支援 SFTP 伺服器

通訊協定橋接器支援 IETF「網際網路草稿」（標題為「SSH 檔案傳送通訊協定」第 6 版草稿 13）所定義的 SFTP 通訊協定。

使用 SFTP 通訊協定連接至檔案伺服器時，通訊協定橋接器代理程式支援下列密碼:

- blowfish-cbc
- 3des-cbc
- aes128-cbc
- aes192-cbc
- aes256-cbc
- aes128-ctr
- aes192-ctr
- aes256-ctr
- 3des-ctr
- arcfour
- arcfour128
- arcfour256

依預設，通訊協定橋接器代理程式使用的密碼清單為 aes128-cbc,aes192-cbc,aes256-cbc。如需如何配置通訊協定橋接器代理程式使用不同密碼的相關資訊，請參閱 [第 484 頁的『通訊協定橋接器內容檔格式』](#)。

鑑別方法

如果您已提供具有私密金鑰及伺服器密碼的 IBM MQ Managed File Transfer (MFT) 通訊協定橋接器代理程式程式碼，則對於 ProtocolBridgeCredentials.xml 檔案內的單一使用者，依預設，MFT 通訊協定橋接器代理程式會將 JSch 程式庫配置為在建立連線時使用兩種鑑別方法 (如果 SFTP 檔案伺服器需要的話)。

如果在 ProtocolBridgeCredentials.xml 檔內同時為單一使用者配置私密金鑰和伺服器密碼，但 SFTP 檔案伺服器只需要其中一種鑑別方法，則 MFT 通訊協定橋接器代理程式會將 JSch 程式庫配置成使用公開/私密金鑰鑑別，而不是密碼型鑑別。

如果 SFTP 檔案伺服器拒絕嘗試使用公開/私密金鑰鑑別，則使用 JSch 程式庫的 MFT 通訊協定橋接器代理程式會嘗試使用者名稱及密碼型鑑別。

如果其中任一鑑別單獨成功，則會建立與 SFTP 檔案伺服器的連線。

若要為與 MFT 通訊協定橋接器代理程式相關聯的 ProtocolBridgeCredentials.xml 檔案配置私密金鑰及密碼鑑別，您需要指定：

- 元素中從 MFT 使用者名稱對映至通訊協定伺服器使用者名稱的 **serverPassword** 屬性 (具有相關聯的值)，以及
- 母元素所定義之 MFT 使用者的元素。

例如，語法可能如下：

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

鍵盤互動方法

MFT 通訊協定橋接器代理程式使用 JSch (協力廠商程式庫) 來連接至 SFTP 檔案伺服器。您可以配置 JSch 程式庫，以便在 ProtocolBridgeCredentials.xml 檔中未指定私密金鑰時，它可以嘗試使用 鍵盤-互動式方法向 SFTP 檔案伺服器進行鑑別。

請注意，只有在 SFTP 檔案伺服器使用字串 password: (大寫、小寫或混合大小寫) 提示輸入密碼時，使用 *keyboard-interactive* 方法進行鑑別才有效。如果您使用 *keyboard-interactive* 鑑別方法，且 SFTP 檔案伺服器以不同於 password: 的字串回應，則連線嘗試會失敗。

當 SFTP 檔案伺服器使用此字串回應起始連線嘗試時，通訊協定橋接器代理程式會使用 JSch 程式庫，傳送到 ProtocolBridgeCredentials.xml 檔案內 user 元素的 **serverPassword** 屬性中配置的密碼。

相關資訊

[通訊協定橋接器](#)

MFT 中的 FIPS 支援

Managed File Transfer 支援在從代理程式、指令及 IBM MQ Explorer 到佇列管理程式的用戶端連線中，使用符合 FIPS 標準的加密模組。所有與佇列管理程式的 SSL 連線都僅使用 TLS 通訊協定。該支援針對 JKS 及 PKCS#12 金鑰儲存庫類型提供。

指定您是否要為代理程式、協調佇列管理程式或指令佇列管理程式啟用 FIPS 支援，如下所示：

- 如果您想要為特定代理程式啟用 FIPS，請在 agent.properties 檔案中針對該代理程式設定適當的 agentSsl 內容。如需相關資訊，請參閱 [第 508 頁的『MFT 的 SSL 內容』](#)。
- 如果您想要為特定協調佇列管理程式啟用 FIPS，請在 coordination.properties 檔案中針對該協調佇列管理程式設定適當的 coordinationSsl 內容。如需相關資訊，請參閱 [第 508 頁的『MFT 的 SSL 內容』](#)。
- 如果您想要為特定指令佇列管理程式啟用 FIPS，請在 command.properties 檔案中針對該指令佇列管理程式設定適當的 connectionSsl 內容。如需相關資訊，請參閱 [第 508 頁的『MFT 的 SSL 內容』](#)。

 若為  IBM i，FIPS 在 Managed File Transfer 上不受支援。

通訊協定橋接器或 Connect:Direct 橋接器的雙向連線不支援 FIPS。

如需 IBM MQ 與 FIPS 及所需配置步驟的相關資訊，請參閱 [Federal Information Processing Standards \(FIPS\)](#)。

如果您想要使用 FIPS，CipherSuite 必須符合 FIPS 標準，否則連線會失敗。如需 IBM MQ 支援的 CipherSpec 的相關資訊，請參閱 [IBM MQ classes for Java](#) 中的 [SSL/TLS CipherSpec 及 CipherSuite](#) 與 [IBM MQ classes for JMS](#) 中的 [SSL/TLS CipherSpec 及 CipherSuite](#)。

MFT 資料庫日誌程式表格

安裝並配置日誌程式之後，會建立下列資料庫表格：

AUTH_EVENT

與權限檢查相關的事件，通常會因為權限不足而拒絕要求。

- **ID**：列 ID。
- **ACTION**：所發生動作的類型。
- **COMMAND_ID**：要求事件的原始訊息的 IBM MQ 訊息 ID。在傳送要求的情況下，這也指傳送 ID。
- **TIME**：事件的發生時間。
- **ORIGINATOR_MQ_USER**：對其執行權限檢查的 IBM MQ 訊息中所包含的使用者 ID。
- **AUTHORITY**：要求的動作所需的權限。
- **ORIGINAL_XML_REQUEST**：指令訊息的內容，指示所拒絕的動作。
- **RESULTCODE**：用於識別結果的數值代碼。
- **RESULT_TEXT**：說明權限事件結果的訊息。

CALL

遠端執行作業系統指令，或者 Ant Script ，或者 z/OS JCL 工作，由 Managed File Transfer 管理。呼叫可以內嵌在傳送中，或由 call_request 列參照。

「呼叫」（即此表格中的列）可以是一般傳送的一部分（這種情況下，TRANSFER_CALLS 用於將其鏈結至 TRANSFERS 中的相關項目），或者其本身為獨立式受管理呼叫（僅適用於透過 Ant 或直接插入訊息）。對於後者，將使用 CALL_REQUEST 表格而非 TRANSFERS 表格；因為每個呼叫要求只能有一個呼叫，因此不需要 TRANSFER_CALLS 的對等項目。

- **ID**：列 ID。
- **COMMAND**：執行的指令。此欄位不包含傳遞至指令或指令所在路徑的任何引數。
- **TYPE**：指令類型，例如 Ant 或 JCL。
- **RETRIES**：所要求的重試次數。
- **RETRY_WAIT**：按最初要求要在重試間等待的間隔（以秒為單位）。
- **SUCCESS_RC**：指示成功完成指令的回覆碼。如果收到另一個回覆碼，則表示執行報告為失敗。
- **EXECUTED_COMMAND**：所執行指令的完整名稱（包括路徑）。
- **CAPPED_RETRIES**：可用的重試次數；如果代理程式的重試限制低於所要求的重試次數，則此數字可能會小於所要求的數字。
- **CAPPED_RETRY_WAIT**：重試間所使用的間隔；如果配置的代理程式限制低於所要求的重試等待時間，則此數字可能小於所要求的數字。
- **OUTCOME**：呼叫是否整體成功。如果曾進行多次嘗試，則 CALL_RESULT 表格中會分別記錄每次嘗試的結果。

CALL_ARGUMENT

提供給所呼叫指令的引數或參數。

- **ID**：列 ID。

- **CALL_ID**: 與引數相關聯的呼叫。
- **KEY**: 引數屬於鍵值組類型、索引鍵或名稱。
- **TYPE**: 引數類型: 部分為作業系統指令的位置參數, 其他為用於 Ant 的具名內容。
- **VALUE**: 引數的值。

CALL_REQUEST

不屬於檔案傳送的指令呼叫的媒介。您可以使用 Ant 及直接 XML 注入來提交 ManagedCall 訊息。

- **ID**: 受管理呼叫要求的十六進位 ID。
- **CALL_ID**: CALL 表格中說明此呼叫所在列的資料庫 ID。
- **ACTION_TIME**: 動作的發生時間。
- **AGENT**: 執行指令的代理程式。
- **AGENT_QM**: 執行指令的代理程式所使用的佇列管理程式。
- **ARCHITECTURE**: 執行代理程式的系統的機器架構。
- **OS_NAME**: 執行代理程式的作業系統名稱。
- **OS_VERSION**: 作業系統的版本。
- **ORIGINATOR_HOST**: 提交呼叫要求的機器的主機名稱。
- **ORIGINATOR_USER**: 提交呼叫要求的使用者名稱 (與要求 XML 中所報告的一樣)。
- **ORIGINATOR_MQ_USER**: 提交呼叫要求的使用者的名稱 (與要求的 IBM MQ 訊息描述子中所包含的一樣)。
- **JOB_NAME**: 使用者指定的工作名稱。
- **RESULTCODE**: 呼叫的整體結果碼。
- **RESULTTEXT**: 呼叫的整體結果訊息。

CALL_RESULT

呼叫指令的詳細結果。如果已啟用重試, 一個呼叫可能具有多個結果。

- **ID**: 列 ID。
- **CALL_ID**: CALL 表格中此結果所適用列的資料庫 ID。
- **SEQUENCE**: 在進行多次嘗試的情況下, 此結果所適用的嘗試。
- **OUTCOME**: 指令的執行結果 (例如, 成功或失敗)。
- **RETURN_CODE**: 指令回覆碼。
- **TIME**: 指令的完成時間。
- **STDOUT**: 來自指令的標準輸出串流 (如果已啟動)。
- **STDERR**: 來自指令的標準錯誤串流 (如果已啟動)。
- **ERROR**: 如果指令無法啟動, 由 Managed File Transfer 產生用於說明問題的錯誤訊息。

FILE_SPACE_ENTRY

每一列均代表一個已傳送至具名檔案空間的檔案。

- **ID**: 檔案空間項目的 ID。
- **FILE_SPACE_NAME**: 檔案空間的名稱。這是檔案空間所屬的使用者的名稱。
- **TRANSFER_ITEM_ID**: 與此列相關的傳送項目的 ID。
- **ALIAS**: 此檔案空間項目的別名。一般而言, 此別名為傳送的來源檔案的名稱。
- **DELETED**: 從檔案空間刪除檔案的時間。如果檔案尚未刪除, 則值為 null。

METADATA

與傳送相關聯的 meta 資料。

- **ID:** 列 ID。
- **TRANSFER_EVENT_ID:** 如果此 meta 資料與傳送相關，則指與其相關聯的 transfer_event 列。如果 meta 資料與獨立式受管理呼叫相關聯，則此欄位為空值。
- **STANDALONE_CALL_ID:** 如果 meta 資料與獨立式受管理呼叫相關聯，則指所涉及受管理呼叫要求的 ID。
- **KEY:** meta 資料項目的名稱。
- **VALUE:** meta 資料項目的值。

MONITOR

根據外部條件觸發 Managed File Transfer 作業的資源監視器。

- **AGENT:** 執行監視器的代理程式。
- **ID:** 監視器的十六進位 ID。
- **NAME:** 監視器的名稱。
- **QMGR:** 執行監視器的代理程式的佇列管理程式。

MONITOR_ACTION

每一列都代表發生的與監視器有關的一個動作（例如，建立及觸發）。

- **ID:** 列 ID。
- **ACTION:** 所發生動作的類型。
- **JOB_NAME:** 提交工作的時間（如果適用）。
- **MONITOR:** 發生此動作的監視器。如果動作因其所要求的監視器不存在而失敗，則可能為空值。
- **ORIGINAL_XML_REQUEST:** 如果此動作為 *create* 或 *triggerSatisfied* 動作，則指觸發監視器後所啟動的 XML 要求。
- **ORIGINATOR_MQ_USER:** 起始動作的 IBM MQ 訊息中所包含的使用者 ID。
- **ORIGINATOR_USER:** 提交要求以執行動作的使用者名稱。
- **ORIGINATOR_HOST:** 使用者從中提交要求以執行動作的機器。
- **TIME:** 動作的發生時間。
- **UPDATED_XML_REQUEST:** 如果動作為 *triggerSatisfied*，則指已啟動的 XML 要求。此要求會依因變數替代最初產生的 XML 要求而有所不同。

MONITOR_EXIT_RESULT

執行資源監視器結束程式的結果。

- **ID:** 列 ID。
- **ACTION_ID:** 與結果相關聯的監視器動作。
- **EXIT_NAME:** 產生此結果的結束程式的名稱。
- **RESULTCODE:** 結束程式傳回的值，不是取消就是繼續。
- **RESULTTEXT:** 來自結束程式的文字輸出（若提供）。

MONITOR_METADATA

與資源監視器相關聯的 meta 資料項目。

- **ID:** 列 ID。
- **ACTION_ID:** 與 meta 資料相關聯的 monitor_action。

- **KEY:** meta 資料項目的名稱。
- **PHASE:** 此 meta 資料項目是代表最初提交的資料，還是進行變數替代後的更新版本。
- **VALUE:** meta 資料項目的值。

排程

已向代理程式登錄的傳送排程。

- **AGENT:** 排定此排程的代理程式的名稱。
- **CREATION_DATE:** 建立此排程的時間點。
- **ID:** 排程的唯一資料庫（而非代理程式）ID。
- **ID_ON_AGENT:** 代理程式針對資料庫 ID 所使用的 ID。此 ID 在代理程式間不是唯一的，如果重設代理程式的持續性狀態，它甚至在一個代理程式中都不是唯一的。
- **LATEST_ACTION:** 修改此排程狀態的最新動作。

SCHEDULE_ACTION

當發生修改排程狀態的事件時，就會記錄動作。

- **ACTION_TYPE:** 發生的動作。
- **ID:** 列 ID
- **ORIGINATOR_HOST:** 提交導致變更要求的機器。
- **ORIGINATOR_USER:** 提交導致變更的使用者的名稱。
- **SCHEDULE_ID:** 此動作所適用的排程。
- **SPEC_AFTERWARDS:** 執行動作後代表此排程狀態的 schedule_spec。
- **STATUS_CODE:** 說明動作結果的數值回覆碼。
- **STATUS_TEXT:** 動作結果的文字說明。如果動作成功執行，通常為空值。
- **TIME:** 執行動作的時間點。

SCHEDULE_SPEC

個別排定傳送的詳細資料。

- **ID:** 列 ID。
- **DESTINATION_AGENT:** 向其傳送檔案的代理程式。
- **DESTINATION_QM:** 目的地代理程式使用的佇列管理程式。
- **REPEAT_COUNT:** 當排程重複且受出現次數（而非結束時間）約束時，要重複的次數。
- **REPEAT_FREQUENCY:** 排定傳送間的 repeat_interval 個數。
- **REPEAT_INTERVAL:** 如果傳送重複，要重複的間隔時間單位（例如，分鐘或週）。
- **SOURCE_AGENT:** 從中傳送檔案的代理程式。
- **SOURCE_QM:** 來源代理程式使用的佇列管理程式。
- **START_TIME:** 排程中第一個傳送的發生時間。
- **START_TIMEBASE:** 與傳送相關聯的時間的時間基準。例如，要以代理程式的時區運作還是以管理者的時區運作。
- **START_TIMEZONE:** 時間基準所對應且將用於運作排程的時區。

SCHEDULE_ITEM

每個檔案（或在傳送時要符合的型樣）都用 schedule_item 表示。

- **ID:** 列 ID。

- **CHECKSUM_METHOD**: 檔案總和檢查的計算方式
- **DESTINATION_EXISTS_ACTION**: 在目的地上已存在相同檔案時，目的地代理程式要採取的動作。
- **DESTINATION_FILENAME**: 要將檔案傳送至其中的檔案或目錄。
- **DESTINATION_QUEUE**: 檔案到訊息傳送的目的地佇列名稱。
- **Multi DESTINATION_TYPE**: destination_filename 直欄是指檔案或目錄。
- **z/OS DESTINATION_TYPE**: destination_filename 直欄是指檔案、目錄還是資料集。
- **FILE_MODE**: 傳送檔案所採用的模式 (例如, *text* 或 *binary*) 。
- **RECURSIVE**: 當代理程式根據排程建立傳送時，代理程式是 (Y) 否 (N) 要遞迴來源目錄。
- **SCHEDULE_SPEC_ID**: 與此項目相關聯的 schedule_spec。
- **SOURCE_DISPOSITION**: 傳送完成後要對來源檔案執行的動作。
- **SOURCE_FILENAME**: 來源檔案、目錄名稱或型樣。
- **SOURCE_QUEUE**: 訊息到檔案傳送的來源佇列名稱

TRANSFER

一個以上檔案的單一傳送。

- **TRANSFER_ID**: 傳送的十六進位 ID。
- **JOB_NAME**: 傳送的使用者指定的工作名稱。
- **SCHEDULE_ID**: 如果此傳送是排程的結果，則指所涉及排程的資料庫列 ID。
- **START_ID**: 代表傳送開始的 transfer_event 的列 ID。
- **COMPLETE_ID**: 代表傳送結束的 transfer_event 的列 ID。
- **RESULTCODE**: 傳送的整體結果碼。以下主題中列出此直欄的可能值: 第 284 頁的『MFT 的回覆碼』。這些回覆碼適用於整個傳送; 如需每個個別項目的狀態資訊, 請參閱 [TRANSFER_ITEM.RESULTCODE](#)。
- **RESULTTEXT**: 傳送的整體結果文字 (如果有) 。
- **STATUS**: 傳送的狀態。此直欄的可能值包括: started、success、partial success、failure 及 cancelled。
- **RELATED_TRANSFER_ID**: 與此傳送相關的上一個傳送的十六進位 ID。例如, 如果傳送是檔案下載, 則此欄位指的是上傳檔案的傳送。

TRANSFER_CALLS

將可執行的指令呼叫鏈結至傳送

- **ID**: 列 ID。
- **POST_DESTINATION_CALL**: 傳送完成後在目的地進行的呼叫。
- **POST_SOURCE_CALL**: 傳送完成後在來源代理程式進行的呼叫。
- **PRE_DESTINATION_CALL**: 傳送開始前在目的地代理程式進行的呼叫。
- **PRE_SOURCE_CALL**: 傳送開始前在來源代理程式進行的呼叫。
- **TRANSFER_ID**: 與此列中的呼叫相關聯的傳送。

TRANSFER_CD_NODE

傳送中所使用 Connect:Direct 節點的相關資訊。

- **PNODE**: 傳送中的主要節點。
- **SNODE**: 傳送中的次要節點。
- **BRIDGE_IS_PNODE**: 指示作為 Connect:Direct 橋接器一部分的節點的字元。如果此值為 Y, 則主要節點為橋接器節點。如果此值為 N, 則次要節點為橋接器節點。

- **ID**: 此列的 ID。

TRANSFER_CORRELATOR

每一列都包含與傳送項目相關聯的相關性字串及數字。

- **CORRELATION_BOOLEAN**: 布林相關性值。用單一字元表示，Y 表示 true、N 表示 false。
- **CORRELATION_STRING**: 字串相關性值。
- **CORRELATION_NUMBER**: 數字相關性值。
- **ID**: 此列的 ID。

TRANSFER_EVENT

與傳送相關的事件（開始或結束）。

- **ID**: 列 ID。
- **ACTION_TIME**: 傳送動作發生時間。
- **SOURCE_AGENT**: 從其傳送檔案的代理程式的名稱。
- **SOURCE_AGENT_TYPE**: 從其傳送檔案的代理程式的類型。下列為可能值: 1 = STANDARD、2 = BRIDGE、3 = WEB_GATEWAY、4 = EMBEDDED、5 = CD_BRIDGE、6 = SFG。
註: 從 IBM MQ 9.0 開始，Managed File Transfer 不支援「Web 閘道」或 Web 代理程式。
- **SOURCE_QM**: 來源代理程式使用的佇列管理程式。
- **SOURCE_ARCHITECTURE**: 管理來源代理程式的系統的機器架構。
- **SOURCE_OS_NAME**: 來源代理程式機器的作業系統。
- **SOURCE_OS_VERSION**: 來源代理程式機器的作業系統版本。
- **SOURCE_BRIDGE_URL**: 如果來源代理程式是通訊協定橋接器代理程式，則指構成橋接器的資料來源 URL。
- **SOURCE_CD_NODE_ID**: 作為傳送來源的 Connect:Direct 節點。
- **DESTINATION_AGENT**: 向其傳送檔案的代理程式的名稱。
- **DESTINATION_AGENT_TYPE**: 向其傳送檔案的代理程式的類型。下列為可能值: 1 = STANDARD、2 = BRIDGE、3 = WEB_GATEWAY、4 = EMBEDDED、5 = CD_BRIDGE、6 = SFG。
註: 從 IBM MQ 9.0 開始，Managed File Transfer 不支援「Web 閘道」或 Web 代理程式。
- **DESTINATION_QM**: 目的地代理程式使用的佇列管理程式。
- **DESTINATION_BRIDGE_URL**: 如果目的地代理程式是通訊協定橋接器代理程式，則指構成橋接器的資料來源 URL。
- **DESTINATION_CD_NODE_ID**: 作為傳送目的地的 Connect:Direct 節點。
- **ORIGINATOR_HOST**: 從中提交傳送要求的機器的主機名稱。
- **ORIGINATOR_USER**: , 提交傳送要求的使用者的名稱（與 `fteCreateTransfer` 指令所報告的一樣）。
- **ORIGINATOR_MQ_USER**: 提交傳送要求的使用者的名稱（與要求的 IBM MQ 訊息描述子中所包含的一樣）。
- **TRANSFERSET_TIME**: 傳送集的建立時間。
- **TRANSFERSET_SIZE**: 正在傳送的項目數。
- **TRIGGER_LOG**: 對於涉及觸發程式的傳送定義，指是否記載未導致傳送的觸發程式評估。

TRANSFER_EXIT

每一列均代表作為檔案傳送一部分執行的一個傳送結束程式。

- **ID**: 列 ID。

- **EXIT_NAME**: 結束程式的名稱。
- **TRANSFER_ID**: 套用此結束程式的已完成或已取消傳送的 ID。
- **TYPE**: 結束程式的類型。可以是下列其中一個值: *SourceStart*、*SourceEnd*、*DestinationStart* 或 *DestinationEnd*。
- **STATUS**: 結束程式傳回的值。可以是 *cancel* 或 *proceed*。
- **SUPPLEMENT**: 說明結束程式狀態的選用性訊息。

TRANSFER_ITEM

每一列均代表作為傳送一部分傳送的一個檔案。

- **DESTINATION_CHECKSUM_METHOD**: 對目的地檔案進行總和檢查計算所使用的演算法。如果因傳送未順利完成而導致未進行總和檢查計算，則可能為空值。
- **DESTINATION_CHECKSUM_VALUE**: 目的地檔案的總和檢查值。如果總和檢查功能已停用，則該值可能為空值。
- **DESTINATION_ENCODING**: 將目的地檔案作為文字傳送時，目的地檔案使用的字元編碼。
- **DESTINATION_EXISTS_ACTION**: 目的地上存在相同檔案時要執行的動作。
- **DESTINATION_FILE_SIZE**: 目的地所要使用之檔名  或資料集名稱的大小。
- **DESTINATION_FILENAME**: 要在目的地使用的檔名  或資料集名稱。
- **DESTINATION_LINEEND**: 將目的地檔案作為文字傳送時，目的地檔案使用的行尾格式。
- **DESTINATION_MESSAGE_QUEUE_NAME**: 在「檔案轉為訊息」傳送期間，從來源檔案產生的訊息的目的地訊息佇列。
- **DESTINATION_MESSAGE_GROUP_ID**: 在「檔案轉為訊息」傳送期間，從來源檔案產生多則訊息時，用於這些訊息的群組 ID。
- **DESTINATION_MESSAGE_MESSAGE_ID**: 在「檔案轉為訊息」傳送期間，僅從來源檔案產生一則訊息時，該訊息的訊息 ID。
- **DESTINATION_MESSAGE_COUNT**: 在「檔案轉為訊息」傳送期間，將來源檔案分割為訊息的訊息數。
- **DESTINATION_MESSAGE_LENGTH**: 在「檔案轉為訊息」傳送期間，從來源檔案產生訊息的訊息長度（以位元組為單位）。僅當您指定輸出訊息的長度（例如，使用 **fteCreateTransfer** 指令的 **-qs** 選項指定）時才設定此值。如果您指定 **-qs 20K**，且來源檔的大小為 50 KB，則產生的三則訊息大小為 20 KB、20 KB 及 10 KB。在此案例中，**DESTINATION_MESSAGE_LENGTH** 的值設定為 20480。
- **DESTINATION_CORRELATOR_ID**: 目的地的相關性因子資訊的 ID。
- **FILE_MODE**: 檔案傳送模式，例如 *text* 或 *binary*。
- **ID**: 列 ID
- **RESULTCODE**: 指示此項目傳送結果的數值代碼。以下主題中列出此直欄的可能值: 第 288 頁的『傳送中檔案的回覆碼』。這些代碼適用於傳送中的個別項目; 如需整個傳送的結果，請參閱 [TRANSFER.RESULTCODE](#)。
- **RESULT_TEXT**: 傳送結果的文字說明。如果傳送成功，通常為空值。
- **SOURCE_CHECKSUM_METHOD**: 對來源檔案進行總和檢查計算所使用的演算法。
- **SOURCE_CHECKSUM_VALUE**: 來源檔案的總和檢查值。如果總和檢查功能已停用，則該值可能為空值。
- **SOURCE_DISPOSITION**: 傳送完成時要對來源檔案執行的動作。
- **SOURCE_ENCODING**: 將來源檔案作為文字傳送時，來源檔案使用的字元編碼。
- **SOURCE_FILE_SIZE**: 來源所要使用之檔名  或資料集名稱的大小。
- **SOURCE_FILENAME**: 來源檔案名稱  或資料集名稱。
- **SOURCE_LINEEND**: 將來源檔案作為文字傳送時，來源檔案使用的行尾格式。
- **SOURCE_MESSAGE_QUEUE_NAME**: 訊息到檔案傳送的目的地檔案中所包含訊息的來源佇列。

- **SOURCE_MESSAGE_GROUP_ID**: 訊息到檔案傳送的目的地檔案中所包含訊息的群組 ID。
- **SOURCE_MESSAGE_COUNT**: 訊息到檔案傳送的目的地檔案中所包含的訊息數。
- **SOURCE_CORRELATOR_ID**: 來源的相關性因子資訊的 ID。
- **TRANSFER_ID**: 此項目所屬的傳送。
- **TRUNCATE_RECORDS**: 指出太長的資料集記錄是要截斷還是換行。

TRANSFER_STATS

傳送結束時產生的一組統計資料。

- **ID**: 列 ID。
- **TRANSFER_ID**: 統計資料所參照的傳送。
- **START_TIME**: 傳送的開始時間。在忙碌或具有間歇性連線的系統中，此時間可能晚於「已啟動」訊息中所報告的時間，因為該時間代表開始起始處理的時間，而非開始成功傳送資料的時間。
- **RETRY_COUNT**: 因負載或可用性問題而重試傳送的次數。
- **FILE_FAILURES**: 無法傳送的檔案數。
- **FILE_WARNINGS**: 傳送時報告發出警告的檔案數。

TRIGGER_CONDITION

基本 Managed File Transfer 條件式傳送中的一個條件。例如，「`example.file` 檔案已存在」。

- **ID**: 列 ID。
- **TRANSFER_EVENT_ID**: 與觸發程式相關的傳送事件。
- **CONDITION_TYPE**: 觸發程式中使用的檢查類型。例如，檢查檔案是否存在或檔案大小。
- **COMPARISON**: 要執行的特定比較。例如「大於或等於」。
- **VALUE**: 要比較的值。
- **FILENAME**: 要執行的檔名。

相關概念

[第 89 頁的『配置 MFT 日誌程式』](#)

相關參考

[第 447 頁的『fteStartLogger』](#)

fteStartLogger 指令會啟動 Managed File Transfer 記載應用程式。

[第 415 頁的『fteModify 日誌程式 \(以 Windows 服務方式執行 MFT 日誌程式\)』](#)

使用 **fteModifyLogger** 指令來修改 Managed File Transfer 日誌程式，以便它可以 Windows 服務方式執行。您只能在 Windows 平台上使用此指令，必須由 IBM MQ 管理者及 mqm 群組成員的使用者執行，且必須先使用 **fteStopLogger** 指令停止日誌程式。

[第 450 頁的『fteStopLogger』](#)

fteStopLogger 指令會停止 Managed File Transfer 日誌程式。

MFT 日誌程式的權限

執行日誌程式的作業系統使用者需要對日誌程式佇列及 SYSTEM.FTE 主題具有特定 IBM MQ 權限。

執行日誌程式的作業系統使用者需要具有下列 IBM MQ 權限：

- 協調佇列管理程式的 CONNECT 及 INQUIRE。
- 對 SYSTEM.FTE 主題的 SUBSCRIBE 權限。
- 對 SYSTEM.FTE.LOG.RJCT.*logger_name* 佇列的 PUT 權限。
- 對 SYSTEM.FTE.LOG.CMD.*logger_name* 佇列的 GET 權限。

相關參考

[第 296 頁的『限制 MFT 特定資源的群組權限』](#)

您可以為管理 Managed File Transfer 存取控制目的建立兩個安全群組：FTEUSER 及 FTEAGENT，而非針對可能涉及各種不同物件將權限授與個別使用者。IBM MQ 管理者負責建立及移入這些群組。管理者可以選擇延伸或修改這裡所說明的提議配置。

第 301 頁的『限制 MFT 代理程式動作的使用者權限』

除了使用群組管理對資源的存取外，您還可以啟用額外的一層安全來限制使用者可以採取的 Managed File Transfer 代理程式動作。將代理程式權限佇列的權限授與使用者，使使用者有權執行特定的代理程式動作。

目的地檔案的檔案許可權

Managed File Transfer 目的地代理程式寫入之目的地檔案的檔案許可權由代理程式執行所在的平台決定。

z/OS、UNIX 及 Linux 平台上的目的地代理程式



您需要變更系統上 **umask** 的值。

例如，假設 z/OS 系統上使用 ID 的預設 **umask** 值為 **0022**。

當 MFT 代理程式以這個使用者身分執行，並寫入目的地檔案時，檔案具有下列許可權：

```
-IW-I--I--
```

如果您將 **umask** 值變更為 (例如，**0006**)，請執行下列指令：

```
umask 0006
```

且代理程式已重新啟動，則代理程式寫入的任何目的地檔案都具有許可權：

```
-IW-IW----
```

請注意，在執行 **umask** 指令之後，您必須重新啟動代理程式，代理程式才能取得新值。

雖然這裡使用 z/OS 作為範例，但相同的資訊適用於 UNIX 及 Linux 平台。

Windows 上的目的地代理程式



依預設，許可權會從根資料夾繼承至其下的檔案及子資料夾，但可以關閉此繼承。

您的 Windows 管理者或網域管理者應該檢閱及管理許可權，並在必要時變更它們。他們可以使用 [icalcs](#) 指令來檢視、新增、更新及移除許可權。

相關資訊

[限制 MFT 特定資源的群組權限](#)

[限制 MFT 代理程式動作的使用者權限](#)

寫入目的地佇列的訊息上由 MFT 設定的 MQ 訊息內容

從檔案傳送至訊息時，Managed File Transfer 可對第一個寫入至目的地佇列的訊息設定 IBM MQ 訊息內容。當「檔案轉為訊息」的傳送失敗時，會設定其他 IBM MQ 訊息內容。

IBM MQ 訊息內容可讓應用程式選取要處理的訊息，或在未存取 MQ 訊息描述子 (MQMD) 或 MQRFH2 標頭的情況下擷取訊息的相關資訊。請參閱[訊息內容](#)。

本主題說明 **fteCreateTransfer** 和 **fteCreateTemplate** 指令中使用的參數，指出應該將訊息內容新增至寫入目的地佇列的第一則訊息。您也可以使用 **fte:filespec** 參數的 [dstmsgprop](#) 值，指定應該將訊息內容新增至寫入目的地佇列的第一則訊息。

標準內容

您可以在 `fteCreateTransfer` 指令或 `fteCreateTemplate` 指令上使用 `-qmp` 參數，來指定是否在由傳送寫入目的地佇列的第一則訊息上設定 IBM MQ 訊息內容。如需如何使用此參數的範例，請參閱第 168 頁的『範例：設定有關「檔案轉為訊息」傳送的 IBM MQ 訊息內容』主題

IBM MQ 訊息內容包含傳送 meta 資料。訊息內容的名稱前面會加上 `usr.WMQFTE` 字首。`usr.` 字首會讓這些訊息內容可供 JMS 應用程式使用。

`usr.WMQFTETransferId`

唯一的十六進位傳送 ID。

`usr.WMQFTETransferMode`

檔案傳送的類型：二進位模式或文字模式。

`usr.WMQFTESourceAgent`

來源代理程式的名稱。

`usr.WMQFTEDestinationAgent`

目的地代理程式的名稱。

`usr.WMQFTEFileName`

來源檔案的名稱。

`usr.WMQFTEFileSize`

來源檔案的大小（以位元組為單位）。

`usr.WMQFTEFileLastModified`

來源檔案的前次修改時間。此值的單位是毫秒，自世界標準時間 1970 年 1 月 1 日 00:00:00 起算。

`usr.WMQFTEFileIndex`

所傳送的檔案清單中的現行檔案索引。清單中第一個檔案的索引為 0。

`usr.WMQFTEMqmdUser`

提交傳送要求的使用者的 MQMD 使用者 ID。

失敗內容

如果「檔案轉為訊息」傳送在目的地代理程式至少已將一個訊息寫入目的地佇列之後失敗，則 Managed File Transfer 會將空白訊息寫入目的地佇列。如果 `-qmp` 參數設為 `true`，則此空白訊息會有兩個 IBM MQ 訊息內容集。如需「檔案轉為訊息」傳送失敗的範例，請參閱第 171 頁的『檔案轉為訊息傳送失敗』。

當「檔案轉為訊息」傳送完全失敗時，Managed File Transfer 會將空白訊息寫入目的地佇列。如果 `-qmp` 參數設為 `true`，且訊息資料的長度大於 `maxInputOutputMessageLength` 值，則指令行上會顯示下列錯誤訊息。

```
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 訊息內容包含失敗的相關資訊。如同標準訊息內容，訊息內容的名稱前面也會加上 `usr.WMQFTE` 字首，可供 JMS 應用程式使用。

`usr.WMQFTEReturnCode`

傳送的回覆碼。如需此回覆碼的可能值清單，請參閱第 284 頁的『MFT 的回覆碼』主題。

`usr.WMQFTESupplement`

詳細說明傳送失敗原因的增補訊息。

使用者定義內容

在 `fteCreateTransfer` 指令中使用 `-md` 參數指定的 meta 資料可以設為 IBM MQ 訊息內容。如果 `-qmp` 參數設為 `true`，則使用者所指定的任何 meta 資料都會新增至第一個訊息的訊息標頭上。

meta 資料名稱前面會加上 **usr.** 字首。比方說，如果 meta 資料是 department=accounts，則 IBM MQ 訊息標頭會設為 **usr.department=accounts**。

您無法使用 meta 資料指定以 **usr.WMQFTE** 或 **usr.com.ibm.wmqfte** 為開頭的標頭。如果您使用以 **WMQFTE** 或 **com.ibm.wmqfte** 為開頭的名稱來指定 meta 資料，此 meta 資料將不會用於訊息內容中，而會被忽略。

相關概念

第 164 頁的『將資料從檔案傳送至訊息』

您可以使用 Managed File Transfer 的「檔案轉為訊息」特性，將資料從一個檔案傳送至 IBM MQ 佇列上的單一訊息或多則訊息。

相關工作

第 168 頁的『範例：設定有關「檔案轉為訊息」傳送的 IBM MQ 訊息內容』

您可以在 **fteCreateTransfer** 指令上使用 **-qmp** 參數，以指定是否在傳送寫入目的地佇列的第一則訊息上設定 IBM MQ 訊息內容。IBM MQ 訊息內容可讓應用程式選取要處理的訊息，或在存取 IBM MQ 訊息描述子 (MQMD) 或 MQRFH2 標頭的情況下擷取訊息的相關資訊。

相關參考

第 571 頁的『MFT 從來源佇列上的訊息讀取的 IBM MQ 訊息內容』

在「訊息轉為檔案」傳送中從來源佇列讀取訊息的代理程式，會從訊息中讀取 IBM MQ 訊息內容。這些內容的值可用來決定傳送的行為。

第 284 頁的『MFT 的回覆碼』

Managed File Transfer 指令、Ant 作業及日誌訊息提供回覆碼，指出功能是否順利完成。

第 171 頁的『檔案轉為訊息傳送失敗』



在代理程式已啟動將檔案資料寫入目的地佇列作業後，如果「檔案轉為訊息」傳送失敗，代理程式會將一則訊息寫入佇列，以指示使用訊息的應用程式發生失敗。

第 367 頁的『fteCreateTransfer: 啟動新的檔案傳送』

fteCreateTransfer 指令從指令行建立並啟動新的檔案傳送。此指令可立即啟動檔案傳送、將檔案傳送排定在未來的時間和日期、將排定的傳送重複一次以上，以及根據特定條件觸發檔案傳送。

第 744 頁的『fte:filespec Ant 巢狀元素』

fte:filespec 參數在其他作業中用作巢狀元素。使用 **fte:filespec** 說明一個以上來源檔案、目錄

 或資料集與目的地之間的對映。表示要移動或複製的一組檔案、目錄  或是資料集時，通常會使用此元素。

MFT 從來源佇列上的訊息讀取的 IBM MQ 訊息內容

在「訊息轉為檔案」傳送中從來源佇列讀取訊息的代理程式，會從訊息中讀取 IBM MQ 訊息內容。這些內容的值可用來決定傳送的行為。

用來取消訊息到檔案傳送的標頭

對群組中的最後一個訊息設定下列 IBM MQ 訊息內容，以取消該群組的「訊息轉為檔案」傳送：

usr.UserReturnCode

必要項目。傳送的回覆碼。當此標頭設為非零值時，表示要取消傳送。

usr.UserSupplement

選用項目。說明取消傳送之原因的文字。

如果「訊息轉為檔案」傳送的來源代理程式從 **usr.UserReturnCode** 訊息內容設為非零值的來源佇列中讀取訊息，它將會停止從該佇列讀取訊息，並在傳送日誌 XML 中報告傳送失敗。傳送日誌 XML 包含回覆碼以及在訊息標頭中設定的增補文字。如果目的地代理程式已將資料寫入暫存檔，則會從目的地中刪除此檔案。

變數替代所使用的標頭

在第一個要從受監視佇列讀取的訊息中，任何 IBM MQ 訊息內容的值都可代入作業 XML 定義中。使用者定義訊息內容的字首為 **usr.**，但在變數名稱中不包括此字首。變數名稱前面必須有錢幣符號 (\$) 字元，並以大括弧 ({}) 括住。例如，**\${destFileName}** 會取代為要從來源佇列讀取之第一則訊息的 **usr.destFileName** 訊息內容值。

例如，將訊息放置到受監視佇列上的使用者或程式，可以對群組中的第一個訊息設定 IBM MQ 訊息內容，以指定要使用哪個代理程式作為檔案傳送的目的地，以及資料傳送目標的檔案名稱。

如需相關資訊，請參閱 [第 154 頁的『監視佇列及使用變數替代』](#)。

設定與訊息大小相關聯的 MQ 屬性及 MFT 內容的指引

您可以變更 IBM MQ 屬性及 Managed File Transfer 內容，以影響 Managed File Transfer 在讀取或寫入不同大小訊息時的行為。

如果從來源佇列讀取或寫入目的地佇列的訊息大小超過 1048576 個位元組 (1 MB)，您必須將 Managed File Transfer Agent 內容 `maxInputOutputMessageLength` 的值增加到大於或等於要讀取或寫入的訊息大小上限的值。

如果來源佇列上的訊息大於 1048576 個位元組，您必須設定來源代理程式的 `maxInputOutputMessageLength` 內容。如果目的地佇列上的訊息大於 1048576 個位元組，您必須設定目的地代理程式的 `maxInputOutputMessageLength` 內容。如需 `maxInputOutputMessageLength` 內容的相關資訊，請參閱 [進階代理程式內容](#)。

- 如果代理程式寫入或從中讀取的佇列位於代理程式佇列管理程式的本端位置，您必須變更 IBM MQ 佇列管理程式、佇列及通道 `MAXMSGL` 屬性。

請確定來源或目的地佇列的訊息大小上限值大於或等於 `maxInputOutputMessageLength` 代理程式內容的值。

請確定下列每個 IBM MQ 屬性的值（以位元組為單位）：

- 代理程式佇列管理程式的訊息大小上限
- `SYSTEM.FTE.STATE.agent_name` 佇列的訊息大小上限
- 用戶端通道的訊息大小上限（如果您的代理程式以用戶端模式連接至佇列管理程式）

大於或等於下列計算的結果：

若為「檔案轉為訊息」傳送（支援的檔案大小上限為 100 MB）：
`maxInputOutputMessageLength` 的值

若為「訊息轉為檔案」傳送：

$$\text{值} = 3 * (\text{maxInputOutputMessageLength}) + 1048576$$

（此計算衍生自三個檢查點可以儲存在狀態訊息中的事實，並且每一個檢查點可能必須緩衝達資料訊息量的大小上限。）

- 如果代理程式要寫入的目標佇列為遠端佇列，您必須變更 IBM MQ 佇列管理程式、佇列及通道 `MAXMSGL` 屬性。

請確定下列每個 IBM MQ 屬性的值大於或等於 `maxInputOutputMessageLength` 代理程式內容的值：

- 代理程式佇列管理程式上的遠端佇列管理程式傳輸佇列的訊息大小上限
- 代理程式佇列管理程式至遠端佇列管理程式之通道的訊息大小上限
- 遠端佇列管理程式上的目的地佇列的訊息大小上限
- 遠端佇列管理程式的訊息大小上限

請確定下列每個 IBM MQ 屬性的值（以位元組為單位）：

- 代理程式佇列管理程式的訊息大小上限
- `SYSTEM.FTE.STATE.agent_name` 佇列的訊息大小上限
- 用戶端通道的訊息大小上限（如果您的代理程式以用戶端模式連接至佇列管理程式）

大於或等於下列計算的結果：

若為「檔案轉為訊息」傳送（支援的檔案大小上限為 100 MB）：
`maxInputOutputMessageLength` 的值

若為「訊息轉為檔案」傳送：

$$\text{值} = 3 * (\text{maxInputOutputMessageLength}) + 1048576$$

(此計算衍生自三個檢查點可以儲存在狀態訊息中的事實，並且每一個檢查點可能必須緩衝達資料訊息量的大小上限。)

如果超出其中一個內容的值，代理程式即會停止，並在代理程式事件日誌中記載下列錯誤：

```
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
```

代理程式事件日誌的上述訊息中可能會包括下列 IBM MQ 原因碼：

- **rc=2010** 此原因碼對映至 **MQRC_DATA_LENGTH_ERROR**，表示已超出用戶端通道的訊息大小上限值。若要解決此問題，請確定代理程式佇列管理程式的用戶端通道訊息大小上限大於或等於下列計算的結果：

$$3 * (\text{maxInputOutputMessageLength}) + 1048576$$

- **rc=2030** 此原因碼對映至 **MQRC_MSG_TOO_BIG_FOR_Q**，表示已超出 **SYSTEM.FTE.STATE.agent_name** 佇列的訊息大小上限值。若要解決此問題，請確定 **SYSTEM.FTE.STATE.agent_name** 佇列的訊息大小上限大於或等於下列計算的結果：

$$3 * (\text{maxInputOutputMessageLength}) + 1048576$$

- **rc=2031** 此原因碼對映至 **MQRC_MSG_TOO_BIG_FOR_Q_MGR**，表示已超出代理程式佇列管理程式的訊息大小上限值。若要解決此問題，請確定代理程式佇列管理程式的訊息大小上限大於或等於下列計算的結果：

$$3 * (\text{maxInputOutputMessageLength}) + 1048576$$

如果您要傳送許多小型訊息

如果代理程式對佇列讀取或寫入的訊息平均大小小於 1310 個位元組，且代理程式所讀取或寫入的訊息數超過 10000 個，您必須增加佇列管理程式的「未確定的訊息數量上限」屬性值，或減少一個檢查點間隔內的資料量。

當代理程式對佇列讀取訊息或寫入訊息時，會將對應的 **GET** 或 **PUT** 歸類到交易中。一個交易中的 **GET** 或 **PUT** 數目，取決於處理一個檢查點間隔內所有資料時所需的數量。一個檢查點間隔內的資料量概數可透過下列計算從代理程式內容來判斷：

$$\text{Checkpoint interval data size (in bytes)} = \text{agentCheckpointInterval} * \text{agentFrameSize} * \text{agentWindowSize} * \text{agentChunkSize}.$$

預設檢查點資料大小是 $1 * 5 * 10 * 262144$ 個位元組 = 13107200 個位元組 (12.5MB)。佇列管理程式在一個交易中支援的未確定訊息數目上限，由 **MaxUncommittedMsgs** 佇列管理程式屬性控制。此屬性的預設值是 10000 個訊息。如果平均訊息大小約小於 1310 個位元組，則在寫入的訊息數超過 10000 個時，即會超出預設的未確定訊息數目上限。

如果超出 **MaxUncommittedMsgs** 上限，代理程式即會停止，並在代理程式事件日誌中記載下列錯誤：

```
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.
```

原因碼 2024 對映至：**MQRC_SYNCPOINT_LIMIT_REACHED**。

若要解決此問題，請執行下列其中一個動作：

- 對從中讀取或寫入至佇列的代理程式所連接的佇列管理程式，增加其 **MaxUncommittedMsgs** 佇列管理程式屬性的值。請參閱 [MaxUncommittedMsgs \(MQLONG\)](#)。
- 減少一個檢查點間隔中的資料量。若要這麼做，請降低下列一個以上代理程式內容的值：
 - agentCheckpointInterval
 - agentFrameSize
 - agentWindowSize
 - agentChunkSize

如需這些代理程式內容的相關資訊，請參閱[進階代理程式內容](#)。

如果您要持續將訊息寫入佇列

如果您要傳送至佇列並持續將訊息寫入佇列，您可能必須增加佇列管理程式日誌檔空間的大小，才能記載一個檢查點間隔中的所有資料。

如果超出佇列管理程式日誌檔空間，代理程式即會停止，並在代理程式事件日誌中記載下列錯誤：

```
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
```

原因碼 '2102' 對映至：MQRC_RESOURCE_PROBLEM。

若要解決此問題，請增加目的地代理程式佇列管理程式日誌檔空間的大小。

指定「訊息轉為檔案」傳送等待時間的指引

在指定「訊息轉為檔案」傳送時，您可以選擇性地使用 **-sqwt** 參數來指定傳送的等待時間。**-sqwt** 的值表示來源代理程式等待訊息出現在來源佇列上（如果來源佇列是空的或已變成空的），或等待完整群組出現在來源佇列上（如果已指定 **-sqgi** 屬性）的時間量。

本主題說明 **fteCreateTransfer** 指令中用於指定等待時間的參數。您也可以使用 **fte:filespec** 參數的 **srcqueuetimeout** 值來指定等待時間。

如果 **-sqwt** 參數值大於或等於目的地代理程式等待來源代理程式完成傳送的時間量，傳送則不會完成。目的地代理程式等待傳送完成的時間量可透過下列計算得出：

```
transferAckTimeout * transferAckTimeoutRetries
```

transferAckTimeout 及 **transferAckTimeoutRetries** 內容會設定在目的地代理程式 **agent.properties** 檔中。如需這些代理程式內容的相關資訊，請參閱 [第 463 頁的『MFT agent.properties 檔案』](#)。

若要避免傳送無法完成，您必須執行下列其中一個步驟：

- 降低 **-sqwt** 參數的值，使其小於目的地代理程式 **transferAckTimeout** 內容的值。

註：**transferAckTimeout** 內容的預設值是 60,000 毫秒。**-sqwt** 參數的值是以秒為單位指定的，請將此值設為 59 或更低的值。
- 增加目的地代理程式 **transferAckTimeout** 內容的值，使其大於 **-sqwt** 參數的值。

註：**transferAckTimeout** 內容的值是以毫秒為單位指定的。**-sqwt** 參數的值是以秒為單位指定的。

相關參考

[第 367 頁的『fteCreateTransfer: 啟動新的檔案傳送』](#)

fteCreateTransfer 指令從指令行建立並啟動新的檔案傳送。此指令可立即啟動檔案傳送、將檔案傳送排定在未來的時間和日期、將排定的傳送重複一次以上，以及根據特定條件觸發檔案傳送。

[第 463 頁的『MFT agent.properties 檔案』](#)

每一個 Managed File Transfer Agent 都有自己的內容檔 **agent.properties**，其中必須包含代理程式用來連接至其佇列管理程式的資訊。**agent.properties** 檔案也可以包含變更代理程式行為的內容。

第 744 頁的『fte:filespec Ant 巢狀元素』

fte:filespec 參數在其他作業中用作巢狀元素。使用 **fte:filespec** 說明一個以上來源檔案、目錄

z/OS 或資料集與目的地之間的對映。表示要移動或複製的一組檔案、目錄 **z/OS** 或是資料集時，通常會使用此元素。

MFT 的可用字碼頁

本參考主題列出了適用於在 Managed File Transfer 支援的各種平台上，進行文字檔轉換的所有字元編碼格式。

共用編碼

這些字元編碼格式適用於所有支援的平台。如果來源檔案是使用下表中的其中一種格式編碼，但您要使用下表中的另一種格式來撰寫目的地檔案，則可執行此作業，而絲毫不用考量平台。您可以使用標準名稱或任何別名來指定編碼格式。

標準名稱	別名
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

標準名稱	別名
IBM-1383	Cp1383, ibm1383
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	csipc852, ibm852, Cp852
IBM-850	Cp850, ibm850, csipc850multilingual
IBM-838	Cp838, ibm838
IBM-834	Cp834, ibm834
IBM-775	ibm775, Cp775
IBM-737	Cp737, ibm737
IBM-500	Cp500, ibm500
IBM-437	ibm437, Cp437, csipc8codepage437
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

標準名稱	別名
IBM-870	Cp870, ibm870
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

標準名稱	別名
MacCyrillic	ibm-1283
MacGreek	ibm-1280
MacIceland	ibm-1286
MacRoman	ibm-1275
MacRomania	ibm-1285
MacSymbol	Adobe-Symbol-Encoding、ibm-1038
MacTurkish	ibm-1281

來源平台預設編碼

如果您沒有指定來源檔案或目的地檔案的編碼，則會使用該平台的預設編碼。轉換是由目的地代理程式執行，因此，來源及目的地編碼在目的地代理程式平台上必須皆受支援，才能進行轉換。目的地預設編碼在目的地代理程式上一律會受支援，因此，將此編碼保留成未指定，一律是安全的。但是，使用預設來源編碼可能會不安全，因為目的地代理程式可能會不支援來源編碼的預設值。

如果您使用的是預設來源編碼，則應使用本主題中的表格，來確保該組合將受支援。

平台	預設編碼
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

平台專用編碼

附註：下列兩個表格包含相同的資訊。此資訊以兩種不同方式編排，以協助您尋找正確的資訊（視您依平台還是依編碼進行查閱而定）。

依平台排序的編碼



標準名稱以粗體列出，後面接以括弧括住的別名。


僅支援在「共用編碼」表格中列出的編碼之平台，在這裡並未列出。

平台	支援的編碼 (不在「共用編碼」表格中)
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	支援的編碼 (不在「共用編碼」表格中)
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, I8, 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) X-UnicodeLittle (UnicodeLittle) X-UnicodeBig (UnicodeBig) IBM-859 (Cp859, ibm859) </p>

平台	支援的編碼 (不在「共用編碼」表格中)
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	支援的編碼 (不在「共用編碼」表格中)
<p>   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	支援的編碼 (不在「共用編碼」表格中)
 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	支援的編碼 (不在「共用編碼」表格中)
HP-UX (Itanium)	<p> UTF-16 (UTF16, Unicode, UTF_16, UCS-2) MacUkraine () MacThai () MacHebrew () MacDingbat () JISO208 () JISO201 () IBM-949C (Cp949C, ibm949C) IBM-943C (ibm943C, Cp943C) IBM-942C (Cp942C, ibm942C) IBM00858 (cp858, ccsid00858, 858, cp00858) IBM01140 (ccsid01140, cp01140, 1140, cp1140) x-eucJP-Open (EUC_JP_Solaris, eucJP-open) x-IBM33722 (ibm33722, 33722, ibm-33722_vascii_vpua, ibm-5050, ibm-33722, cp33722) x-IBM930 (cp930, ibm930, ibm-930, 930) x-IBM939 (ibm-939, ibm939, cp939, 939) x-IBM964 (964, cp964, ibm-964, ibm964) x-ISO-2022-CN-CNS (ISO-2022-CN-CNS, ISO2022CN_CNS) x-iso-8859-11 (iso-8859-11, iso8859_11) x-JISAutoDetect (JISAutoDetect) x-MS950-HKSCS (MS950_HKSCS) x-PCK (pck) x-windows-50220 (cp50220, ms50220) x-windows-50221 (ms50221, cp50221) x-windows-iso2022jp (windows-iso2022jp) x-Big5-Solaris (Big5_Solaris) IBM918 (cp918, ebcdic-cp-ar2, ibm-918, 918) IBM1047 (cp1047, 1047, ibm-1047) IBM01149 (cp1149, cp01149, ccsid01149, 1149) IBM01148 (cp1148, ccsid01148, 1148, cp01148) IBM01147 (ccsid01147, cp1147, 1147, cp01147) IBM01146 (ccsid01146, cp01146, cp1146, 1146) IBM01145 (cp1145, cp01145, ccsid01145, 1145) IBM01144 (cp01144, cp1144, ccsid01144, 1144) IBM01143 (cp01143, 1143, ccsid01143, cp1143) IBM01142 (cp01142, cp1142, 1142, ccsid01142) IBM01141 (cp1141, ccsid01141, cp01141, 1141) </p>

平台	支援的編碼 (不在「共用編碼」表格中)
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, ccsid00858, 858, cp00858) X-UnicodeLittle (UnicodeLittle) X-UnicodeBig (UnicodeBig) IBM-859 (Cp859, ibm859) </p>

平台	支援的編碼 (不在「共用編碼」表格中)
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	支援的編碼 (不在「共用編碼」表格中)
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) JIS0208 () JIS0201 () 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	支援的編碼 (不在「共用編碼」表格中)
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	支援的編碼 (不在「共用編碼」表格中)
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) X-UnicodeBig (UnicodeBig) IBM-859 (Cp859, ibm859) IBM-837 (ibm837, Cp837) </p>

平台	支援的編碼 (不在「共用編碼」表格中)
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	支援的編碼 (不在「共用編碼」表格中)
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, ccsid00858, 858, cp00858) X-UnicodeLittle (UnicodeLittle) X-UnicodeBig (UnicodeBig) IBM-859 (Cp859, ibm859) </p>

平台	支援的編碼 (不在「共用編碼」表格中)
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	支援的編碼 (不在「共用編碼」表格中)
<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) JIS0208 () JIS0201 () 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	支援的編碼 (不在「共用編碼」表格中)
  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	支援的編碼 (不在「共用編碼」表格中)
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) JIS0208 () JIS0201 () 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	支援的編碼 (不在「共用編碼」表格中)
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	支援的編碼 (不在「共用編碼」表格中)
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>

依編碼排序的平台

編碼	別名	支援此編碼的平台
x-MacUkraine	macukraine	Solaris, SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x

編碼	別名	支援此編碼的平台
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	mac hebrew	Solaris, SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
x-MacDingbat	mac dingbat	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
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

編碼	別名	支援此編碼的平台
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
  IBM-924_LF	Cp924_LF, ibm924_LF	IBM i






編碼	別名	支援此編碼的平台
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)
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)












編碼	別名	支援此編碼的平台
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)
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)











編碼	別名	支援此編碼的平台
x-ISO-2022-CN-CNS	ISO-2022-CN-CNS, ISO2022CN_CNS	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z 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)
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)











編碼	別名	支援此編碼的平台
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)
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)











編碼	別名	支援此編碼的平台
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)
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)











編碼	別名	支援此編碼的平台
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)
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)









編碼	別名	支援此編碼的平台
IBM-864S	ibm864S, Cp864S	 IBM i,  AIX,  z/OS, Linux on POWER Systems - Big Endian
IBM-859	Cp859, ibm859	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z 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
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









編碼	別名	支援此編碼的平台
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
IBM-420S	Cp420S, ibm420S	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian

















編碼	別名	支援此編碼的平台
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
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













編碼	別名	支援此編碼的平台
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
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











編碼	別名	支援此編碼的平台
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
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











編碼	別名	支援此編碼的平台
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
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)













編碼	別名	支援此編碼的平台
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)
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













編碼	別名	支援此編碼的平台
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
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











編碼	別名	支援此編碼的平台
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
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













編碼	別名	支援此編碼的平台
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
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













編碼	別名	支援此編碼的平台
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
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









編碼	別名	支援此編碼的平台
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)
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



編碼	別名	支援此編碼的平台
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
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

編碼	別名	支援此編碼的平台
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
JIS0201		 IBM i, HP-UX (Itanium), AIX,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)

編碼	別名	支援此編碼的平台
JIS0208		 IBM i, HP-UX (Itanium), AIX,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
Johab	x-johab	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian
KOI8-RU	ibm-1168, koi8_ru	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z 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
MacDingbat		 IBM i, HP-UX (Itanium), AIX,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)

編碼	別名	支援此編碼的平台
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
UTF-16	UTF16、Unicode、UTF_16、UCS-2	 IBM i, HP-UX (Itanium), AIX,  z/OS, Linux on POWER Systems - Big Endian

編碼	別名	支援此編碼的平台
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)
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

編碼	別名	支援此編碼的平台
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

相關概念

第 133 頁的『使用傳送定義檔』

您可以指定傳送定義檔，它可用來建立檔案傳送。此傳送定義檔是 XML 檔，它定義建立傳送需要的部分或全部資訊。

相關參考

第 540 頁的『使用 MFT 傳送文字檔』

文字檔傳送涉及將檔案的字碼頁從一種字碼頁轉換成另一種字碼頁。文字檔傳送也涉及在系統之間轉換 CRLF（歸位-換行）字元。本主題彙總 Managed File Transfer 的文字檔傳送行為。

第 367 頁的『fteCreateTransfer: 啟動新的檔案傳送』

fteCreateTransfer 指令從指令行建立並啟動新的檔案傳送。此指令可立即啟動檔案傳送、將檔案傳送排定在未來的時間和日期、將排定的傳送重複一次以上，以及根據特定條件觸發檔案傳送。

MFT 代理程式如何使用 Java 資料堆及原生資料堆記憶體

IBM MQ Managed File Transfer 代理程式以 Java 處理程序執行。因此，代理程式會在 Java 虛擬機器 (JVM) 的虛擬化環境中執行。

JVM 本身是原生程序，受硬體和作業系統限制。JVM 維護兩個記憶體區域：

- Java 資料堆

這包含 Java 物件的實例，並由記憶體回收處理管理。在 JVM 啟動期間，會使用 **-Xmx** JVM 選項來配置 Java 資料堆的大小上限。

- 原生資料堆

原生資料堆包含 JVM 本身的資源；例如，即時編譯器、類別及 ClassLoaders。

代理程式主要使用 Java 資料堆。執行受管理傳送時，代理程式會使用 Java 資料堆來建立傳送所需的 Java 物件。代理程式讀取至緩衝區的任何檔案資料也會儲存在 Java 資料堆記憶體中。

代理程式本身不包含任何使用原生資料堆的程式碼。不過，在 Java 訊息佇列作業介面 (JMQUI) 中，代理程式會使用原生程式碼來與其代理程式佇列管理程式進行通訊。

當代理程式使用 BINDINGS 傳輸來連接至其代理程式佇列管理程式時，會使用這個原生程式碼。這是本端共用記憶體連線（有時稱為交互程序通訊或 IPC），而不是代理程式使用 CLIENT 傳輸連接時使用的 TCP/IP 連線。當代理程式配置為使用 BINDINGS 傳輸時，會使用原生資料堆在代理程式與代理程式佇列管理程式之間傳遞訊息及指令。

這表示與使用 CLIENT 傳輸連接的對等代理程式相比，使用 BINDINGS 傳輸連接至其代理程式佇列管理程式的高負載代理程式會更廣泛使用原生資料堆。

一個常見的誤解是代理程式的 Java 資料堆必須等於（或大於）要傳送的最大檔案大小。這不正確，因為暫置將檔案資料讀入記憶體。

作為指引，可以粗略計算用於儲存每次傳送的檔案資料的 Java 資料堆數量上限，如下所示：

```
Memory allocated for a transfer = agentCheckpointInterval *
agentFrameSize * agentWindowSize * agentChunkSize
```

Java 資料堆及原生資料堆使用情形如何影響代理程式

當 `java.lang.OutOfMemoryError` 發生時，您可能認為使用 `-Xmx` Java 系統內容來增加應用程式可用的 Java 資料堆數量是合理的。例如，下列內容設定會嘗試配置 Java 資料堆大小上限 2GB：

```
-Xmx2048M
```

不過，由於原生資料堆耗盡，為應用程式配置太多 Java 資料堆可能會導致 `java.lang.OutOfMemoryError` 發生。這是因為隨著 Java 資料堆空間增長，原生資料堆必須收縮以容納它。

如需如何防止原生資料堆耗盡所導致的 `java.lang.OutOfMemoryErrors` 的相關資訊，請參閱 [如果 MFT 代理程式 ABENDSS 由於原生記憶體耗盡而具有 `java.lang.OutOfMemoryError`，怎麼辦。](#)

MFT 所使用的 XML 訊息格式

Managed File Transfer 使用 XML 格式的訊息有許多用途：用來指令代理程式；記載監視器、排程及傳送的相關資訊；以及定義用於配置的資訊。XML 格式在這些用途下的邏輯結構，由 XML 綱目進行說明。

各版本的 Managed File Transfer 分別會使用一個 XML 綱目來驗證以 XML 寫入的訊息。代理程式會擷取 XML 綱目版本，並判斷此綱目是否受支援。

安裝 Managed File Transfer 之後，您可以在下列目錄中找到 Managed File Transfer 訊息綱目檔：
`MQ_INSTALLATION_PATH/mqft/samples/schema`。其中包括下列綱目：

可放置在代理程式指令佇列上的 XML 訊息綱目

- `FileTransfer.xsd`
- `Internal.xsd`
- `Monitor.xsd`
- `PingAgent.xsd`

如需將 XML 訊息放置在代理程式指令佇列上的相關資訊，請參閱第 227 頁的『將訊息放置在代理程式指令佇列上以控制 MFT』。

發佈至 **SYSTEM.FTE** 主題的 XML 訊息綱目

- `MonitorList.xsd`
- `MonitorLog.xsd`
- `ScheduleList.xsd`
- `ScheduleLog.xsd`
- `TransferLog.xsd`
- `TransferStatus.xsd`

如需發佈至 **SYSTEM.FTE** 主題的 XML 訊息及 **SYSTEM.FTE** 主題結構的相關資訊，請參閱第 521 頁的『**SYSTEM.FTE** 主題』。

Managed File Transfer 所使用的其他綱目

`fteutils.xsd`。此綱目包含一般元素定義，並由部分其他綱目所包含。

- `Notification.xsd`
- `ProtocolBridgeCredentials.xsd`
- `ProtocolBridgeProperties.xsd`
- `ConnectDirectCredentials.xsd`
- `ConnectDirectNodeProperties.xsd`
- `ConnectDirectProcessDefinitions.xsd`
- `Reply.xsd`
- `UserSandboxes.xsd`

MFT 代理程式狀態訊息格式

建立或啟動 Managed File Transfer Agent 時，代理程式會將其詳細資料發佈至協調佇列管理程式上的 SYSTEM.FTE 主題（在 SYSTEM.FTE/Agents/agent name 主題上）。

包含下列資訊：

- 代理程式名稱
- 執行代理程式所在的平台
- 代理程式說明（如果已提供）
- 代理程式的佇列管理程式
- 執行代理程式所在的時區
- 代理程式版本
- 代理程式傳送限制
- 代理程式的每一項現行傳送的狀態。[代理程式傳送狀態](#)中列出了這些狀態。
- 代理程式的類型

如果代理程式是通訊協定橋接器代理程式，則還包含下列資訊：

- 通訊協定橋接器代理程式的類型
- 通訊協定橋接器伺服器的主機名稱或 IP 位址

每當代理程式傳送狀態變更時，會重新發佈代理程式狀態，但依預設間隔不會超過 30 秒。您可以使用 `agentStatusPublishRateLimit` 代理程式內容變更此預設值，[進階代理程式內容](#)中說明了此內容。

下列範例輸出顯示用於代理程式狀態中每一個資料元素的鍵：

```
<?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>
```

下列範例輸出顯示用於通訊協定橋接器代理程式的代理程式狀態中每一個資料元素的鍵：

```
<?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>
</properties>
```

```
<entry key="SourceTransferStates">414d51204d554e474f20202020202020d857374a93a72622=NegotiatingTransfer
</entry>
</properties>
```

MFT 代理程式傳送狀態

啟動的 Managed File Transfer Agent 會將其詳細資料發佈至協調佇列管理程式上的 SYSTEM.FTE 主題。這些詳細資料包括涉及該代理程式的每一個現行傳送的狀態。

傳送狀態	說明
NewSenderTransfer	來自尚未啟動協議的來源代理程式的新建傳送。
NewReceiverTransfer	已在協議過程中於目的地代理程式上建立新建傳送，但尚未執行此傳送。
NegotiatingTransfer	在執行傳送之前，來源代理程式正在與目的地代理程式協議。
RunningTransfer	來自處於正常執行中狀態的來源代理程式或目的地代理程式的傳送
RecoveringTransfer	在來源或目的地代理程式啟動回復程序時，執行中狀態的任何傳送都會移至傳送狀態。在重新同步訊息傳送至對等代理程式時，傳送會從此狀態移出至 ReSynchronisingTransfer 狀態。 比方說，例如目的地代理程式對執行中傳送啟動回復程序，則在重新同步訊息傳送至其來源代理程式時，傳送會移至 ReSynchronisingTransfer 狀態。
ReSynchronisingTransfer	傳送來源或目的地代理程式發現問題，並已將重新同步訊息傳送至個別目的地或來源代理程式。
CompletedTransfer	目的地代理程式已完成傳送，並已將完成訊息傳送至來源代理程式。目的地代理程式正在等待來自來源代理程式的確認通知訊息。
CompleteReceivedTransfer	來源代理程式已收到來自目的地代理程式的完成訊息，並已將訊息傳送回目的地代理程式以確認完成訊息。
CancelledNewTransfer	來源代理程式已收到新建傳送的取消訊息。
CancelledInProgressTransfer	來源代理程式已收到正在進行的傳送的取消訊息。
ResumingTransfer	來源代理程式已收到重新同步化回應訊息，現在可將傳送排定為重新啟動。
RestartingTransfer	來源或目的地代理程式已收到重新同步化要求訊息，正在等待個別目的地或來源代理程式重新啟動。
WaitingForDestinationCapacity	來源代理程式已從目的地代理程式收到 DESTINATION_CAPACITY_EXCEEDED 錯誤。傳送現在處於一段期間後重試的等待中狀態。
FailedTransferEnding	傳送失敗，但尚未發佈完成日誌訊息，且尚未從狀態儲存庫中移除傳送。例如，在已從目的地代理程式收到失敗回應之後，但在已完成後續處理之前停止代理程式程序，則可能會出現此狀態。

MFT 監視器清單訊息格式

以「保留的發佈資訊」形式發佈至主題字串 SYSTEM.FTE/monitors/agent_name/monitor_name 的 XML 訊息，符合 MonitorList.xsd 綱目。每一個 XML 訊息都會列出屬於該代理程式的作用中監視器。

fteListMonitors 指令與「IBM MQ Explorer」外掛程式可使用這項資訊，對使用者顯示監視器的清單。MonitorList.xsd 綱目文件位於 MQ_INSTALLATION_PATH/mqft/samples/schema 目錄中。MonitorList.xsd 綱目會匯入 Monitor.xsd，它位於相同的目錄。

綱目

下列綱目說明在監視器清單 XML 訊息中有效的元素。

```
<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>
```

瞭解監視器清單訊息

下列清單中說明在監視器清單訊息中使用的元素及屬性：

<monitorList>

群組元素，包含的元素說明為代理程式定義的監視器。

屬性	說明
代理程式 (agent)	必要項目。定義資源監視器的代理程式名稱。
monitor	必要項目。監視器的名稱。對此代理程式而言是唯一的。
version	必要項目。監視器清單訊息格式的版本。

<status>

監視器的狀態。

屬性	說明
state	監視器的狀態。

<configuration>

群組元素，包含的元素說明監視器的配置。

<description>

監視器的說明。（目前未使用。）

<resources>

所監視的一個或多個資源。

<directory>

要監視的目錄。

屬性	說明
recursionLevel	要監視的目錄層次數（從最上層往下算）。
ID	資源的 ID。

<queue>

要監視的佇列。

屬性	說明
ID	資源的 ID。

<triggerMatch>

包含 <conditions> 元素的元素。

<conditions>

包含資源監視器所要監視的一個或多個條件的元素。此元素只能包含下列其中一個元素：

<allOf>、<anyOf> 或 <condition>。

<allOf>

包含資源監視器所要監視的一個或多個條件的元素。此元素可以包含一或多個 <condition> 元素。必須符合此元素中所有的條件，才會觸發資源監視器。

<anyOf>

包含資源監視器所要監視的一個或多個條件的元素。此元素可以包含一或多個 <condition> 元素。必須符合此元素中的單一條件，才會觸發資源監視器。

<condition>

包含資源監視器所要監視的單一條件的元素。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.

如果所要監視的資源是目錄，則必須在條件中指定下列三個元素之一：

- fileMatch
- fileNoMatch
- fileSize

如果所要監視的資源是佇列，則必須在條件中指定下列兩個元素之一：

- queueNotEmpty
- completeGroups

<fileMatch>

檔案名稱符合條件的群組元素。

<pattern>

指定檔案名稱符合型樣。資源上的檔案必須符合此型樣，才能滿足條件。預設型樣是 *（任何檔案皆符合）。

<fileNoMatch>

反向檔案名稱符合條件的群組元素。

<pattern>

指定反向檔案名稱符合型樣。如果受監視資源上沒有符合的檔案，則滿足條件。預設型樣是 *（無任何檔案即符合）。

<fileSize>

檔案大小比較的群組元素。

<compare>

指定檔案大小比較。這個值必須是非負整數。

屬性	說明
operator	要使用的比較運算子。僅支援 > = '。
單元:	指定檔案大小的單位，可以是下列其中一項： <ul style="list-style-type: none"> • B - 位元組 • KB - 千位元組 • MB - 百萬位元組 • GB - 十億位元組 單位值不區分大小寫，因此 mb 的作用與 MB 相同。

<pattern>

要比對的檔案名稱型樣。預設值是 *（任何檔案皆符合）。

<queueNotEmpty>

只有在資源是佇列時，才可指定此項目。指定佇列上必須有訊息，才會觸發監視器。

<completeGroups>

只有在資源是佇列時，才可指定此項目。指定佇列上必須有完整訊息群組存在，才會觸發監視器。對於佇列上的每一個完整群組，分別執行單一傳送作業。

<name>

條件的名稱。

<resource>

識別據以比較條件的資源定義。

屬性	說明
ID	資源的唯一 ID。

<tasks>

群組元素，包含的元素指定滿足監視器觸發條件時所要呼叫的作業。

<task>

群組元素，定義監視器在滿足觸發條件時所要呼叫的個別作業。目前只能指定一項作業。

<name>

作業的名稱。接受任何英數字元。

<description>

作業的說明。接受任何文字值。

<taskXML>

此 XML 訊息說明監視器將執行的作業。此元素的內容使用跳出 XML 格式。

<pollInterval>

每一次根據觸發條件檢查資源之間的時間間隔。

屬性	說明
單元:	指定輪詢問隔的時間單位。有效值為： <ul style="list-style-type: none"> • 秒 • 分鐘 • 時

屬性	說明
	<ul style="list-style-type: none"> • 日 • 週 • 月 • 年

<batch>

單一批次中可包括的觸發相符數上限。

屬性	說明
maxSize	單一批次中可包括的觸發相符數上限

下列 XML 說明在 AGENT_JUPITER 上建立稱為 MONITORTWO 的監視器時，發佈至主題字串 SYSTEM.FTE/monitors/agent_name/MONITORTWO 的「保留的發佈資訊」。<taskXML> 元素內的跳出 XML 說明監視條件符合時所提交的作業。

```
<?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 排程清單訊息格式

發佈至主題字串 SYSTEM.FTE/Scheduler/agent_name 的保留發佈資訊的 XML 訊息，符合 ScheduleList.xsd 綱目。此 XML 訊息列出屬於該代理程式的所有作用中排程。

fteListScheduledTransfers 指令與「IBM MQ Explorer」可使用這項資訊，對使用者顯示排程清單。ScheduleList.xsd 綱目文件位於 MQ_INSTALLATION_PATH/mqft/samples/schema 目錄中。ScheduleList.xsd 綱目會匯入 FileTransfer.xsd，它位於相同的目錄。

綱目

下列綱目說明在監視器清單 XML 訊息中有效的元素。

```
<?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>
```

瞭解排程清單訊息

下列清單中說明在排程清單訊息中使用的元素及屬性：

<schedules>

群組元素，包含單一代理程式上定義的所有排程的相關資訊。

屬性	說明
代理程式 (agent)	必要項目。定義排程的來源代理程式名稱。
大小	必要項目。此代理程式上定義的排程數目。
version	必要項目。排程清單訊息格式的版本。

<managedTransfer>

包含單一排程相關資訊的群組元素。

屬性	說明
ID	必要項目。排程要求訊息的十六進位字串 ID。

<originator>

排程要求的發送端。

<hostName>

提交排程要求的機器的主機名稱。

<userID>

提交排程要求的使用者的使用者 ID。

<mqmdUserID>

提交排程要求的使用者的 MQMD 使用者 ID。

<schedule>

一個元素，包含說明排定的傳送何時執行的元素。

<submit>

指定排定傳送應該開始的日期和時間。

屬性	說明
timebase	指定要使用的時區。此屬性值可以是下列其中一個值： <ul style="list-style-type: none"> • source - 使用來源代理程式的時區 • admin - 使用發出此指令的管理者的時區 • UTC - 使用世界標準時間 (UTC)
時區	根據 timebase 值的時區說明

<repeat>

群組元素，包含排定傳送多久重複一次、排定傳送重複的次數及排定傳送何時停止重複的相關詳細資料。

屬性	說明
區間	間隔單位，必須是下列其中一個值： <ul style="list-style-type: none"> • 分鐘 • 時 • 日 • 週 • 月 • 年

<frequency>

在重複傳送之前必須經歷的時段。

屬性	說明
區間	間隔單位，必須是下列其中一個值： <ul style="list-style-type: none"> • 分鐘 • 時 • 日 • 週 • 月 • 年

<expireTime>

選用元素，指定排定傳送停止重複的日期和時間。此元素與 <expireCount> 元素互斥。

<expireCount>

選用元素，指定排定檔案傳送在停止之前發生的次數。此元素與 <expireTime> 元素互斥。

<next>

指定下一個排定傳送應該開始的日期和時間。

<sourceAgent>

指定來源檔案所在系統上的代理程式名稱。

屬性	說明
代理程式 (agent)	指定代理程式的名稱。

屬性	說明
QMGr	代理程式佇列管理程式的名稱。

<destinationAgent>

指定系統上要做為檔案傳送目的地的代理程式名稱。

屬性	說明
代理程式 (agent)	指定代理程式的名稱。
QMGr	代理程式佇列管理程式的名稱。

<trigger>

選用元素，指定要進行檔案傳送時必須具備的條件。

屬性	說明
log	指出是否記載觸發失敗的旗標。有效值是： <ul style="list-style-type: none"> • yes - 會針對失敗的觸發傳送建立日誌項目 • no - 不會針對失敗的觸發傳送建立日誌項目

<reply>

指定對同步檔案傳送產生的暫時回覆佇列名稱（在指令行上使用 **-w** 參數指定）。佇列名稱是由 `command.properties` 配置檔中的索引鍵 **dynamicQueuePrefix** 或預設值 `WMQFTE.*` 所定義。（如果未指定的話）。

屬性	說明
QMGR	產生暫時動態佇列以接收回覆的指令佇列管理程式的名稱。

<transferSet>

指定您想要讓排定傳送一起執行的檔案傳送群組。在傳輸期間，<transferSet> 是包含 <item> 元素的群組元素。

屬性	說明
priority	傳送的優先順序層次。優先順序是範圍 0-9 中的值，其中 0 是最低優先順序。預設優先順序層次是 0，依預設，傳送時會使用來源代理程式的優先順序層次。

<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 637 頁的『檔案傳送日誌訊息格式』](#)。

範例

```
<?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 範本 XML 訊息範例

建立範本時，會將訊息發佈至 SYSTEM.FTE 主題，其主題字串為 `Templates/template_ID`。這個 XML 範例說明定義在 Managed File Transfer 網路中的單一範本。

```

<?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>

```

相關工作

第 163 頁的『[使用 IBM MQ Explorer 建立檔案傳送範本](#)』

您可以從「IBM MQ Explorer」或從指令行建立檔案傳送範本。然後，您可以使用該範本，利用範本詳細資料建立新的檔案傳送，或提交範本以啟動檔案傳送。

相關參考

第 354 頁的『[fteCreateTemplate: 建立新的檔案傳送範本](#)』

fteCreateTemplate 指令可建立檔案傳送範本，以備將來使用。唯一必要的參數是 **-tn** (*template_name*) 參數。其他所有參數是選用的，雖然如果指定來源檔案規格，您還必須提供目的地檔案。同樣地，如果指定目的地檔案，您還必須提供來源檔案規格。

檔案傳送狀態訊息格式

訊息發佈至協調佇列管理程式，以指出傳送集中每一個檔案的傳送狀態。代理程式每次處理檔案傳送要求時，符合 `TransferStatus.xsd` XML 綱目的交易訊息會發佈至協調佇列管理程式（在 `SYSTEM.FTE/Transfers/agent_name/transfer ID` 主題）。`TransferStatus.xsd` 檔案位於 `WMQMFT` 安裝的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目錄中。

綱目

下列綱目說明在傳送狀態 XML 訊息中有效的元素。

```
<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="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>
```

瞭解傳送狀態訊息

下列清單中說明在傳送狀態訊息中使用的元素及屬性：

<transaction>

群組元素，包含檔案傳送的所有元素。

屬性	說明
version	指定此元素的版本，由 Managed File Transfer 提供。
ID	檔案傳送的唯一 ID。

<sourceAgent>

指定來源檔案所在系統上的代理程式名稱。

屬性	說明
代理程式 (agent)	代理程式的名稱。
QMgr	代理程式佇列管理程式的名稱。

<destinationAgent>

指定系統上要做為檔案傳送目的地的代理程式名稱。

屬性	說明
代理程式 (agent)	代理程式的名稱。
QMgr	代理程式佇列管理程式的名稱。

<transferSet>

指定一起執行的一組檔案傳送。傳送的所有檔案必須從相同來源代理程式發出，並於相同目的地代理程式上結束。

屬性	說明
time	指定日期和時間（以日期時間格式表示）。

<stats>

必要項目。定義傳送的度量，包括在給定的秒數內到目前為止已複製的位元組數。同時也提供 <transferSet> 中現行項目數目佔項目總數的比例。

屬性	說明
位元組	到目前為止複製的位元組數。
秒	傳送這些位元組所花費的秒數。
currentItem	所傳送現行項目的索引。
totalItems	所傳送項目總數。

<current>

選用元素。群組元素，包含指定目前進行中的檔案傳送的元素。 <current> 元素指出現行項目截至目前為止已傳送的資料位元組數，以及預期的位元組總數

<source>

群組元素，包含指定來源檔案名稱的元素。

<file>

指定要傳送之檔案的來源路徑。此路徑指定專供傳送之用。此路徑可能會與傳送日誌輸出中的路徑不同，後者會採用絕對路徑格式。

<destination>

群組元素，包含指定目的地檔案名稱或規格的元素。

<file>

指定要傳送之檔案的目的地路徑。此路徑指定專供傳送之用。此路徑可能會與傳送日誌輸出中的路徑不同，後者會採用絕對路徑格式。

屬性	說明
別名 (alias)	指定目的地檔案的別名。此別名是來源檔案的名稱，不包括針對傳送所指定的任何目錄路徑。
fileSpace	指定供寫入目的地檔案的檔案空間的名稱。

<queue>

與 <destination> 元素一起使用時，指定您想要傳送至其中的佇列名稱。此名稱的格式為 QUEUE 或 QUEUE@QUEUE_MANAGER。

檔案傳送進度訊息範例

進行傳送時，會將訊息發佈至 SYSTEM.FTE 主題，其主題字串為 Transfers/agent_name/transfer_ID。XML 範例會顯示單一檔案傳送及多重檔案傳送的進度訊息。

單一檔案傳送

下列範例顯示進行中的單一檔案傳送的詳細資料。

```
<?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>
```

多重檔案傳送

如果傳送集中包含更多檔案，傳送狀態訊息會指出所處理的檔案及到目前為止傳送的位元組數。

```
<?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>
```

檔案傳送日誌訊息格式

檔案傳送日誌訊息發佈至具有 Log/agent_name/transfer_ID 的主題字串的 SYSTEM.FTE 主題。這些訊息符合綱目 TransferLog.xsd，此綱目位於 Managed File Transfer 安裝的 MQ_INSTALLATION_PATH/mqft/samples/schema 目錄中。

如果您要監視檔案傳送或收集其相關資料，請將訂閱設定為符合您感興趣的傳送的萬用字元主題。例如：

```
Log/#
```

或

```
Log/FTEAGENT/#
```

這可以是可延續或不可延續的訂閱。關閉訂閱應用程式與佇列管理程式的連線時，可延續訂閱會繼續存在。不可延續訂閱只有在訂閱應用程式與佇列管理程式的連線開啟時才會存在。

綱目

下列綱目說明傳送日誌 XML 訊息中的有效元素。

```
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
  <xsd:include schemaLocation="fteutils.xsd"/>
  <xsd:element name="transaction">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="action" type="actionType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="sourceAgent" type="agentExitStatusType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="sourceWebGateway" type="webGatewayType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="sourceWebUser" type="webUserType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="destinationAgent" type="agentExitStatusType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="destinationWebGateway" type="webGatewayType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="destinationWebUser" type="webUserType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="agent" type="agentExitStatusType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="originator" type="origRequestType"
          maxOccurs="1" minOccurs="1"/>
        <xsd:element name="status" type="statusType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="trigger" type="triggerType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="transferSet" type="transferSetType"
          maxOccurs="1" minOccurs="1"/>
        <xsd:element name="job" type="jobType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="scheduleLog" type="scheduleLogType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="statistics" type="statisticsType"
          maxOccurs="1" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="version" type="versionType" use="required"/>
      <xsd:attribute name="ID" type="IDType" use="required"/>
      <xsd:attribute name="relatedID" type="IDType" use="optional"/>
      <xsd:attribute name="agentRole" type="agentRoleType" use="optional"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="agentExitStatusType">
    <xsd:complexContent>
      <xsd:extension base="agentType">
        <xsd:sequence>
          <xsd:element name="startExits" type="exitGroupType"
            minOccurs="0" maxOccurs="1"/>
          <xsd:element name="endExits" type="exitGroupType"
            minOccurs="0" maxOccurs="1"/>
          <xsd:element name="systemInfo" type="systemInfoType"
            minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
```

```

<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>

```

註: 從 IBM MQ 9.0 開始, Managed File Transfer 不支援「Web 閘道」或 Web 代理程式。

瞭解傳送日誌訊息

<transaction>

可指定您要一起執行的一組傳送的群組元素。

屬性	說明
version	指定此元素的版本, 由 Managed File Transfer 詳述。
ID	指定唯一交易 ID。ID 最多可為 48 個英數字元。

屬性	說明
relatedID	選用項目。如果交易是從檔案空間刪除或下載檔案， relatedID 可指定已將檔案上傳至檔案空間的傳送的交易 ID。
agentRole	選用項目。指定相關的代理程式是否在來源或目的地系統上。
xmlns:xsi	名稱空間宣告。指出此綱目中使用的元素及資料類型衍生自 "https://www.w3.org/2001/XMLSchema-instance" 名稱空間。
xsi:noNamespaceSchemaLocation	指定 XML 綱目文件的名稱及位置，以驗證此訊息中沒有名稱空間宣告。您為此屬性指定的值必須參照 Managed File Transfer TransferLog.xsd 文件。

<action>

說明時間屬性記載的時間的檔案傳送狀態。狀態可以是下列其中一值：

- 已啟動
- 進度
- 已完成
- 已取消
- malformed (指示無法解譯檔案傳送要求訊息內容)
- notAuthorized
- deleted

屬性	說明
time	擷取傳送狀態的時間，以 UTC 格式表示。

<sourceAgent>

指定來源檔案所在系統上的代理程式名稱。 Only <sourceAgent> or <sourceWebUser> can be specified.

<startExits>

包含一個以上使用者結束程式元素的群組元素。此元素只能出現一次。

<endExits>

包含一個以上使用者結束程式元素的群組元素。此元素只能出現一次。

<systemInfo>

說明系統架構、名稱及版本。此元素只能出現一次。

屬性	說明
代理程式 (agent)	來源系統上的代理程式名稱。
QMgr	來源系統上的佇列管理程式的名稱。
agentType	代理程式的類型。有效值為： <ul style="list-style-type: none"> • STANDARD - 一般代理程式 • BRIDGE - 通訊協定橋接器代理程式 • CD_BRIDGE - Connect:Direct 橋接器代理程式 • EMBEDDED - 內嵌式代理程式 • SFG - Sterling File Gateway 內嵌式代理程式
bridgeURL	選用項目。如果代理程式是通訊協定橋接器代理程式，則為管理通訊協定伺服器的主機名稱。

屬性	說明
pnode	選用項目。如果代理程式是 Connect:Direct 橋接器代理程式，則為傳送中涉及的 Connect:Direct 主要節點的名稱。
snode	選用項目。如果代理程式是 Connect:Direct 橋接器代理程式，則為傳送中涉及的 Connect:Direct 次要節點的名稱。
bridgeNode	選用項目。如果代理程式是 Connect:Direct 橋接器代理程式，則為屬於 Connect:Direct 橋接器一部分的 Connect:Direct 節點的名稱。此節點與主要節點或次要節點相同。

<destinationAgent>

指定檔案傳送至其中的系統上的代理程式名稱。 Either <destinationAgent> or <destinationWebUser> can be specified.

屬性	說明
代理程式 (agent)	目的地系統上的代理程式名稱。
QMgr	目的地系統上的佇列管理程式名稱。
agentType	代理程式的類型。有效值為： <ul style="list-style-type: none"> • STANDARD - 一般代理程式 • BRIDGE - 通訊協定橋接器代理程式 • CD_BRIDGE - Connect:Direct 橋接器代理程式 • EMBEDDED - 內嵌式代理程式 • SFG - Sterling File Gateway 內嵌式代理程式
bridgeURL	選用項目。如果代理程式是通訊協定橋接器代理程式，則為管理通訊協定伺服器的主機名稱。
pnode	選用項目。如果代理程式是 Connect:Direct 橋接器代理程式，則為傳送中涉及的 Connect:Direct 主要節點的名稱。
snode	選用項目。如果代理程式是 Connect:Direct 橋接器代理程式，則為傳送中涉及的 Connect:Direct 次要節點的名稱。
bridgeNode	選用項目。如果代理程式是 Connect:Direct 橋接器代理程式，則為屬於 Connect:Direct 橋接器一部分的 Connect:Direct 節點的名稱。此節點與主要節點或次要節點相同。

<startExits>

包含一個以上使用者結束程式元素的群組元素。此元素只能出現一次。

<endExits>

包含一個以上使用者結束程式元素的群組元素。此元素只能出現一次。

<systemInfo>

說明系統架構、名稱及版本。此元素只能出現一次。

<originator>

群組元素，包含指定要求發送端的元素。

<hostName>

來源檔案所在系統的主機名稱。

<userID>

發出檔案傳送的使用者 ID。

<mqmdUserID>

在訊息描述子 (MQMD) 中提供的 IBM MQ 使用者 ID。

<webUserID>

選用項目。提供給提交傳送要求的 Web 瀏覽器的使用者 ID。

<webBrowser>

選用項目。從中提交傳送要求的 Web 瀏覽器。

<status>

結果碼及補充訊息。

<trigger>

包含原始傳送要求中定義的觸發程式元素的群組元素。這些元素可以是下列其中一個或全部：

<fileExist>

根據檔案是否存在觸發條件

<fileSize>

根據檔案符合或超過指定的大小觸發條件

<transferSet>

指定您想要一起執行的檔案傳送群組。在傳輸期間，<transferSet> 是包含 <item> 元素的群組元素。

屬性	說明
startTime	記錄傳送集啟動的時間，以 UTC 格式表示。
total	指定此傳送集中的項目總數。
索引 (index)	選用屬性。指定傳送集中進行的第一個項目的位置。索引屬性從零開始增量。比方說，例如索引設定為 1，則進度訊息為兩個項目的第二個。
大小	選用屬性。指定進度報告中的項目數。
priority	選用屬性。傳送的優先順序層次。優先順序是範圍 0-9 中的值，其中 0 是最低優先順序。預設優先順序層次為 0，依預設，傳送使用來源代理程式優先順序層次。

<metaDataSet>

包含一個以上下列屬性的群組元素：

<metaData>

屬性	說明
key	meta 資料鍵值組一半的鍵。<metaData> 元素內容包含配對的一半值。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 668 頁的『檔案傳送要求訊息格式』](#)。

<name>

name 的值可以是任何字串。

<scheduleLog>

群組元素，包含指定來源及目的地檔案名稱和位置的元素。

屬性	說明
ID	與排程 ID 相符（如果傳送是排定的傳送）。

<item>

群組元素，包含指定來源及目的地檔案名稱和位置的元素。

<source>

包含來源系統上檔案的 <file> 元素或 <queue> 元素及 <checksum> 元素的群組元素。

屬性	說明
recursive	指定當 <source> 元素是目錄或包含萬用字元時，在子目錄中遞迴地傳送檔案。
disposition	指定當 <source> 已順利傳送至其目的地時，對 <source> 元素採取的動作。有效選項如下所示： <ul style="list-style-type: none"> • leave - 來源檔案維持不變。 • delete - 順利傳送來源檔案之後，即從來源系統中刪除來源檔案。
correlationBoolean	布林相關性值。如果來源是 Connect:Direct 橋接器，這會指定 Connect:Direct 程序是否由使用者定義。
correlationString1	字串相關性值。如果來源是 Connect:Direct 橋接器，這會指定在傳送的目的地發生的 Connect:Direct 程序的名稱。
correlationNum1	數字相關性值。如果來源是 Connect:Direct 橋接器，這會指定在傳送的目的地發生的 Connect:Direct 程序的 ID 號。

<queue>



在與 <source> 元素搭配使用時，指定從中讀取已傳送訊息的佇列名稱，此佇列位於來源代理程式的佇列管理程式上。

屬性	說明
messageCount	從佇列中讀取的訊息數。
groupId	從佇列中讀取的訊息的 IBM MQ 群組 ID。

<destination>

包含目的地之 <file> 元素或 <queue> 元素及 <checksum> 元素的群組元素。

只有 <file> 和 <queue> 其中之一呈現為目的地的子元素。

屬性	說明
類型	目的地的類型。有效選項如下所示： <ul style="list-style-type: none"> • queue - 指定 IBM MQ 佇列作為目的地 • file - 指定檔案作為目的地 • directory - 指定目錄作為目的地 •  dataset - 指定 z/OS 資料集作為目的地 •  pds - 指定 z/OS 分割的資料集作為目的地 <p>只有在 <destination> 元素具有 <queue> 的子元素時，選項佇列才會存在。只有在 <destination> 元素具有 <file> 的子元素時，其他選項才會呈現。</p>
exist	指定如果目的地系統上存在目的地檔案時要採取的動作。有效選項如下所示： <ul style="list-style-type: none"> • error - 報告錯誤，且不傳送檔案。 • overwrite - 改寫現有目的地檔案。 <p>如果 <destination> 元素具有 <queue> 的子元素，則無法呈現此屬性。</p>
correlationBoolean	布林相關性值。如果目的地是 Connect:Direct 橋接器，這會指定 Connect:Direct 程序是否由使用者定義。

屬性	說明
correlationString1	字串相關性值。如果目的地是 Connect:Direct 橋接器，這會指定在傳送的目的地發生的 Connect:Direct 程序的名稱。
correlationNum1	數字相關性值。如果目的地是 Connect:Direct 橋接器，這會指定在傳送的目的地發生的 Connect:Direct 程序的 ID 號。

<file>

指定已傳送的檔案的絕對路徑（同時在來源及目的地上）。完整路徑採用與您的作業系統一致的格式，例如，C:/from/here.txt。不使用檔案 URI。

<queue>

在與 <destination> 元素搭配使用時，指定已傳送至其中的佇列名稱，此佇列位於連接至目的地代理程式佇列管理程式的任何佇列管理程式上。

屬性	說明
messageCount	寫入佇列的訊息數。
messageLength	寫入佇列的訊息長度。
groupId	如果傳送要求已指定檔案分為多則訊息，則此屬性的值是寫入佇列的訊息的 IBM MQ 群組 ID。
messageId	如果傳送要求未指定檔案分為多則訊息，則此屬性的值是寫入佇列的訊息的 IBM MQ 訊息 ID。

<checksum>

選用元素。

指定產生訊息摘要以建立數位簽章的雜湊演算法類型。目前，Managed File Transfer 僅支援訊息摘要演算法 5 (MD5)。總和檢查提供了讓您確認已傳送檔案的完整性是否完整的方法。

<malformed>

形態異常的訊息的群組元素。

屬性	說明
version	
ID	
agentRole	來源代理程式或目的地代理程式

<statistics>

傳送的統計資訊的群組元素（可用時）。

<actualStartTime>

代理程式開始執行傳送的實際時間。通常，此時間與記錄的傳送的啟動時間相同（或非常接近）。但是，在代理程式忙碌時，提交的佇列可能會排在佇列中，直至代理程式可以執行傳送為止。

<retryCount>

傳送進入回復狀態及由代理程式重試的次數。傳送可能會進入回復狀態，因為來源及目的地代理程式中斷通訊（由於 IBM MQ 網路錯誤，或由於這些代理程式在某段時間未接收到資料或確認通知訊息）。此期間由下列代理程式內容確定：transferAckTimeout 及 transferAckTimeoutRetries。

<numFileFailures>

transferSet 中無法順利傳送的檔案數。

<numFileWarnings>

在傳送時產生警告，但已順利傳送的 transferSet 中的檔案數。

範例

為下列每一種傳送類型，提供了符合此綱目的 XML 訊息的範例：

- [單一檔案的傳送](#)
- [包含多個檔案的傳送](#)
- [失敗的檔案傳送](#)
- [使用觸發程式定義的傳送](#)
- [按排程啟動的傳送](#)
- [呼叫使用者結束程式的傳送](#)
- [透過 Connect:Direct 橋接器節點的傳送](#)

單一傳送日誌訊息範例

傳送進行時，會將訊息發佈至 SYSTEM.FTE 主題，其主題字串為 `Log/agent_name/transfer_ID`。XML 範例顯示啟動中、進行中及已完成的單一檔案傳送的日誌訊息。

單一檔案傳送 - 已啟動

```
<?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>
```

單一檔案傳送成功 - 進行中

```
<?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>

```

單一檔案傳送成功 - 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>

```

多重檔案傳送日誌訊息範例

這些範例說明在包含多個檔案的傳送執行時，會將訊息發佈至 SYSTEM.FTE 主題，其主題字串為 Log/agent_name/transfer_ID。

多重檔案傳送 - 啟動

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
version="4.00"
ID="414d51205553322e42494e44494e47538b0f404d035c0020"
agentRole="sourceAgent"

```

```

        xsi:noNamespaceSchemaLocation="TransferLog.xsd"
        xmlns=""
<action time="2011-01-26T13:12:58.534Z">started</action>
<sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
  <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
</sourceAgent>
<destinationAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER"/>
<originator>
  <hostName>example.com</hostName>
  <userID>mqm</userID>
  <mqmdUserID>mqm</mqmdUserID>
</originator>
<transferSet startTime="2011-01-26T13:12:58.534Z" total="6" bytesSent="0">
  <metaDataSet>
    <metaData key="com.ibm.wmqfte.SourceAgent">AGENT_JUPITER</metaData>
    <metaData key="com.ibm.wmqfte.DestinationAgent">AGENT_JUPITER</metaData>
    <metaData key="com.ibm.wmqfte.MqmdUser">mqm</metaData>
    <metaData key="com.ibm.wmqfte.OriginatingUser">mqm</metaData>
    <metaData key="com.ibm.wmqfte.OriginatingHost">example.com</metaData>
    <metaData key="com.ibm.wmqfte.TransferId">414d51205553322e42494e44494e47538b0f404d035c0020</
metaData>
    <metaData key="com.ibm.wmqfte.Priority">0</metaData>
  </metaDataSet>
</transferSet>
</transaction>

```

多重檔案傳送 - 進行中

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  ID="414d51205553322e42494e44494e47538b0f404d035c0020"
  agentRole="sourceAgent"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns=""
<action time="2011-01-26T13:12:58.753Z">progress</action>
<sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
  <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
</sourceAgent>
<destinationAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER">
  <systemInfo architecture="x86" name="Linux" version="2.6.31-21-generic"/>
</destinationAgent>
<originator>
  <hostName>example.com.</hostName>
  <userID>mqm</userID>
  <mqmdUserID>mqm</mqmdUserID>
</originator>
<transferSet index="0" size="6" startTime="2011-01-26T13:12:58.534Z" total="6" bytesSent="440">
  <item mode="binary">
    <source disposition="leave" type="file">
      <file size="0" last-modified="2011-01-26T13:10:19.000Z"/>/srv/nfs/incoming/file01.txt</
file>
      <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
    </source>
    <destination type="file">
      <file size="0" last-modified="2011-01-26T13:12:58.000Z"/>/srv/nfs/outgoing/file01.txt</
file>
      <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
    </destination>
    <status resultCode="0"/>
  </item>
  <item mode="binary">
    <source disposition="leave" type="file">
      <file size="0" last-modified="2011-01-26T13:10:19.000Z"/>/srv/nfs/incoming/file02.txt</
file>
      <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
    </source>
    <destination type="file">
      <file size="0" last-modified="2011-01-26T13:12:58.000Z"/>/srv/nfs/outgoing/file02.txt</
file>
      <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
    </destination>
    <status resultCode="0"/>
  </item>
  <item mode="binary">
    <source disposition="leave" type="file">
      <file size="0" last-modified="2011-01-26T13:10:19.000Z"/>/srv/nfs/incoming/file03.txt</
file>
      <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>

```



```

        </source>
        <destination type="file">
            <file size="0" last-modified="2011-01-26T13:12:58.000Z"/>/srv/nfs/outgoing/file03.txt</
file>
            <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
        </destination>
        <status resultCode="0"/>
    </item>
    <item mode="binary">
        <source disposition="leave" type="file">
            <file size="0" last-modified="2011-01-26T13:10:19.000Z"/>/srv/nfs/incoming/file04.txt</
file>
            <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
        </source>
        <destination type="file">
            <file size="0" last-modified="2011-01-26T13:12:58.000Z"/>/srv/nfs/outgoing/file04.txt</
file>
            <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
        </destination>
        <status resultCode="0"/>
    </item>
    <item mode="binary">
        <source disposition="leave" type="file">
            <file size="0" last-modified="2011-01-26T13:10:19.000Z"/>/srv/nfs/incoming/file05.txt</
file>
            <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
        </source>
        <destination type="file">
            <file size="0" last-modified="2011-01-26T13:12:58.000Z"/>/srv/nfs/outgoing/file05.txt</
file>
            <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
        </destination>
        <status resultCode="0"/>
    </item>
    <item mode="binary">
        <source disposition="leave" type="file">
            <file size="0" last-modified="2011-01-26T13:10:19.000Z"/>/srv/nfs/incoming/file06.txt</
file>
            <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
        </source>
        <destination type="file">
            <file size="0" last-modified="2011-01-26T13:12:58.000Z"/>/srv/nfs/outgoing/file06.txt</
file>
            <checksum method="MD5">d41d8cd98f00b204e9800998ecf8427e</checksum>
        </destination>
        <status resultCode="0"/>
    </item>
</transferSet>
</transaction>

```

多重檔案傳送 - 已完成

```

<?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="2011-01-26T13:12:58.766Z">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>

```

失敗的檔案傳送日誌訊息範例

傳送進行時，會將訊息發佈至 SYSTEM.FTE 主題，其主題字串為 `Log/agent_name/transfer_ID`。下列 XML 範例顯示檔案傳送失敗、正在啟動、進行中及已完成時的日誌訊息。

檔案傳送失敗 - 已啟動

```

<?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>

```

檔案傳送失敗 - 進行中

```

<?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>

```

檔案傳送失敗 - 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>

```

觸發的檔案傳送日誌訊息範例

進行傳送時，會將訊息發佈至 SYSTEM.FTE 主題，其主題字串為 `Log/agent_name/transfer_ID`。XML 範例顯示的日誌訊息是在啟動包含觸發條件的檔案傳送時建立的。

觸發單一檔案傳送成功 - 已啟動

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
    ID="414d5120514d31202020202020202020207e970d492000a102" 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>

```

排定的檔案傳送日誌訊息範例

進行傳送時，會將訊息發佈至 SYSTEM.FTE 主題，其主題字串為 `Log/agent_name/transfer_ID`。XML 範例顯示依據排程而進行檔案傳送時所建立的日誌訊息。

排程傳送交易訊息

當排程項目到期而開始進行檔案傳送時，檔案傳送會遵循慣常順序對 SYSTEM.FTE/Log/agent_name 主題發佈交易訊息：

- 動作已啟動 (TransferLog.xsd)
- 動作進行中 (TransferLog.xsd)
- 動作已完成 (TransferLog.xsd)

只有動作為已啟動的日誌交易訊息，才會在 `<scheduleLog>` 元素的 ID 屬性中包含排定傳送的 ID。在整個傳送的完整生命週期內，這可讓排程 ID 連結至傳送 ID。

已啟動：

```

<?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>

```

進行中：

```

<?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 使用者結束程式日誌訊息範例

進行傳送時，會將訊息發佈至 SYSTEM.FTE 主題，其主題字串為 `Log/agent_name/transfer_ID`。下列 XML 範例顯示進行檔案傳送（包含對使用者結束程式的呼叫）時所建立的日誌訊息。

結束程式單一檔案傳送繼續 - 已啟動

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
  ID="414d5120514d31202020202020202020207e970d492000d502" agentRole="sourceAgent"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2008-11-02T22:36:13.046Z">started</action>
  <sourceAgent agent="FTEAGENT" QMgr="QM1">
    <systemInfo architecture="x86" name="Windows 7"
      version="6.1 build 7601 Service Pack 1"/>
  </sourceAgent>
  <destinationAgent agent="FTEAGENT" QMgr="QM1"/>
  <originator>
    <hostName>reportserver.com</hostName>
    <userID>USER1</userID>
    <mqmdUserID>USER1 </mqmdUserID>
  </originator>
  <transferSet startTime="2008-11-02T22:36:13.046Z" total="1">
    <metaDataSet>
      <metaData key="testkey1">testvalue1</metaData>
      <metaData key="testkey2">testvalue2</metaData>
    </metaDataSet>
  </transferSet>
</transaction>
```

結束程式單一檔案傳送繼續 - 已完成

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
  ID="414d5120514d31202020202020202020207e970d492000d502"
  agentRole="sourceAgent"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2008-11-02T22:36:13.546Z">completed</action>
  <sourceAgent agent="FTEAGENT" QMgr="QM1">
    <startExits>
      <exit name="class testExits.SourceExit1">
        <status resultCode="proceed">
          <supplement>Source Start, modified metadata</supplement>
        </status>
      </exit>
    </startExits>
    <endExits>
      <exit name="class testExits.SourceExit1">
        <status>
          <supplement>Source End</supplement>
        </status>
      </exit>
    </endExits>
    <systemInfo architecture="x86" name="Windows 7"
      version="6.1 build 7601 Service Pack 1"/>
  </sourceAgent>
  <destinationAgent agent="FTEAGENT" QMgr="QM1">
    <startExits>
      <exit name="class testExits.DestinationExitProceed">
        <status resultCode="proceed">
          <supplement>Destination start, with proceed</supplement>
        </status>
      </exit>
    </startExits>
    <endExits>
      <exit name="class testExits.DestinationExitProceed">
        <status>
          <supplement>destination end</supplement>
        </status>
      </exit>
    </endExits>
  </destinationAgent>
</transaction>
```

```

        </exit>
    </endExits>
    <systemInfo architecture="x86" name="Windows 7"
        version="6.1 build 7601 Service Pack 1"/>
</destinationAgent>
<originator>
    <hostName>reportserver.com</hostName>
    <userID>USER1</userID>
    <mqmdUserID>USER1        </mqmdUserID>
</originator>
<transferSet startTime="2008-11-02T22:36:13.046Z" total="1">
    <metaDataSet>
        <metaData key="newkey2">newvalue2</metaData>
        <metaData key="newkey1">newvalue1</metaData>
        <metaData key="newkey4">newvalue4</metaData>
        <metaData key="newkey3">newvalue3</metaData>
        <metaData key="newkey5">newvalue5</metaData>
        <metaData key="testkey1">testvalue1</metaData>
        <metaData key="testkey2">testvalue2</metaData>
    </metaDataSet>
</transferSet>
</transaction>

<!--
    In this example the source transfer start exit has modified the
    metadata as follows:

    Added keys and values for:
    newkey1, newvalue1
    newkey2, newvalue2
    newkey3, newvalue3
    newkey4, newvalue4
    newkey5, newvalue5

    Replaced values for:
    key1 to modifiedValue1

    Deleted keys and values for:
    key2
-->

```

結束程式單一檔案傳送取消 - 已取消

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
    ID="414d5120514d31202020202020202020207e970d492000c702" 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 橋接器傳送日誌訊息範例

當目的地代理程式或來源代理程式為 Connect:Direct 橋接器代理程式時，destinationAgent 或 sourceAgent 元素包含其他屬性。「已啟動」日誌訊息僅包含有關 Connect:Direct 傳送的部分資訊。「進行中」及「已完成」日誌訊息，包含有關 Connect:Direct 傳送的完整資訊。

來源代理程式為 Connect:Direct 橋接器代理程式

已啟動：

```

<?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>

```

進行中：

```

<?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>

```

目的地代理程式為 Connect:Direct 橋接器代理程式

已啟動:

```

<?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>

```

進行中:

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  ID="414d5120514d5f696b6b796f20202020a704654d2008e102"
  agentRole="sourceAgent"
  version="4.00"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2011-03-07T10:29:46.682Z">progress</action>
  <sourceAgent QMgr="QM_ASTEROID" agent="PALLAS" agentType="STANDARD">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </sourceAgent>
  <destinationAgent QMgr="QM_ASTEROID" agent="VESTA" agentType="CD_BRIDGE"
    bridgeNode="CDNODE_VESTA" pnode="CDNODE_VESTA" snode="CDNODE_HYGIEA">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </destinationAgent>
  <originator>
    <hostName>belt.example.com</hostName>
    <userID>sol</userID>
    <mqmdUserID>sol</mqmdUserID>
  </originator>
  <transferSet bytesSent="48" index="0" size="1" startTime="2011-03-07T10:29:44.854Z" total="1">
    <item mode="binary">
      <source disposition="leave" type="file">
        <file last-modified="2011-03-04T14:53:28.323Z" size="4">D:\AGENTS\PALLAS\test.txt</
file>
        <checksum method="MD5">098f6bcd4621d373cade4e832627b4f6</checksum>
      </source>
      <destination processName="f2006965" processNumber="59" type="file">
        <file size="4">CDNODE_VESTA:D:/AGENTS/CDNODE_VESTA/test.txt</file>
        <checksum method="MD5">098f6bcd4621d373cade4e832627b4f6</checksum>
      </destination>
      <status resultCode="0"/>
    </item>
  </transferSet>
</transaction>

```

已完成:

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  ID="414d5120514d5f696b6b796f20202020a704654d2008e102"
  agentRole="sourceAgent"
  version="4.00"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2011-03-07T10:29:46.698Z">completed</action>
  <sourceAgent QMgr="QM_ASTEROID" agent="PALLAS" agentType="STANDARD">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </sourceAgent>
  <destinationAgent QMgr="QM_ASTEROID" agent="VESTA" agentType="CD_BRIDGE"
    bridgeNode="CDNODE_VESTA" pnode="CDNODE_VESTA" snode="CDNODE_HYGIEA">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </destinationAgent>
  <originator>
    <hostName>belt.example.com</hostName>
    <userID>sol</userID>
    <mqmdUserID>sol</mqmdUserID>
  </originator>
  <status resultCode="0">
    <supplement>BFGRP0032I: The file transfer request has successfully completed.</supplement>
  </status>
  <transferSet bytesSent="48" startTime="2011-03-07T10:29:44.854Z" total="1">

```

```

    <metaDataSet>
      <metaData key="com.ibm.wmqfte.SourceAgent">PALLAS</metaData>
      <metaData key="com.ibm.wmqfte.DestinationAgent">VESTA</metaData>
      <metaData key="com.ibm.wmqfte.MqmdUser">sol</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingUser">sol</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingHost">belt.example.com.</metaData>
      <metaData key="com.ibm.wmqfte.TransferId">414d5120514d5f696b6b796f20202020a704654d2008e102</
metaData>
      <metaData key="com.ibm.wmqfte.Priority">0</metaData>
    </metaDataSet>
  </transferSet>
</statistics>
  <actualStartTime>2011-03-07T10:29:45.010Z</actualStartTime>
  <retryCount>0</retryCount>
  <numFileFailures>0</numFileFailures>
  <numFileWarnings>0</numFileWarnings>
</statistics>
</transaction>

```

排定的檔案傳送日誌訊息格式

代理程式每次處理排定的檔案傳送要求時，排程日誌訊息就會發佈至協調佇列管理程式（在其 SYSTEM.FTE/Log/agent name/schedule ID 主題）。此訊息符合 ScheduleLog.xsd XML 綱目。

綱目

下列綱目說明在排程日誌 XML 訊息中有效的元素。

```

<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
  <xsd:include schemaLocation="fteutils.xsd"/>

  <xsd:element name="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:complexType>

```

```

    </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>

```

瞭解排程日誌訊息

以下說明在排程日誌訊息中使用的元素及屬性：

<schedulelog>

群組元素，說明提交的單一排定檔案傳送。

屬性	說明
version	指定此元素的版本，由 Managed File Transfer 詳述。
ID	提交的排程檔案傳送的唯一 ID。

<originator>

群組元素，包含指定要求發送端的元素。

<hostName>

來源檔案所在系統的主機名稱。

<userID>

發出檔案傳送的使用者 ID。

<mqmdUserID>

在訊息描述子 (MQMD) 中提供的 MQ 使用者 ID

<action>

針對符合 <schedulelog> 元素之 ID 屬性的排定傳送，指定要採取的動作。此元素必須是下列其中一個值：

- submit - 新建排定的傳送
- delete - 取消排程傳送
- expire - 即將處理排程傳送項目
- skipped - 因為代理程式離線，無法啟動排定的傳送。當代理程式變成可用時會記載此訊息，指出已跳過傳送。

屬性	說明
time	指定發佈日誌項目的日期和時間（以日期時間格式表示）。

<sourceAgent>

指定來源檔案所在系統上的代理程式名稱。

屬性	說明
代理程式 (agent)	指定代理程式的名稱。
QMgr	代理程式佇列管理程式的名稱。

<destinationAgent>

指定系統上要做為檔案傳送目的地的代理程式名稱。

屬性	說明
代理程式 (agent)	指定代理程式的名稱。
QMgr	代理程式佇列管理程式的名稱。

<status>

結果碼及補充訊息。

<transferSet>

指定您想要一起執行的檔案傳送群組。在傳輸期間，<transferSet> 是包含 <item> 元素的群組元素。

屬性	說明
大小	指定傳送項目的數目。
priority	傳送的優先順序層次。優先順序是範圍 0-9 中的值，其中 0 是最低優先順序。預設優先順序層次是 0，依預設，傳送時會使用來源代理程式的優先順序層次。

<item>

群組元素，包含指定來源及目的地檔案名稱和位置的元素。

屬性	說明
mode	指定傳送模式為二進位或文字模式。
checksumMethod	指定產生訊息摘要以建立數位簽章的雜湊演算法類型。允許的值是 MD5 或 none



<source>

群組元素，包含來源系統上檔案的 <file> 及 <checksum> 元素。

屬性	說明
recursive	指定當 <source> 元素是目錄或包含萬用字元時，在子目錄中遞迴地傳送檔案。
disposition	指定當 <source> 已順利傳送至其目的地時，對 <source> 元素採取的動作。有效選項如下所示： <ul style="list-style-type: none">• leave - 來源檔案維持不變。• delete - 順利傳送來源檔案之後，即從來源系統中刪除來源檔案。

<destination>

群組元素，包含目的地系統上檔案的 <file> 及 <checksum> 元素。

屬性	說明
類型	目的地的檔案或目錄的類型。有效選項如下所示： <ul style="list-style-type: none">• file - 指定檔案作為目的地• directory - 指定目錄作為目的地•  dataset - 指定 z/OS 資料集作為目的地•  PDS - 指定 z/OS 分割的資料集作為目的地
exist	指定如果目的地系統上存在目的地檔案時要採取的動作。有效選項如下所示： <ul style="list-style-type: none">• error - 報告錯誤，且不傳送檔案。• overwrite - 改寫現有目的地檔案。

<file>

指定要傳送的檔案名稱。請使用完整路徑，其格式與作業系統一致，例如 C:/from/here.txt。請不要使用檔案 URI。

屬性	說明
encoding	文字檔傳送的編碼。
EOL	指定行尾標記。允許的值如下： <ul style="list-style-type: none">• LF - 僅限換行字元• CRLF - 換行 (carriage return/line feed) 字元序列

<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 668 頁的『檔案傳送要求訊息格式』](#)。

<name>

name 的值可以是任何字串。

範例

以下範例說明針對下列每一個排定的傳送動作，提供符合此綱目的 XML 訊息：

- [建立排定的傳送](#)
- [取消排定的傳送](#)
- [排程傳送到期](#)

以與標準傳送相同的方式，記載由排程啟動的傳送。如需由排程啟動的傳送的日誌訊息範例，請參閱 [第 652 頁的『排定的檔案傳送日誌訊息範例』](#)。

排定檔案傳送日誌訊息範例

些範例說明在排定的傳送動作執行時，會將訊息發佈至 SYSTEM.FTE 主題，其主題字串為 Log/agent_name/schedule_ID。

排定的傳送日誌訊息

代理程式每次處理排定的檔案傳送要求時，排程日誌訊息就會發佈至協調佇列管理程式（在其 SYSTEM.FTE/Log/agent name/schedule ID 主題）。此訊息符合 ScheduleLog.xsd XML 綱目。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>
```

此訊息是包含下列資訊的日誌：

- 發出要求的使用者
- 提交要求的時間
- 排定傳送的開始時間
- 來源及目的地代理程式詳細資料
- 傳送規格

<schedulelog> 元素的 ID 屬性是此排定傳送的唯一 ID（在來源代理程式中）。此 ID 可用以將排程項目與實際的檔案傳送產生關聯。

提交的 <action> 元素值確認已接收要求。

排定的傳送取消日誌訊息

當代理程式收到取消擱置排定檔案傳送的要求時，會將下列訊息發佈至 `SYSTEM.FTE/Log/agent_name` 主題：

```
<?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 屬性值對應於排程訊息中的擱置傳送要求 ID 的 ID。

排定的傳送到期日誌訊息

當現行時間與排程清單中最早擱置檔案傳送的時間相符時（由 <next> 元素的值指出），就會發佈排程日誌訊息來指出排定的傳送項目已過期：

```
<?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>
```

<action> 元素值 "expire" 確認排程項目現在已從排程清單中移除，且正在處理。將會發佈代理程式的排程訊息，且過期項目已不存在。

相關參考

第 658 頁的『排定的檔案傳送日誌訊息格式』

代理程式每次處理排定的檔案傳送要求時，排程日誌訊息就會發佈至協調佇列管理程式（在其 `SYSTEM.FTE/Log/agent name/schedule ID` 主題）。此訊息符合 `ScheduleLog.xsd` XML 綱目。

第 652 頁的『排定的檔案傳送日誌訊息範例』

進行傳送時，會將訊息發佈至 SYSTEM.FTE 主題，其主題字串為 `Log/agent_name/transfer_ID`。XML 範例顯示依據排程而進行檔案傳送時所建立的日誌訊息。

MFT 監視器日誌訊息格式

監視器日誌訊息會發佈至 SYSTEM.FTE 主題，其主題字串為 `Log/agent_name/monitors/monitor_name/monitor_ID`。

如果您想要收集資料或檢視監視器動作，請將訂閱設為相關監視器適用的萬用字元主題。例如：

```
Log/#
```

或

```
Log/agent_name/#
```

這可以是可延續或不可延續的訂閱。關閉訂閱應用程式與佇列管理程式的連線時，可延續訂閱會繼續存在。不可延續訂閱只有在訂閱應用程式與佇列管理程式的連線開啟時才會存在。

`MonitorLog.xsd` 綱目文件位於 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目錄中。`MonitorLog.xsd` 綱目會匯入 `fteutils.xsd`，它位於相同的目錄。

綱目

下列綱目說明在監視器日誌 XML 訊息中有效的元素。

```
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
  <xsd:include schemaLocation="fteutils.xsd"/>
  <xsd:element name="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:restriction>
  </xsd:simpleType>
</xsd:schema>
```

```

        <xsd:enumeration value="stop"/>
        <xsd:enumeration value="triggerSatisfied"/>
        <xsd:enumeration value="triggerNotSatisfied"/>
        <xsd:enumeration value="triggerFail"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="monitorMetaDataType">
    <xsd:sequence>
        <xsd:element name="originalMetaData" type="metaDataSetType" maxOccurs="1" minOccurs="0"/>
        <xsd:element name="updatedMetaData" type="metaDataSetType" maxOccurs="unbounded"
minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="taskXMLRequestType">
    <xsd:sequence>
        <xsd:element name="originalRequest" type="xsd:string" maxOccurs="1" minOccurs="1"/>
        <xsd:element name="updatedRequest" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="taskId" type="xsd:string" use="required"/>
</xsd:complexType>

<xsd:complexType name="referencesType">
    <xsd:sequence>
        <xsd:element name="createRequest" type="xsd:string" maxOccurs="1" minOccurs="0"/>
        <xsd:element name="taskRequest" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="monitorXMLRequestType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute name="type" type="xmlContentEnumType" use="required"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:simpleType name="xmlContentEnumType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="escapedXML"/>
    </xsd:restriction>
</xsd:simpleType>
</xsd:schema>

```

瞭解監視器日誌訊息

下列清單中說明在監視器日誌訊息中使用的元素及屬性：

<monitorLog>

群組元素，包含的元素說明監視器所執行的動作。

屬性	說明
version	必要項目。監視器清單訊息格式的版本。
monitorName	必要項目。監視器的名稱。對於定義監視器的代理程式而言是唯一的。
referenceId	監視器動作的 ID。

<originator>

群組元素，包含指定要求發送端的元素。

<hostName>

來源檔案所在系統的主機名稱。

<userID>

發出檔案傳送的使用者 ID。

<mqmdUserID>

選用項目。在訊息描述子 (MQMD) 中提供的 IBM MQ 使用者 ID。

<references>

參照與此監視器動作相關聯的其他訊息的 ID。

<createRequest>

用來建立監視器的 XML 要求訊息的訊息 ID。

<taskRequest>

監視器在此動作執行後所提交的 XML 要求訊息的訊息 ID。

<action>

與此日誌訊息相關聯的已執行動作。元素內的值可以是下列其中一項：create、delete、start、stop、triggerSatisfied、triggerNotSatisfied 或 triggerFail。

<monitorAgent>

監視資源的代理程式。

屬性	說明
代理程式 (agent)	必要項目。代理程式的名稱。
QMgr	選用項目。代理程式所連接的佇列管理程式的名稱。
bridgeURL	選用項目。如果代理程式是通訊協定橋接器代理程式，則為通訊協定伺服器的 URL。

<status>

所記載的資源監視器動作的狀態。

屬性	說明
resultCode	必要項目。動作傳回的整數結果碼。

<supplement>

所記載的資源監視器動作狀態的其他相關資訊。

<monitorMetaData>

群組元素，包含 <originalMetaData> 及 <updatedMetaData> 元素。

<originalMetaData>

一個元素，包含一個以上說明動作執行之前監視器 meta 資料的 <metadata> 元素。

<updatedMetaData>

一個元素，包含一個以上說明動作執行之後監視器 meta 資料的 <metadata> 元素。

<metadata>

定義 meta 資料鍵值組。索引鍵是元素的屬性；值是元素的內容。

屬性	說明
key	meta 資料的索引鍵。

<monitorExits>

群組元素，包含一個以上 <exit> 元素。

<exits>

說明資源監視器所執行之結束程式的元素。

屬性	說明
名	必要項目。資源監視器結束程式的名稱。

<status>

所記載的資源監視器結束程式的狀態。

屬性	說明
resultCode	必要項目。結束程式傳回的整數結果碼。

<supplement>

所記載的資源監視器結束程式狀態的其他相關資訊。

<jobDetails>

包含單一 <name> 元素的元素。

<name>

工作的名稱。

<taskXMLRequest>

群組元素，包含 <originalRequest> 及 <updatedRequest> 元素。

屬性	說明
taskId	作業要求訊息的 ID。

<originalRequest>

一個元素，包含監視器執行之作業的跳出 XML 要求訊息。

<updatedRequest>

一個元素，包含監視器執行之作業的已更新跳出 XML 要求訊息。

<monitorXMLRequest>

監視器 XML 要求。

屬性	說明
類型	必要項目。 <monitorXMLRequest> 元素內的監視器 XML 要求資料格式。唯一的有效值是 escapedXML。

範例

以下範例說明針對下列每一個監視器動作，提供符合此綱目的 XML 訊息：

- [建立監視器](#)
- [在監視器輪詢資源時，滿足監視器的條件](#)
- [在監視器輪詢資源時，不滿足監視器的條件](#)
- [刪除監視器](#)

相關參考

[第 666 頁的『MFT 監視器日誌訊息範例』](#)

這些範例說明在監視器動作執行時，會將訊息發佈至 SYSTEM.FTE 主題，其主題字串為 Log/agent_name/monitor_ID。

MFT 監視器日誌訊息範例

這些範例說明在監視器動作執行時，會將訊息發佈至 SYSTEM.FTE 主題，其主題字串為 Log/agent_name/monitor_ID。

監視器已建立的日誌訊息

```
<?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>
```

監視條件滿足的日誌訊息

```
<?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>
          &lt;/taskXMLRequest>
        </monitorLog>
```

監視條件不滿足的日誌訊息

```
<?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>
```

監視器已刪除的日誌訊息

```
<?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>
```

檔案傳送要求訊息格式

檔案傳送是由送達代理程式指令佇列的 XML 訊息起始，通常是由使用者發出檔案傳送指令或使用「IBM MQ Explorer」外掛程式來進行。The transfer request XML must conform to the FileTransfer.xsd schema

and have the <request> element as the root element. FileTransfer.xsd 綱目文件位於 MQ_INSTALLATION_PATH/mqft/samples/schema 目錄中。FileTransfer.xsd 綱目會匯入 fteutils.xsd，它位於相同的目錄中。

檔案傳送訊息可具有下列三個根元素之一：

- <request> - 用於新的檔案傳送要求、受管理的呼叫要求或刪除擱置中的排定傳送
- <cancel> - 用於取消正在進行的檔案傳送
- <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 [使用傳送定義檔](#).

綱目

下列綱目說明哪些元素在傳送要求 XML 訊息中是有效的。

```
<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>

```

瞭解傳送要求訊息

下列清單中說明在傳送要求訊息中使用的元素及屬性：

元素說明

<request>

群組元素，包含指定檔案傳送要求需要的所有元素。

屬性	說明
version	指定此元素的版本，由 Managed File Transfer 提供。

<managedTransfer>

群組元素，包含單一檔案傳送或單一檔案傳送群組需要的所有元素。

<deleteScheduledTransfer>

群組元素，包含用來取消排程傳送的發送端及 ID 資訊。

<managedCall>

群組元素，包含程式或執行檔的單一受管理呼叫需要的所有元素。

<ID>

唯一 ID，指定要從擱置的排定傳送清單中刪除的傳送要求。

<originator>

群組元素，包含指定要求發送端的元素。

<hostName>

來源檔案所在系統的主機名稱。

<userID>

發出檔案傳送的使用者 ID。

<mqmdUserID>

選用項目。在訊息描述子 (MQMD) 中提供的 IBM MQ 使用者 ID。

<schedule>

群組元素，說明檔案傳送的排定時間、重複行為及下次發生時間。

<submit>

指定排定傳送應該開始的日期和時間。

屬性	說明
timebase	指定要使用的時區。此屬性可具有下列其中一個值： <ul style="list-style-type: none">• source - 使用來源代理程式的時區• admin - 使用發出此指令的管理者的時區• UTC - 使用世界標準時間 (UTC)
時區	根據 timebase 值的時區說明

<repeat>

群組元素，包含排定傳送多久重複一次、排定傳送重複的次數及排定傳送何時停止重複的相關詳細資料。

<frequency>

在重複傳送之前必須經歷的時段。

屬性	說明
區間	間隔單位，必須是下列其中一個值：

屬性	說明
	<ul style="list-style-type: none"> • 分鐘 • 時 • 日 • 週 • 月 • 年

<expireTime>

選用元素，指定排定傳送停止重複的日期和時間。此元素與 <expireCount> 元素互斥。

<expireCount>

選用元素，指定排定檔案傳送到停止之前發生的次數。此元素與 <expireTime> 元素互斥。

<sourceAgent>

指定來源檔案所在系統上的代理程式名稱。

屬性	說明
代理程式 (agent)	指定代理程式的名稱。
QMgr	代理程式佇列管理程式的名稱。

<destinationAgent>

指定系統上要做為檔案傳送目的地的代理程式名稱。

屬性	說明
代理程式 (agent)	指定代理程式的名稱。
QMgr	代理程式佇列管理程式的名稱。
hostName	代理程式佇列管理程式的主機名稱或 IP 位址。
portNumber	用於與目的地代理程式佇列管理程式進行用戶端連線的埠號。
channel	用來連接至目的地代理程式佇列管理程式的通道名稱。

<trigger>

選用元素，指定要進行檔案傳送時必須具備的條件。

屬性	說明
log	指出是否記載觸發失敗的旗標。有效值如下所示： <ul style="list-style-type: none"> • yes - 會針對失敗的觸發傳送建立日誌項目 • no - 不會針對失敗的觸發傳送建立日誌項目

<fileExist>

指定以逗點區隔的檔案名稱清單，這些檔案名稱位於與來源代理程式相同的系統上。如果此名稱清單中的檔案滿足觸發的條件，就會進行傳送。此元素與 <fileSize> 元素互斥。

屬性	說明
comparison	指出如何根據名稱清單評估來源檔案名稱。有效值如下所示： <ul style="list-style-type: none"> • = 名稱清單中至少有一個檔案名稱必須符合

屬性	說明
	<ul style="list-style-type: none"> • != 名稱清單中至少有一個檔案不存在
value	指出比較類型： <ul style="list-style-type: none"> • exist: 檔案必須存在

<fileSize>

指定以逗點區隔的檔案名稱清單，這些檔案名稱位於與來源代理程式相同的系統上。如果此名稱清單中的檔案滿足觸發的條件，就會進行傳送。此元素與 <fileExist> 元素互斥。

屬性	說明
comparison	指出如何根據名稱清單評估來源檔案名稱。有效值如下所示： <ul style="list-style-type: none"> • >= 名稱清單中有一個檔案名稱存在，且檔案大小最少為 value 屬性指定的大小
value	指定為整數值的檔案大小，其單位指定為下列其中一項： <ul style="list-style-type: none"> • B - 位元組 • KB - 千位元組 • MB - 百萬位元組 • GB - 十億位元組 (單位值不區分大小寫)

<reply>

指定對同步檔案傳送產生的暫時回覆佇列名稱（在指令行上使用 **-w** 參數指定）。佇列名稱是由 `command.properties` 配置檔中的索引鍵 **dynamicQueuePrefix** 或預設值 `WMQFTE.*` 所定義。（如果未指定的話）。

屬性	說明
詳細資料	回覆訊息中是否應包含詳細的傳輸結果資訊。可以針對每個傳送產生多個回覆訊息。有效值如下所示： <ul style="list-style-type: none"> • true - 必須包含詳細回覆資訊。此資訊格式與進度訊息中發佈到傳送日誌的格式相同，亦即 <transferSet> 元素。如需相關資訊，請參閱 第 637 頁的『檔案傳送日誌訊息格式』。僅當傳輸來源代理程式的 <code>enableDetailedReplyMessages</code> 內容設定為 true 時，才會顯示詳細回覆資訊。 • false - 無須包含詳細回覆資訊。 預設值是 false。
QMGR	產生暫時動態佇列以接收回覆的指令佇列管理程式的名稱。
persistent	寫入回覆佇列的訊息是否具有持續性。有效值如下所示： <ul style="list-style-type: none"> • true - 訊息具有持續性 • false - 訊息不具有持續性 • qdef - 訊息的持續性由回覆佇列的內容定義 預設值是 false。

<transferSet>

指定您想要一起執行的檔案傳送群組，或您想要一起執行的受管理呼叫群組。在傳輸期間，<transferSet> 是包含 <item> 元素的群組元素。

屬性	說明
priority	傳送的優先順序層次。優先順序是範圍 0-9 中的值，其中 0 是最低優先順序。預設優先順序層次是 0，依預設，傳送時會使用來源代理程式的優先順序層次。

<metaDataSet>

選用群組元素，包含一個以上 meta 資料項目。

<metaData>

指定使用者定義的 meta 資料，其傳送至代理程式呼叫的結束點。此元素包含字串形式的 meta 資料值。

屬性	說明
key	字串形式的 meta 資料名稱

<call>

群組元素，包含指定要呼叫的程式或執行檔的 <command> 元素。

<command>

指定要呼叫的程式或執行檔。此指令必須位於代理程式指令路徑中。如需相關資訊，請參閱第 464 頁的表 45。此元素可以包含選用的 <argument> 元素。

屬性	說明
名	指令的名稱。
successRC	此指令傳回的成功回覆碼。預設值是 0。
retryCount	指令失敗時重試指令的次數。
retryWait	指令重試之間要等待的時間（以秒為單位）。
類型	要呼叫的程式類型。有效值是 antscript、jcl 或 executable。

<argument>

指定要傳遞至指令的引數。

<item>

群組元素，包含指定來源及目的地檔案名稱和位置的元素。


屬性	說明
mode	指定傳送模式為二進位或文字模式。
checksumMethod	指定產生訊息摘要以建立數位簽章的雜湊演算法類型。有效值是 MD5 或 none。



<source>

群組元素，指定來源系統上的檔案以及傳送完成之後是否移除這些檔案

屬性	說明
recursive	指定當 <source> 元素是目錄或包含萬用字元時，在子目錄中遞迴地傳送檔案。
disposition	指定當 <source> 已順利傳送至其目的地時，對 <source> 元素採取的動作。有效值如下所示： <ul style="list-style-type: none"> • leave - 來源檔案維持不變。 • delete - 順利傳送來源檔案之後，即從來源系統中刪除來源檔案。

<file>

指定傳送來源。  若為 Multiplatforms，傳送來源可以是檔案或目錄名稱。對於 z/OS 平台，傳送來源可以是檔案、目錄、資料集或 PDS 名稱。請使用完整路徑，其格式與作業系統一致，例如 C:/from/here.txt。請不要使用檔案 URI。

屬性	說明
別名 (alias)	指定來源檔案的別名。此別名是來源檔案的名稱，不包括針對傳送所指定的任何目錄路徑。
EOL	指定文字傳送的行尾標記。有效值為： <ul style="list-style-type: none">• LF - 僅限換行字元• CRLF - 換行 (carriage return/line feed) 字元序列
encoding	文字檔傳送的來源檔案編碼。
 delimiter	指定在記錄導向的來源檔案（例如，z/OS 資料集）中，記錄之間所內含的定界字元。將定界字元值指定為介於 00-FF 範圍內的兩個十六進位數字，並以 x 作為字首。例如，x12 或 x03,x7F。
delimiterType	指定包括在目的地檔案中的個別訊息資料之後的定界字元類型。有效值如下所示： <ul style="list-style-type: none">• binary - 十六進位定界字元 僅當您已啟用 7.0.4.1 功能時，此屬性才可用。
delimiterPosition	指定將記錄導向的來源檔案記錄寫入一般檔案時，要插入定界字元的位置。有效值如下所示： <ul style="list-style-type: none">• prefix - 將定界字元插入到目的地檔案中，每一個來源記錄導向檔案記錄的資料之前。• postfix - 將定界字元插入到目的地檔案中，每一個來源記錄導向檔案記錄的資料之後。
includeDelimiterInFile	指定是否在記錄導向的來源檔案記錄之間併入定界字元。
 keepTrailingSpaces	指定在文字模式傳送時，是否在從固定長度格式資料集讀取的來源記錄中，保留尾端空格。預設值是刪除尾端空格。有效值如下所示： <ul style="list-style-type: none">• true - 在從固定長度格式資料集讀取的來源記錄中，保留尾端空格。• false - 在從固定長度格式資料集讀取的來源記錄中，刪除尾端空格。

<queue>

與 <source> 元素一起使用時，指定要從中傳送的佇列名稱，該佇列必須位於來源代理程式佇列管理程式上。請使用 *QUEUE* 格式。請不要併入佇列管理程式名稱，該佇列必須在來源代理程式佇列管理程式上。如果您在 <destination> 元素內使用 <queue> 元素，就不能在 <source> 元素內使用它。



屬性	說明
useGroups	指定是否只從來源佇列傳送第一個完整訊息群組。有效值如下所示： <ul style="list-style-type: none">• true - 只傳送第一個完整訊息群組• false - 傳送來源佇列上的所有訊息
groupId	指定要從來源佇列讀取的訊息群組。只有在 useGroups 屬性的值為 true 時，此屬性才有效。
delimiterType	指定包括在目的地檔案中的個別訊息資料之後的定界字元類型。有效值如下所示：

屬性	說明
	<ul style="list-style-type: none"> • text - 文字或 Java 文字定界字元 • binary - 十六進位定界字元
delimiter	指定包括在目的地檔案中的個別訊息資料之間的定界字元。
delimiterPosition	指定將定界字元併入在目的地檔案中的個別訊息資料之前或之後。有效值如下所示： <ul style="list-style-type: none"> • prefix - 將定界字元併入在資料之前 • postfix - 將定界字元併入在資料之後
encoding	指定來源佇列編碼。
waitTime	指定來源代理程式等待下列情況的時間（以秒為單位）： <ul style="list-style-type: none"> • 訊息出現在來源佇列上（如果該佇列是空的或已變成空的） • 一個完整群組出現在來源佇列上（如果 useGroups 屬性已設為 true） 如需設定 waitTime 值的相關資訊，請參閱第 574 頁的『指定「訊息轉為檔案」傳送等待時間的指引』。

<destination>

指定目的地及行為的群組元素（如果檔案已存在於目的地代理程式上）。

您只能指定 <file> 和 <queue> 其中之一作為目的地的子元素。

屬性	說明
類型	目的地的類型。有效值如下所示： <ul style="list-style-type: none"> • file - 將檔案指定為目的地 • directory - 將目錄指定為目的地 •  dataset - 指定 z/OS 資料集作為目的地 •  pds - 指定 z/OS 分割的資料集作為目的地 • queue - 指定 IBM MQ 佇列作為目的地 • filespace - 將檔案空間指定為目的地 只有在 <destination> 元素具有 <queue> 的子元素時，值 佇列 才有效。 只有在 <destination> 元素具有 <filespace> 的子元素時，值 檔案空間 才有效。 只有在 <destination> 元素具有 <file> 的子元素時，其他值才有效。
exist	指定如果目的地系統上存在目的地檔案時要採取的動作。有效值如下所示： <ul style="list-style-type: none"> • error - 報告錯誤，且不傳送檔案。 • overwrite - 改寫現有目的地檔案。 如果 <destination> 元素具有 <queue> 或 <filespace> 的子元素，則此屬性無效。

<file>

指定先前說明的 <destination> 元素的其他設定。請使用完整路徑，其格式與作業系統一致，例如 C:/from/here.txt。請不要使用檔案 URI。

屬性	說明
別名 (alias)	指定 <destination> 檔案的別名。此別名是來源檔案的名稱，不包括針對傳送所指定的任何目錄路徑。
encoding	文字檔傳送的 <destination> 檔編碼。
EOL	指定文字傳送的行尾標記。有效值為： <ul style="list-style-type: none"> • LF - 僅限換行字元 • CRLF - 換行 (carriage return/line feed) 字元序列
truncateRecords	選用項目。指定截斷長度超過 LRECL 資料集屬性的 <destination> 記錄。 <ul style="list-style-type: none"> • True - 將記錄截斷 • False - 將記錄換行 預設值為 false。

<queue>

與 **<destination>** 元素一起使用時，指定要傳送至其中的佇列名稱，該佇列可以位於連接至目的地代理程式佇列管理程式的任何佇列管理程式上。請使用 *QUEUE@QM* 格式，其中 *QUEUE* 是放置訊息的佇列名稱，*QM* 是佇列所在的佇列管理程式。如果您在 **<source>** 元素內使用 **<queue>** 元素，就不能在 **<destination>** 元素內使用它。

屬性	說明
delimiter	將檔案分割成多個訊息的定界字元。
delimiterType	指定定界字元的類型。有效值如下所示： <ul style="list-style-type: none"> • text - Java 正規表示式 • binary - 十六進位位元組序列 • size - 位元組、千位元組或百萬位元組的數目。例如，1 B、1 K 或 1 M。
delimiterPosition	指定預期是在併入個別訊息中的資料之前還是之後會有定界字元。有效選項如下所示： <ul style="list-style-type: none"> • prefix - 預期定界字元在資料之前 • postfix - 預期定界字元在資料之後
includeDelimiterInMessage	此布林值指定是否在訊息結尾併入用來將檔案分割成多個訊息的定界字元。
encoding	指定目的地佇列編碼。
persistent	指定訊息是否具有持續性。有效值如下所示： <ul style="list-style-type: none"> • true - 訊息具有持續性 • false - 訊息不具有持續性 • qdef - 訊息的持續性值由目的地佇列的設定定義
setMqProps	此布林值指定是否將 IBM MQ 訊息內容設定在檔案的第一個訊息上，且在發生錯誤時是否將所有訊息寫入佇列中。
unrecognisedCodePage	指定如果目的地佇列管理程式無法辨識資料字碼頁，則文字模式傳送是失敗，還是執行轉換。有效值如下所示： <ul style="list-style-type: none"> • fail - 傳送報告失敗

屬性	說明
	<ul style="list-style-type: none"> binary - 資料會轉換成目的地字碼頁，且說明資料格式的 IBM MQ 訊息標頭會設定為 MQFMT_NONE。 預設行為是 fail。

<filespace>

群組元素，指定要傳送至其中的檔案空間名稱。

<name>

與 <filespace> 元素一起使用時，此元素的值指定檔案空間的名稱。

<preSourceCall>

群組元素，指定在傳送開始之前，要在傳送來源呼叫的指令。

<postSourceCall>

群組元素，指定在傳送完成之後，要在傳送來源呼叫的指令。

<preDestinationCall>

群組元素，指定在傳送開始之前，要在傳送目的地呼叫的指令。

<postDestinationCall>

群組元素，指定在傳送完成之後，要在傳送目的地呼叫的指令。

<command>

與 <preSourceCall>、<postSourceCall>、<preDestinationCall> 或 <postDestinationCall> 元素一起使用時，此元素指定要呼叫的指令。此指令必須位於代理程式指令路徑中。如需相關資訊，請參閱第 464 頁的表 45。

屬性	說明
名	要執行的指令名稱。
successRC	指令順利執行時所預期的回覆碼。

<argument>

與 <command> 元素一起使用時，此元素指定要傳入指令中的引數。<command> 元素內可以有任意數量的 <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 637 頁的『檔案傳送日誌訊息格式』。

<name>

與 <job> 元素一起使用時，此元素的值指定工作的名稱。

<transferSpecifications>

群組元素，包含用於多個傳送群組的 <item> 元素。如需如何使用此元素的進一步詳細資料，請參閱使用傳送定義檔。

<cancel>

群組元素，包含取消進行中的檔案傳送需要的所有元素。

屬性	說明
version	指定此元素的版本，由 Managed File Transfer 提供。

<transfer>

與 <cancel> 元素一起使用時，此元素的值指定要取消的傳送要求 ID。

<job>

包含工作資訊的群組元素。

<jobName>

指定邏輯工作 ID。

檔案傳送取消訊息格式

檔案傳送要求會傳回 48 個字元的 ID，以識別特定代理程式的傳送。這個 ID 可用來取消傳送。

瞭解傳送取消訊息

以下說明在傳送取消訊息中使用的元素及屬性：

<cancel>

群組元素，包含取消進行中的檔案傳送需要的所有元素。

屬性	說明
version	指定此元素的版本，由 Managed File Transfer 提供。

<originator>

群組元素，包含指定要求發送端的元素。

<hostName>

來源檔案所在系統的主機名稱。

<userID>

發出檔案傳送的使用者 ID。

<mqmdUserID>

選用項目。在訊息描述子 (MQMD) 中提供的 IBM MQ 使用者 ID。

<transfer>

與 <cancel> 元素一起使用時，此元素的值指定要取消的傳送要求 ID。

<job>

選用項目。包含工作資訊的群組元素。

<jobName>

指定邏輯工作 ID。

範例

針對下列每一個要求，提供了符合此綱目的 XML 訊息的範例：

- [建立檔案傳送](#)
- [建立非同步檔案傳送要求](#)
- [取消檔案傳送](#)
- [建立排定的傳送](#)
- [刪除排定的傳送](#)
- [建立受管理的呼叫](#)
- [建立包括受管理呼叫的檔案傳送](#)

檔案傳送要求訊息範例

您可以放置在代理程式指令佇列上以要求代理程式建立或取消傳送的訊息範例。

建立傳送要求

```
<?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>
```

建立傳送要求 - 同步

當使用者要求區塊傳輸同步要求時（亦即，他們會等待傳送完成並接收狀態訊息），放置在指令佇列上的訊息會包含一個回覆元素，指定回覆訊息要傳送至其中的佇列。下列範例顯示放置在 FTEAGENT 所使用的指令佇列上的訊息：

```
<?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>
```

在 <reply> 元素中移入了指令佇列管理程式的名稱，此佇列管理程式中已建立暫時動態佇列來接收順利（或未順利）完成傳送的相關回覆。暫時動態佇列的名稱由下列兩個部分組成：

- command.properties 配置檔中金鑰 **dynamicQueuePrefix** 所定義的字首（它是 WMQFTE。依預設）
- IBM MQ 所產生的佇列 ID

取消傳送要求

```
<?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>414D5120555322E42494E44494E47538B0F404D032C0020</transfer>
  <reply QMGR="QM_JUPITER">WMQFTE.4D400F8B20002007</reply>
</cancel>

```

相關參考

第 668 頁的『檔案傳送要求訊息格式』

檔案傳送是由送達代理程式指令佇列的 XML 訊息起始，通常是由使用者發出檔案傳送指令或使用「IBM MQ Explorer」外掛程式來進行。The transfer request XML must conform to the FileTransfer.xsd schema and have the <request> element as the root element. FileTransfer.xsd 綱目文件位於 MQ_INSTALLATION_PATH/mqft/samples/schema 目錄中。FileTransfer.xsd 綱目會匯入 fteutils.xsd，它位於相同的目錄中。

排定的檔案傳送訊息範例

以下範例說明您可以放置在代理程式指令佇列上以要求代理程式建立或刪除排程的訊息。

建立排定的傳送

```

<?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>

```

刪除排定的傳送

```

<?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>

```

相關參考

第 668 頁的『檔案傳送要求訊息格式』

檔案傳送是由送達代理程式指令佇列的 XML 訊息起始，通常是由使用者發出檔案傳送指令或使用「IBM MQ Explorer」外掛程式來進行。The transfer request XML must conform to the FileTransfer.xsd schema

and have the <request> element as the root element. FileTransfer.xsd 綱目文件位於 MQ_INSTALLATION_PATH/mqft/samples/schema 目錄中。FileTransfer.xsd 綱目會匯入 fteutils.xsd，它位於相同的目錄中。

MFT 代理程式呼叫要求訊息範例

訊息範例，您可以將訊息放在代理程式指令佇列上，以要求代理程式建立受管理呼叫，或是建立呼叫程式的傳送。

受管理呼叫要求範例

```
<?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>
    </job>
  </managedCall>
</request>
```

具有呼叫的受管理傳送要求範例

```
<?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>
  </managedTransfer>
</request>

```

相關概念

第 209 頁的『指定要使用 MFT 執行的程式』

您可以在執行 Managed File Transfer Agent 的系統上執行程式。在檔案傳送要求過程中，您可以指定程式在傳送開始前或結束後執行。此外，您也可以提交受管理的呼叫要求，以啟動不在檔案傳送要求過程中的程式。

相關參考

第 668 頁的『檔案傳送要求訊息格式』

檔案傳送是由送達代理程式指令佇列的 XML 訊息起始，通常是由使用者發出檔案傳送指令或使用「IBM MQ Explorer」外掛程式來進行。The transfer request XML must conform to the FileTransfer.xsd schema and have the <request> element as the root element. FileTransfer.xsd 綱目文件位於 MQ_INSTALLATION_PATH/mqft/samples/schema 目錄中。FileTransfer.xsd 綱目會匯入 fteutils.xsd，它位於相同的目錄中。

MFT 監視器要求訊息格式

當適用的 XML 訊息送達代理程式的指令佇列時，即會建立資源監視器；這通常是因為使用者發出 fteCreateMonitor 指令或使用「IBM MQ Explorer」介面所造成。

監視器 XML 必須符合 Monitor.xsd 綱目，使用 <monitor> 元素作為根元素。

監視器訊息可具有下列其中一個根元素：

- <monitor> - 用於建立及啟動新的資源監視器
- <deleteMonitor> - 用於停止及刪除現有監視器

fteListMonitors 指令沒有指令訊息，因為此指令會直接從 SYSTEM.FTE 主題擷取符合的監視定義。

綱目

下列綱目說明在監視器要求 XML 訊息中有效的元素。

```

<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"

```

```

                maxOccurs="1"           minOccurs="0"/>
        <xsd:element name="defaultVariables" type="defaultVariablesType"
                maxOccurs="1"           minOccurs="0"/>
    </xsd:sequence>
</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>

```

瞭解建立監視器訊息

以下說明在建立監視器訊息中使用的元素及屬性：

元素說明

<monitor>

群組元素，包含取消進行中的檔案傳送需要的所有元素。

屬性	說明
version	指定此元素的版本，由 Managed File Transfer 提供。

<name>

監視器的名稱，必須是監視器代理程式內的唯一名稱。

<description>

監視器的說明（目前未使用）。

<pollInterval>

每一次根據觸發條件檢查資源之間的時間間隔。

屬性	說明
單位：	指定輪詢間隔的時間單位。有效值為：

屬性	說明
	<ul style="list-style-type: none"> • 秒 • 分鐘 • 時 • 日 • 週 • 月 • 年

<agent>

與監視器相關聯的代理程式名稱。

<resources>

群組元素，包含指定監視器資源的元素。

<directory>

完整路徑，指定要在監視器的代理程式機器上監視的目錄。

屬性	說明
recursionLevel	除了指定的目錄外所要監視的子目錄數目。
ID	資源的唯一 ID。

<queue>

佇列名稱，指定要在監視代理程式的佇列管理程式上監視的佇列。

<triggerMatch>

群組元素，包含指定與受監視資源相互比較之觸發條件的元素。

<conditions>

群組元素，包含指定與受監視資源相互比較之條件類型的元素。

<allOf>

指定必須滿足所有內含條件的述詞。

<anyOf>

指定必須滿足任何內含條件的述詞。

<condition>

定義將納入整體監視器觸發條件的比較條件。

<name>

條件的名稱。

<resource>

識別據以比較條件的資源定義。

屬性	說明
ID	資源的唯一 ID。

如果所要監視的資源是目錄，則必須在條件中指定下列三個元素之一：

- fileMatch
- fileNoMatch
- fileSize

如果所要監視的資源是佇列，則必須在條件中指定下列兩個元素之一：

- queueNotEmpty
- completeGroups

<fileMatch>

檔案名稱符合條件的群組元素。

<pattern>

指定檔案名稱符合型樣。資源上的檔案必須符合此型樣，才能滿足條件。預設型樣是 *（任何檔案皆符合）。

<fileNoMatch>

反向檔案名稱符合條件的群組元素。

<pattern>

指定反向檔案名稱符合型樣。如果受監視資源上沒有符合的檔案，則滿足條件。預設型樣是 *（無任何檔案即符合）。

<fileSize>

檔案大小比較的群組元素。

<compare>

指定檔案大小比較。這個值必須是非負整數。

屬性	說明
operator	要使用的比較運算子。僅支援 > = '。
單元:	指定檔案大小的單位，可以是下列其中一項： <ul style="list-style-type: none"> • B - 位元組 • KB - 千位元組 • MB - 百萬位元組 • GB - 十億位元組 單位值不區分大小寫，因此 mb 的作用與 MB 相同。

<pattern>

要比對的檔案名稱型樣。預設值是 *（任何檔案皆符合）。

<queueNotEmpty>

只有在資源是佇列時，才可指定此項目。指定佇列上必須有訊息，才會觸發監視器。

<completeGroups>

只有在資源是佇列時，才可指定此項目。指定佇列上必須有完整訊息群組存在，才會觸發監視器。對於佇列上的每一個完整群組，分別執行單一傳送作業。

<reply>

選用元素，用來指定非同步要求的回覆佇列。

屬性	說明
QMGR	佇列管理程式名稱。

<tasks>

群組元素，包含的元素指定滿足監視器觸發條件時所要呼叫的作業。

<task>

群組元素，定義監視器在滿足觸發條件時所要呼叫的個別作業。目前只能指定一項作業。

<name>

作業的名稱。接受任何英數字元。

<description>

作業的說明。接受任何文字值。

<transfer>

定義傳送作業的群組元素。

<request>

定義作業類型的群組元素。其中必須包含下列其中一個繼承自 FileTransfer.xsd 綱目定義的元素：

- [managedTransfer](#)
- [managedCall](#)

屬性	說明
version	由 Managed File Transfer 提供的要求版本。其格式為 n.mm，其中 n 是主要發行版本，mm 是次要版本。例如，1.00。

<originator>

群組元素，包含指定要求發送端的元素。

<hostName>

來源檔案所在系統的主機名稱。

<userID>

發出檔案傳送的使用者 ID。

<mqmdUserID>

選用項目。在訊息描述子 (MQMD) 中提供的 IBM MQ 使用者 ID。

<job>

包含工作資訊的群組元素。

<jobName>

指定邏輯工作 ID。

<defaultVariables>

群組元素，包含個以上 variable 元素。在監視佇列時，將這些變數用於變數替代中。如需變數替代的相關資訊，請參閱第 149 頁的『[使用變數替代來自訂 MFT 資源監視器作業](#)』。

<variable>

一個元素，包含與 key 屬性指定的索引鍵相關聯的值。

屬性	說明
key	預設變數的名稱。

瞭解刪除監視器訊息

以下說明在刪除監視器訊息中使用的元素及屬性：

元素說明

<deleteMonitor>

群組元素，包含停止及刪除監視器需要的所有元素。

屬性	說明
version	指定此元素的版本，由 Managed File Transfer 提供。

<name>

要刪除的監視器名稱。

<originator>

群組元素，包含指定要求發送端的元素。

<hostName>

來源檔案所在系統的主機名稱。

<userID>

發出檔案傳送的使用者 ID。

<mqmdUserID>

選用項目。在訊息描述子 (MQMD) 中提供的 IBM MQ 使用者 ID。

<reply>

指定對要求產生的暫時回覆佇列名稱。佇列名稱是由 `command.properties` 配置檔中的索引鍵 `dynamicQueuePrefix` 定義。如果未指定此項目，佇列名稱將會使用預設值 `WMQFTE`。

屬性	說明
QMGR	產生暫時動態佇列以接收回覆的指令佇列管理程式的名稱。

範例

以下範例說明針對下列每一個監視器要求，提供符合此綱目的 XML 訊息：

- [建立監視器](#)
- [刪除監視器](#)

MFT 監視器要求訊息範例

以下範例說明您可以放置在代理程式指令佇列上以要求代理程式建立或刪除監視器的訊息。

建立監視器要求

```
<?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>
```

刪除監視器要求

```
<?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>
```

相關參考

第 684 頁的『MFT 監視器要求訊息格式』

當適用的 XML 訊息送達代理程式的指令佇列時，即會建立資源監視器；這通常是因為使用者發出 `fteCreateMonitor` 指令或使用「IBM MQ Explorer」介面所造成。

連線測試 MFT 代理程式要求訊息格式

您可以發出 `ftePingAgent` 指令或將 XML 訊息放置在代理程式指令佇列上，以對代理程式進行連線測試。連線測試代理程式要求 XML 必須符合 `PingAgent.xsd` 綱目。安裝 Managed File Transfer 之後，您可以在下列目錄中找到 `PingAgent.xsd` 綱目檔：`MQ_INSTALLATION_PATH/mqft/samples/schema`。`PingAgent.xsd` 綱目會匯入 `fteutils.xsd`，它位於相同的目錄。

當代理程式在其指令佇列上收到連線測試代理程式要求訊息時，如果代理程式在作用中，則會傳回 XML 回應訊息給指令，或傳回給將連線測試代理程式要求訊息放入指令佇列中的應用程式。代理程式的回應訊息符合 `Reply.xsd` 所定義的格式。如需此格式的相關資訊，請參閱第 694 頁的『MFT 代理程式回覆訊息格式』。

綱目

下列綱目說明哪些元素在連線測試代理程式要求 XML 訊息中是有效的。

```
<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>
```

瞭解連線測試代理程式要求訊息

下列清單中說明在連線測試代理程式要求訊息中使用的元素及屬性：

<pingAgent>

群組元素，包含指定連線測試代理程式要求需要的所有元素。

<originator>

群組元素，包含指定連線測試要求發送端需要的所有元素。

<hostName>

發出要求的機器的主機名稱。

<userID>

要求發送端的使用者名稱。

<mqmdUserID>

要求發送端的 MQMD 使用者名稱。

<agent>

要進行連線測試的代理程式。

屬性	說明
代理程式 (agent)	必要項目。代理程式的名稱。
QMGr	選用項目。代理程式所連接的佇列管理程式。

<reply>

接收代理程式所傳送回覆訊息的佇列名稱。

屬性	說明
QMGR	必要項目。回覆佇列所在的佇列管理程式的名稱。

範例

此範例顯示傳送至代理程式 AGENT_JUPITER 的連線測試代理程式訊息。如果 AGENT_JUPITER 在作用中且能夠處理代理程式要求，則會將回應訊息傳送至 QM_JUPITER 的佇列 WMQFTE.4D400F8B20003708。

```
<?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 代理程式回覆訊息格式

當代理程式在其代理程式指令佇列上收到 XML 訊息時，如果需要回應，代理程式會將 XML 回覆訊息傳送至原始訊息中定義的回覆佇列。回覆 XML 符合 Reply.xsd 綱目。Reply.xsd 綱目文件位於 MQ_INSTALLATION_PATH/mqft/samples/schema 目錄中。Reply.xsd 綱目會匯入 fteutils.xsd，它位於相同的目錄。

綱目

下列綱目說明在回覆 XML 訊息中有效的元素。

```
<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"/>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```



```

xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="TransferLog.xsd"
xmlns=""
<action time="2009-08-28T12:31:15.781Z">not_authorized</action>
<originator>
  <mqmdUserID>test1</mqmdUserID>
</originator>
<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>

```

此訊息是包含下列資訊的日誌：

- 發出要求的使用者
- 執行要求所需的 Managed File Transfer 存取權層次
- 要求的狀態
- 要求規格

瞭解未獲授權日誌訊息

以下說明在未獲授權訊息中使用的元素及屬性：

<notAuthorized>

群組元素，說明單一失敗的使用者權限檢查。

屬性	說明
version	指定此元素的版本，由 Managed File Transfer 詳述。
ID	未獲授權要求的唯一 ID。

<originator>

群組元素，包含指定要求發送端的元素。

<authority>

指定使用者執行要求動作所需的 Managed File Transfer 存取權層次。

<mqmdUserID>

在訊息描述子 (MQMD) 中提供的 IBM MQ 使用者 ID。

<action>

對符合 <notAuthorized> 元素之 ID 屬性的要求，指定授權狀態。

屬性	說明
time	指定發佈日誌項目的日期和時間（以日期時間格式表示）。

<status>

結果碼及補充訊息。

MFT 認證檔格式

MQMFTCredentials.xml 檔案包含機密的使用者 ID 及密碼資訊。MQMFTCredentials.xml 檔案中的元素必須符合 MQMFTCredentials.xsd 綱目。認證檔的安全由使用者負責維護。

V 9.0.0.2 **V 9.0.3** 從 IBM MQ 9.0.0 Fix Pack 2 和 IBM MQ 9.0.3, 您可以停用 Managed File Transfer 代理程式的預設相容模式並啟用 MQCSP 鑑別, 方法是將新參數 **useMQCSPAAuthentication** 新增至相關使用者的 MFT 認證檔 MQMFTCredentials.xml, 並將它設為 true。如需相關資訊, 請參閱啟用 MQCSP 鑑別模式。

z/OS **z/OS** 從 IBM MQ 8.0 開始, MQMFTCredentials.xsd 檔案也可以是 z/OS 上的 PDSE 成員。

在 IBM WebSphere MQ 7.5 之前, MQMFTCredentials.xml 檔案包含的資訊是保留在個別內容檔中。

MQMFTCredentials.xml 檔案必須符合 MQMFTCredentials.xsd 綱目。MQMFTCredentials.xml 綱目文件位於 Managed File Transfer 安裝的 MQ_INSTALLATION_PATH/mqft/samples/schema 目錄中。

綱目

下列綱目說明 MQMFTCredentials.xml 檔案中的有效元素。

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

    Licensed Materials - Property of IBM

    5724-H72

    Copyright IBM Corp. 2012, 2023. All Rights Reserved.

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

  <!--
    This schema defines the format of an MQMFTCredentials file. Files of this type
    store credential information for agent and logger processes. They can contain
    user names and passwords either in clear text or which have been obfuscated
    using the fteObfuscate command.
  -->

  <!-- Example mqmftCredentials.xml file:
  <?xml version="1.0" encoding="UTF-8"?>
    <tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/
MQMFTCredentials"
      xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://wmqfte.ibm.com/MQMFTCredentials MQMFTCredentials.xsd">

      <tns:logger name="LOG1" user="user1" password="passw0rd"/>
      <tns:logger name="ORACLE" userCipher="kj2h3dfkgf" passwordCipher="1a3n67eaer"/>
      <tns:file path="/home/emma/trust.jks" password="passw0rd"/>
      <tns:file path="/var/tmp/keystore.jks" passwordCipher="e71vKCg2pf"/>

      <tns:qmgr name="QM_COORD" user="tim" mqUserId="user1" mqPassword="passw0rd"/>
      <tns:qmgr name="QM_COORD" user="tom" mqUserId="user1" mqPasswordCipher="e71vKCg2pf"/>
V 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>
  </schema>
```

```

</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="passwd0rd"/>
<tns:qmgr name="QM_COORD" user="tom" mqUserIdCipher="xh5U7812x"
mqPasswordCipher="e71vKCg2pf"/>
<tns:qmgr name="QM_COORD" mqUserId="defaultUser" mqPassword="passwd0rd"/>
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>

```

瞭解 MQMFTCredentials.xml 檔案

下列清單中說明 MQMFTCredentials.xml 檔案中使用的元素及屬性。

<mqmftCredentials>

XML 文件的根元素。

<file>

傳送中的檔案。

屬性	說明
path	正在存取的金鑰或信任儲存庫檔案的路徑。
密碼	檔案的存取密碼。

<logger>

負責記載活動的日誌程式。

屬性	說明
名	日誌程式的名稱。
user	日誌程式將用來連接至其資料庫的使用者名稱。
密碼	日誌程式將用來連接至其資料庫的密碼。

<qmgr>

IBM MQ 佇列管理程式連線。

屬性	說明
名	相關聯的 IBM MQ 佇列管理程式名稱。
user	選用項目：發出連線要求的使用者名稱。
mqUserId 或 mqUserIdCipher	要提供給 IBM MQ 佇列管理程式的明碼使用者 ID (mqUserId)，或模糊文字使用者 ID (mqUserIdCipher)。
mqPassword 或 mqPasswordCipher	要提供給 IBM MQ 佇列管理程式的明碼密碼 (mqPassword)，或模糊文字密碼 (mqPasswordCipher)。

註: MQMFTCredentials.xml 檔案中可能含有機密性資訊，因此在建立該檔案時，請務必檢查檔案的權限。當使用沙盤推演時，請將其設定為排除。如需沙盤推演的相關資訊，請參閱第 39 頁的『使用 MFT 代理程式沙盤推演』。

相關概念

第 60 頁的『在 z/OS 上配置 MQMFTCredentials.xml』

如果在已啟用安全的情況下配置 Managed File Transfer (MFT)，則連線鑑別需要所有 MFT 代理程式以及連接至佇列管理程式的指令，才能提供使用者 ID 和密碼認證。

相關參考

第 36 頁的『MFT 及 IBM MQ 連線鑑別』

連線鑑別可讓佇列管理程式配置成使用提供的使用者 ID 和密碼來鑑別應用程式。如果相關聯的佇列管理程式已啟用安全，且需要認證詳細資料 (使用者 ID 和密碼)，則必須先啟用連線鑑別功能，才能順利建立與佇列管理程式的連線。連線鑑別可以在相容模式或 MQCSP 鑑別模式中執行。

第 417 頁的『fteObfuscate: 加密機密資料』

fteObfuscate 指令可為認證檔案中的機密資料加密。這會讓取得檔案存取權的人員無法讀取認證檔案的內容。

通訊協定橋接器認證檔案格式

Managed File Transfer Agent 配置目錄中的 ProtocolBridgeCredentials.xml 檔案定義通訊協定橋接器代理程式用來向通訊協定伺服器授權其本身的使用者名稱及認證資訊。

ProtocolBridgeCredentials.xml 檔必須符合 ProtocolBridgeCredentials.xsd 綱目。ProtocolBridgeCredentials.xsd 綱目文件位於 MQMFT 安裝的 MQ_INSTALLATION_PATH/mqft/samples/schema 目錄中。使用者必須自行手動建立 ProtocolBridgeCredentials.xml 檔案，因為 **fteCreateBridgeAgent** 指令不會再建立此檔案。範例檔案位於 MQMFT 安裝的 MQ_INSTALLATION_PATH/mqft/samples 目錄中。

IBM WebSphere MQ 7.5 introduced a new <agent> element that contains the <server> or <serverHost> element for the named agent.

代理程式會定期重新載入 ProtocolBridgeCredentials.xml 檔案，因此對該檔案所做的任何有效變更，都會影響代理程式的行為。預設重新載入間隔是 30 秒。透過指定 agent.properties 檔案中的代理程式內容 xmlConfigReloadInterval，即可變更此間隔。

綱目- 7.5 或更新版本

下列綱目說明 IBM MQ 8 的 ProtocolBridgeCredentials.xml 檔中哪些元素有效。

```
<schema targetNamespace="http://wmqfte.ibm.com/ProtocolBridgeCredentials" elementFormDefault="qualified"
  xmlns="https://www.w3.org/2001/XMLSchema" xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeCredentials">
  <!--
  <?xml version="1.0" encoding="UTF-8"?>
  <tns:credentials xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeCredentials"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeCredentials
  ProtocolBridgeCredentials.xsd ">
  <tns:agent name="agent1">
  <tns:serverHost name="myserver">
  <tns:user name="fred" serverPassword="pwd" serverUserId="bill"/>
  <tns:user name="jane" serverUserId="jane" hostKey="1F:2e:f3">
```

```

        <tns:privateKey associationName="test" keyPassword="pwd2">
            .... private key ...
        </tns:privateKey>
    </tns:user>
</tns:serverHost>
</tns:agent>

<tns:agent name="agent2">
    <tns:server name="server*" pattern="wildcard">
        <tns:user name="fred" serverPassword="pwd" serverUserId="bill"/>
        <tns:user name="jane" serverUserId="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>

```

瞭解 ProtocolBridgeCredentials.xml 檔

下列清單中說明 ProtocolBridgeCredentials.xml 檔中使用的元素及屬性。

<credentials>

群組元素，包含的元素說明供通訊協定橋接器代理程式用來連接至通訊協定伺服器的認證。

<agent>

包含指定代理程式之 <server> 或 <serverHost> 定義的元素。

<server>

通訊協定橋接器所連接的通訊協定伺服器。

The <server> element is not supported for 7.0.4 or earlier.

屬性	說明
名	通訊協定伺服器的名稱。
型樣	如果您使用萬用字元或正規表示式來指定通訊協定伺服器名稱的型樣，請使用 wildcard 或 regex 。
trustStorePassword 或 trustStorePasswordCipher	當 <server> 元素參照 FTPS 伺服器時需要。 用來存取信任儲存庫的密碼。如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。
keyStorePassword 或 keyStorePasswordCipher	選用項目。用來存取金鑰儲存庫的密碼。如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。

<serverHost>

通訊協定橋接器所連接的通訊協定伺服器的主機名稱。

The ProtocolBridgeCredentials.xml file can either contain <serverHost> elements or <server> elements but you cannot use a mixture of the two different types. 使用 <serverHost> 時，名稱符合通訊協定伺服器的主機名稱。When you use <server>, the name is matched against the protocol server's name (as defined in the ProtocolBridgeProperties.xml file).

屬性	說明
名	通訊協定伺服器的主機名稱或 IP 位址。
trustStorePassword 或 trustStorePasswordCipher	當 <serverHost> 元素參照 FTPS 伺服器時需要。用來存取信任儲存庫的密碼。如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。
keyStorePassword 或 keyStorePasswordCipher	選用項目。用來存取金鑰儲存庫的密碼。此為選用內容，除非設定 keyStore 屬性，在此情況下則為必要內容。如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。

<user>

從 Managed File Transfer 使用者名稱對映至通訊協定伺服器使用者名稱的使用者。

屬性	說明
名	用於 Managed File Transfer 的使用者名稱。
serverUserId 或 serverUserIdCipher	用於通訊協定伺服器的使用者名稱。如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。
serverPassword 或 serverPasswordCipher	在通訊協定伺服器上使用的使用者名稱的密碼。如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。
hostKey	伺服器主機 SSH 指紋。

<privateKey>

使用者的私密金鑰。

屬性	說明
keyPassword 或 keyStorePasswordCipher	私密金鑰的密碼。如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。
associationName	用於追蹤及記載的名稱。

通訊協定橋接器內容檔格式

代理程式配置目錄中的 ProtocolBridgeProperties.xml 檔案，會定義通訊協定檔案伺服器的內容。

ProtocolBridgeProperties.xml 檔必須符合 ProtocolBridgeProperties.xsd 綱目。

ProtocolBridgeProperties.xsd 綱目文件位於 Managed File Transfer 安裝的

MQ_INSTALLATION_PATH/mqft/samples/schema 目錄中。**fteCreateBridgeAgent** 指令會在代理程式配置目錄中建立範本檔 ProtocolBridgeProperties.xml。

代理程式會定期重新載入 ProtocolBridgeProperties.xml 檔案，因此對該檔案所做的任何有效變更，都會影響代理程式的行為。預設重新載入間隔是 30 秒。透過指定 agent.properties 檔案中的代理程式內容 xmlConfigReloadInterval，即可變更此間隔。

綱目

下列綱目說明 ProtocolBridgeProperties.xml 檔。

註: IBM WebSphere MQ 7.5 或 IBM WebSphere MQ File Transfer Edition 7.0.2 或更新版本不支援 maxReconnectRetry 及 reconnectWaitPeriod 屬性。

```
<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"/>
</attributeGroup>

```



```

        <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>

```

瞭解 ProtocolBridgeProperties.xml 檔

下列清單中說明 ProtocolBridgeProperties.xml 檔中使用的元素及屬性：

<serverProperties>

XML 文件的根元素

<credentialsFile>

包含認證的檔案路徑。若為 IBM WebSphere MQ 7.5 或更新版本，此內容值可包含環境變數。如需相關資訊，請參閱第 452 頁的『在 MFT 內容中使用環境變數』

<defaultServer>

通訊協定檔案伺服器，作為檔案傳送的預設伺服器

<ftpServer>

FTP 檔案伺服器

<sftpServer>

SFTP 檔案伺服器

<ftpsServer>

FTPS 檔案伺服器

適用於所有類型通訊協定檔案伺服器的一般伺服器屬性：

屬性	說明
名	必要項目。通訊協定檔案伺服器名稱。通訊協定伺服器名稱的長度必須至少為兩個字元、不區分大小寫，且限於英數字元及下列字元： <ul style="list-style-type: none">句點 (.)底線 (_)正斜線 (/)百分比符號 (%)
host	必要項目。您想要傳送或接收檔案的通訊協定檔案伺服器之主機名稱或 IP 位址。
port	選用項目。您想要傳送或接收檔案的通訊協定檔案伺服器之埠號。
platform	必要項目。您想要傳送或接收檔案的通訊協定檔案伺服器之平台。指定 UNIX 或 WINDOWS。請根據您在 FTP、FTPS 或 SFTP 伺服器上輸入路徑的方式，設定此內容。比方說，如果您在 Windows 上執行 FTP 伺服器，但在登入伺服器時必須輸入 UNIX 樣式的路徑（亦即，使用正斜線），請將此值設為 UNIX，而非 WINDOWS。在 Windows 上執行的伺服器，通常呈現 UNIX 樣式的檔案系統。
fileEncoding	必要項目。定義檔案伺服器所使用的字元編碼。以文字模式傳送檔案時會使用此內容，以於平台之間移動檔案時變更正確的編碼順序。例如 UTF-8。
limitedWrite	選用項目。寫入檔案伺服器時，預設模式是建立暫存檔，然後在傳送完成時重新命名該檔案。若為配置為唯寫的檔案伺服器，則會直接以其最終名稱建立檔案。此內容值可以是 true 或 false。預設值是 false。
controlEncoding	選用項目。傳送至通訊協定檔案伺服器的控制訊息的控制編碼值。此內容會影響使用的檔名編碼，且必須與通訊協定檔案伺服器的控制編碼相容。預設值是 UTF-8。

僅適用於 FTP 及 FTPS 伺服器的一般屬性：

屬性	說明
timeZone	必要項目。您想要傳送或接收檔案的通訊協定檔案伺服器之時區。例如：美洲/紐約或亞洲/東京。
locale	必要項目。您想要傳送或接收檔案的通訊協定檔案伺服器所使用之語言。例如：en_US 或 ja_JP

屬性	說明
listFormat	選用項目。清單格式，定義從通訊協定檔案伺服器傳回的檔案清單資訊的格式。使用 Windows 或 UNIX。預設值是 UNIX。
listFileRecentDateFormat	選用項目。FTP 伺服器上的 FTP 用戶端目錄清單的最近日期格式（一年內）。此屬性及 listFileOldDateFormat 屬性，可讓您重新定義通訊協定檔案伺服器傳回的預期日期格式。預設值由通訊協定檔案伺服器定義。
listFileOldDateFormat	選用項目。FTP 伺服器上的 FTP 用戶端目錄清單的舊日期格式（一年以上）。此屬性及 listFileRecentDateFormat 屬性，可讓您重新定義通訊協定檔案伺服器傳回的預期日期格式。預設值由通訊協定檔案伺服器定義。
monthShortNames	選用項目。用來解碼從通訊協定檔案伺服器所傳回的日期資訊的月份名稱取代清單。此內容由 12 個以逗點區隔的名稱清單組成，可置換預設語言環境月份值。預設值由通訊協定檔案伺服器定義。

僅適用於 FTP 伺服器的一般屬性：

屬性	說明
passiveMode	選用項目。控制 FTP 伺服器的連線為被動或主動。 如果將此內容值設為 false，連線為主動。如果值設為 true，則連線為被動。預設值是 false。

僅適用於 FTPS 伺服器的一般屬性：

屬性	說明
ftpsType	選用項目。指定使用 FTPS 通訊協定的明確或隱含格式。預設值是 explicit。
trustStore	必要項目。用來判斷是否信任 FTPS 伺服器所提供憑證的信任儲存庫位置。
trustStoreType	選用項目。信任儲存庫檔案的格式。預設值是 JKS。
keyStore	選用項目。在 FTPS 伺服器盤查時，用來提供憑證資訊的金鑰儲存庫位置。預設是讓通訊協定橋接器無法連接至已配置為需要用戶端鑑別的 FTPS 伺服器。
keyStoreType	選用項目。金鑰儲存庫檔的格式。預設值是 JKS。
ccc	選用項目。選擇完成鑑別時是否使用清除（未加密）指令通道。預設值是 false，表示在整個 FTPS 階段作業期間，指令通道一直保持加密。只有在 ftpsType 設為 explicit 時，此屬性才適用。
protFirst	選用項目。指定是否在 PBSZ/PROT 指令之前或之後向 FTPS 伺服器發出 USER/PASS 指令。預設值是 false，表示先傳送 PBSZ/PROT 指令之後再發出 USER/PASS 指令。只有在 ftpsType 設為 explicit 時，此屬性才適用。
auth	選用項目。指定在 AUTH 指令中指定的通訊協定。首先會嘗試指定的通訊協定，然後依預設嘗試 TLS、SSL、TLS-C 或 TLS-P，直到 FTPS 伺服器不再以 504 回覆碼拒絕為止。只有在 ftpsType 設為 explicit 時，此屬性才適用。

<limits>

所有類型伺服器共同的屬性及某類型伺服器特定的屬性的儲存器元素：

適用於所有類型的通訊協定檔案伺服器的一般限制屬性：

屬性	說明
maxListFileNames	選用項目。在通訊協定檔案伺服器上的目錄中掃描檔名時收集的名稱數目上限。預設值是 999999999。

屬性	說明
maxListDirectoryLevels	選用項目。通訊協定伺服器上遞迴掃描檔案名稱的目錄層次數目上限。預設值為 1000。
maxReconnectRetry (此屬性現已淘汰。)	已淘汰。 IBM WebSphere MQ 7.5 或 IBM WebSphere MQ File Transfer Edition 7.0.2 或更新版本不支援此屬性。 選用項目。通訊協定橋接器代理程式停止嘗試之前，通訊協定伺服器嘗試重新連接的次數上限。預設值是 2。
reconnectWaitPeriod (此屬性現已淘汰。)	已淘汰。 IBM WebSphere MQ 7.5 或 IBM WebSphere MQ File Transfer Edition 7.0.2 或更新版本不支援此屬性。 選用項目。嘗試重新連接之前等待的時間間隔（以秒為單位）。預設值是 10 秒。
maxSessions	選用項目。通訊協定伺服器的階段作業數目上限。此數必須大於或等於通訊協定橋接器代理程式的來源及目的地傳送次數上限的總和。預設值為代理程式內容 maxSourceTransfers、maxDestinationTransfers 及 maxCommandHandlerThreads 的值總和加上 1。如果這三個內容使用其預設值 25、25 及 5，則 maxSessions 預設值為 56。
socketTimeout	選用項目。Socket 逾時（以秒為單位）。檔案串流期間會使用此屬性的值。預設值是 30 秒。

僅適用於 SFTP 伺服器的限制屬性：

屬性	說明
connectionTimeout	選用項目。等待通訊協定檔案伺服器回應連線要求的時間（以秒為單位）。逾時表示通訊協定檔案伺服器無法使用。預設值是 30 秒。
cipherList	選用項目。指定用於在通訊協定橋接器代理程式與 SFTP 伺服器之間通訊的密碼清單（以逗點區隔）。將會根據此清單中指定的順序來呼叫密碼。此密碼在伺服器及用戶端上必須皆可用，然後才能使用它。 通訊協定橋接器代理程式支援的密碼如下所示： <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 依預設，通訊協定橋接器代理程式使用的密碼清單是 aes128-cbc , aes192-cbc , aes256-cbc。

Connect:Direct 認證檔格式

Managed File Transfer Agent 配置目錄中的 ConnectDirectCredentials.xml 檔案定義 Connect:Direct 代理程式用來向 Connect:Direct 節點進行自我授權的使用者名稱及認證資訊。

ConnectDirectCredentials.xml 檔案必須符合 ConnectDirectCredentials.xsd 綱目。ConnectDirectCredentials.xsd 綱目文件位於 MQMFT 安裝的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目錄中。範例 ConnectDirectCredentials.xml 檔案位於 MQMFT 安裝的 `MQ_INSTALLATION_PATH/mqft/samples/credentials` 目錄中。

代理程式會定期重新載入 ConnectDirectCredentials.xml 檔案，因此對該檔案所做的任何有效變更，都會影響代理程式的行為。預設重新載入間隔是 30 秒。透過在 `agent.properties` 檔案中指定代理程式內容 `xmlConfigReloadInterval`，可以變更此間隔。

綱目

下列綱目說明 ConnectDirectCredentials.xml 檔案中的有效元素。

```
<?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>

```

瞭解 ConnectDirectCredentials.xml 檔案

下列清單中說明了 ConnectDirectCredentials.xml 檔案中使用的元素及屬性。

<credentials>

包含說明認證的元素之群組元素，這些認證可供 Connect:Direct 橋接器代理程式，用來連接至 Connect:Direct 節點。

<agent>

包含指定代理程式之 <pnode> 定義元素的群組元素。

<pnode>

Connect:Direct 傳送中的主要節點 (PNODE)。此節點可起始與次要節點 (SNODE) 的連線。

屬性	說明
名	Connect:Direct 節點的名稱。此屬性的值可以是符合許多節點名稱的型樣。
型樣	指定用於 name 屬性值的型樣類型。pattern 屬性的有效值如下 <ul style="list-style-type: none"> wildcard - 使用萬用字元 regex - 使用 Java 正規表示式

<user>

提交傳送要求的 IBM MQ 使用者。

屬性	說明
名	用於 Managed File Transfer 的使用者名稱。此屬性的值可以是符合許多使用者名稱的型樣。
ignorecase	指定是否不區分名稱大小寫。ignorecase 屬性的有效值如下 <ul style="list-style-type: none"> true - 名稱不區分大小寫 false - 名稱區分大小寫
型樣	指定用於 name 屬性值的型樣類型。pattern 屬性的有效值如下

屬性	說明
	<ul style="list-style-type: none"> wildcard - 使用萬用字元 regex - 使用 Java 正規表示式
cdUserId 或 cdUserIdCipher	Connect:Direct 橋接器用來連接其相關聯 Connect:Direct 節點的使用者名稱。如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。
cdPassword 或 cdPasswordCipher	與 cdUserId 屬性所指定使用者名稱相關聯的密碼。如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。
pnodeUserId 或 pnodeUserIdCipher	Connect:Direct 主要節點的使用者名稱。如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。
pnodePassword 或 pnodePasswordCipher	與 pnodeUserId 屬性所指定使用者名稱相關聯的密碼。如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。

<snode>

在 Connect:Direct 檔案傳送期間，執行次要節點 (SNODE) 角色的 Connect:Direct 節點。

屬性	說明
名	Connect:Direct 節點的名稱。此屬性的值可以是符合許多節點名稱的型樣。
型樣	指定用於 name 屬性值的型樣類型。pattern 屬性的有效值如下 <ul style="list-style-type: none"> wildcard - 使用萬用字元 regex - 使用 Java 正規表示式
userId 或 userIdCipher	在檔案傳送期間，用來連接至此節點的使用者名稱。如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。
password 或 passwordCipher	與 userId 屬性所指定使用者名稱相關聯的密碼。如果已使用 fteObfuscate 指令，則必須使用加密版的屬性。

範例

在此範例中，Connect:Direct 橋接器代理程式會連接至 Connect:Direct 節點 pnode1。如果 IBM MQ 使用者的使用者名稱是以 fteuser 字首開頭，而且後面接著單一字元（例如 fteuser2），當他要求涉及 Connect:Direct 橋接器的傳送時，Connect:Direct 橋接器代理程式即會使用 cduser 使用者名稱及 passwd0rd 密碼，連接至 Connect:Direct 節點 pnode1。當 Connect:Direct 節點 pnode1 執行它所負責的傳送作業時，將會使用 pnodeuser 使用者名稱及 passwd0rd1 密碼。

如果 Connect:Direct 傳送中的次要節點，具有以 FISH 字首開頭的名稱，則節點 pnode1 會使用 fishuser 使用者名稱及 passwd0rd2 密碼，連接至該次要節點。如果 Connect:Direct 傳送中的次要節點，具有以 CHIPS 字首開頭的名稱，則節點 pnode1 會使用 chipsuser 使用者名稱及 passwd0rd3 密碼，連接至該次要節點。

```
<?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="passwd0rd"
        pnodeUserId="pnodeuser" pnodePassword="passwd0rd1">
        <tns:snode name="FISH*" pattern="wildcard"
          userId="fishuser" password="passwd0rd2"/>
        <tns:snode name="CHIPS*" pattern="wildcard"
          userId="chipsuser" password="passwd0rd3"/>
      </tns:user>
    </tns:pnode>
  </tns:agent>
</tns:credentials>
```

```
</tns:agent>
</tns:credentials>
```

Connect:Direct 程序定義檔案格式

Connect:Direct 橋接器代理程式配置目錄中的 ConnectDirectProcessDefinitions.xml 檔案，會指定要在檔案傳送過程中啟動的使用者定義的 Connect:Direct 程序。

ConnectDirectProcessDefinitions.xml 檔案必須符合 ConnectDirectProcessDefinitions.xsd 綱目。ConnectDirectProcessDefinitions.xsd 綱目文件位於 MFT 安裝的 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目錄中。

fteCreateCDAgent 指令會在代理程式配置目錄中建立範本 ConnectDirectProcessDefinitions.xml 檔案。

代理程式會定期重新載入 ConnectDirectProcessDefinitions.xml 檔案，因此對該檔案所做的任何有效變更，都會影響代理程式的行為。預設重新載入間隔是 30 秒。透過指定 agent.properties 檔案中的代理程式內容 `xmlConfigReloadInterval`，即可變更此間隔。

綱目

下列綱目說明 ConnectDirectProcessDefinitions.xml 檔案中的有效元素。

```
<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>

```

瞭解 ConnectDirectProcessDefinitions.xml 檔案

下列清單中說明了 ConnectDirectProcessDefinitions.xml 檔案中使用的元素及屬性。

cdProcess

XML 文件的根元素。

processSet

包含一組使用者定義的程序所有相關資訊的群組元素。

條件 (condition)

包含條件的群組元素，依據這些條件可測試傳送，以決定是否使用 processSet 元素中所含的一組程序。

match

測試變數的值是否符合給定值的條件。

屬性	說明
variable	指定變數。此變數的值會與 value 屬性的值相互比較。此變數為本質符號。如需相關資訊，請參閱第 554 頁的『與使用者定義的 Connect:Direct 程序搭配使用的替代變數』。
value	指定型樣，以符合 variable 屬性所指定之變數的值。
型樣	指定用於 value 屬性值的型樣類型。pattern 屬性的有效值為 <ul style="list-style-type: none"> wildcard - 使用萬用字元 regex - 使用 Java 正規表示式 此為選用屬性，預設值為 wildcard。

defined

測試是否已定義變數的條件。

屬性	說明
variable	指定變數。如果此變數存在，則表示滿足符合條件。此變數為本質符號。如需相關資訊，請參閱第 554 頁的『與使用者定義的 Connect:Direct 程序搭配使用的替代變數』。

處理程序

包含位置相關資訊的群組元素，在找到相符項時，將在此位置尋找要呼叫的 Connect:Direct 程序。

transfer

在執行傳送要求期間要呼叫的 Connect:Direct 程序。

屬性	說明
處理程序	選用項目。指定檔案名稱，該檔案包含在執行傳送要求期間要呼叫的 Connect:Direct 程序。此檔案路徑相對於 Connect:Direct 橋接器代理程式配置目錄。此為選用屬性，預設值是使用 MFT 產生的程序。若為 IBM WebSphere MQ 7.5 或更新版本，此內容值可包含環境變數。如需相關資訊，請參閱第 452 頁的『在 MFT 內容中使用環境變數』。

範例

在此範例中，有三個 processSet 元素。

第一個 processSet 元素指定如果傳送要求具有 **%FTESNODE** 變數 (其值符合型樣 Client*) 及 **%FTESUSER** 變數 (其值為 Admin)，則 Connect:Direct 橋接器代理程式會在傳送過程中提交位於 *agent_configuration_directory/AdminClient.cdp* 中的 Connect:Direct 程序。

第二個 processSet 元素指定如果傳送要求具有 **%FTESNODE** 變數，且其值符合型樣 Client*，則 Connect:Direct 橋接器代理程式會提交位於 *agent_configuration_directory/Client.cdp* 中的 Connect:Direct 程序，作為傳送的一部分。Connect:Direct 橋接器代理程式會依定義的順序讀取 processSet 元素，如果找到相符項，則會使用第一個相符項，而不會尋找另一個相符項。對於同時符合第一個和第二個 processSet 的條件的傳送要求，Connect:Direct 橋接器代理程式只會呼叫第一個 processSet 指定的處理程序。

第三個 processSet 元素沒有條件，且符合所有傳送。如果傳送要求不符合第一個或第二個 processSet 的條件，則 Connect:Direct 橋接器代理程式會提交由第三個條件指定的 Connect:Direct 處理程序。此程序位於 *agent_configuration_directory/Default.cdp* 中，將在傳送過程中提交。

```
<?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 節點內容檔格式

Connect:Direct 橋接器代理程式配置目錄中的 ConnectDirectNodeProperties.xml 檔案，會指定檔案傳送中所涉及的遠端 Connect:Direct 節點的相關資訊。

ConnectDirectNodeProperties.xml 檔案必須符合 ConnectDirectNodeProperties.xsd 綱目。ConnectDirectNodeProperties.xsd 綱目文件位於 MFT 安裝的 *MQ_INSTALLATION_PATH/mqft/samples/schema* 目錄中。**fteCreateCDAgent** 指令會在代理程式配置目錄中建立範本 ConnectDirectNodeProperties.xml 檔案。

代理程式會定期重新載入 ConnectDirectNodeProperties.xml 檔案，因此對該檔案所做的任何有效變更，都會影響代理程式的行為。預設重新載入間隔是 30 秒。透過指定 agent.properties 檔案中的代理程式內容 xmlConfigReloadInterval，即可變更此間隔。

綱目

下列綱目說明 ConnectDirectNodeProperties.xml 檔案中的有效元素。

```
<?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>
```

瞭解 ConnectDirectNodeProperties.xml 檔案

下列清單中說明了 ConnectDirectNodeProperties.xml 檔案中使用的元素及屬性。

nodeProperties


XML 文件的根元素。

credentialsFile

儲存機密性資訊之認證檔案的路徑。若為 IBM WebSphere MQ 7.5 或更新版本，此內容值可包含環境變數。如需相關資訊，請參閱第 452 頁的『在 MFT 內容中使用環境變數』。

node

指定一個以上的 Connect:Direct 節點。

屬性	說明
名	用於識別 Connect:Direct 節點（使用 node 元素指定的定義）名稱的型樣。型樣相符功能不區分大小寫。
型樣	指定用於 name 屬性值的型樣類型。pattern 屬性的有效值如下： <ul style="list-style-type: none">wildcard - 使用萬用字元regex - 使用 Java 正規表示式 如需 MFT 使用的正規表示式類型的相關資訊，請參閱第 554 頁的『MFT 所使用的正規表示式』。
類型	指定符合 name 屬性所給定型樣的一或多個 Connect:Direct 節點的作業系統類型。type 屬性的有效值如下： <ul style="list-style-type: none">Windows - 節點在 Windows 上執行UNIX - 節點在 UNIX 或 Linux 上執行 z/OS、zos、os/390 或 os390 - 節點在 z/OS 上執行 此屬性的值不區分大小寫。

範例

在此範例中，Connect:Direct 認證檔名指定為 ConnectDirectCredentials.xml。此程式碼範例指定下列平台連線：

- 名稱開頭為 "cdnodew" 的所有 Connect:Direct 節點都是在 Windows 平台上執行。
- 名稱開頭為 "cdnodeu" 的所有 Connect:Direct 節點都是在 UNIX 平台上執行。
- 名稱開頭為 "cdnodez" 的所有 Connect:Direct 節點都是在 z/OS 平台上執行。
- 所有其他 Connect:Direct 節點是在 UNIX 平台上執行。

Connect:Direct 橋接器代理程式從檔案開頭到結尾搜尋相符項，並使用它找到的第一個相符項。

```
<?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 綱目檔

此綱目定義其他許多 Managed File Transfer 綱目所使用的元素和類型。

綱目

```
<?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:complexContent>
    </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="filesystem" type="fileSpaceReplyType" minOccurs="0"
maxOccurs="1"/>
            </xsd:choice>
        </xsd:sequence>
        <xsd:attribute name="resultCode" type="resultCodeType" use="required"/>
    </xsd:complexType>
    <!--
        Defines the filesystem type for use with communication between a web agent
        and a web gateway
        <filesystem name="" location=""><Quota bytes=""></filesystem>
    -->
    <xsd:complexType name="fileSpaceReplyType">
        <xsd:attribute name="name" use="required" type="xsd:string"/>
        <xsd:attribute name="location" use="required" type="xsd:string"/>
        <xsd:attribute name="quota" use="required" type="xsd:long"/>

```

```

</xsd:complexType>
<!--
  Defines the destinationAgent with agent and queue manager name, plus connection
details.
-->
<destinationAgent agent="there" QMgr="far"/>
-->
<xsd:complexType name="agentClientType">
  <xsd:attribute name="agent" type="xsd:string" use="required"/>
  <xsd:attribute name="QMGr" type="xsd:string" use="optional"/>
  <xsd:attribute name="hostName" type="xsd:string" use="optional"/>
  <xsd:attribute name="portNumber" type="xsd:nonNegativeInteger" use="optional"/>
  <xsd:attribute name="channel" type="xsd:string" use="optional"/>
</xsd:complexType>
<!--
  Defines the fileURI type as string
-->
<file encoding="UTF8" EOL="CR">C:/from/here.txt</file>
-->
<xsd:complexType name="fileType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="filespace" type="filespaceNameType" use="optional"/>
      <xsd:attribute name="alias" type="xsd:string" use="optional"/>
      <xsd:attribute name="encoding" type="encodingType" use="optional"/>
      <xsd:attribute name="EOL" type="EOLType" use="optional"/>
      <xsd:attribute name="size" type="xsd:long" use="optional"/>
      <xsd:attribute name="last-modified" type="xsd:dateTime" use="optional"/>
      <xsd:attribute name="delimiter" type="xsd:string" use="optional"/>
      <xsd:attribute name="delimiterType" type="xsd:string" use="optional"/>
      <xsd:attribute name="delimiterPosition" type="delimiterPositionType"
use="optional"/>
      <xsd:attribute name="includeDelimiterInFile" type="xsd:boolean" use="optional"/>
      <xsd:attribute name="keepTrailingSpaces" type="xsd:boolean" use="optional"/>
      <xsd:attribute name="truncateRecords" type="xsd:boolean" use="optional"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<!--
  Defines the filespace type as string
-->
<filespace>
  <name>tarquin</name>
</filespace>
-->
<xsd:complexType name="filespaceType">
  <xsd:sequence>
    <xsd:element name="name" type="filespaceNameType"/>
  </xsd:sequence>
</xsd:complexType>
<!--
  Defines a name element
-->
<name>bob</name>
-->
<xsd:simpleType name="filespaceNameType">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<!--
  Defines the accepted choices for the persistent attribute.
-->
<xsd:simpleType name="persistenceType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="true"/>
    <xsd:enumeration value="false"/>
    <xsd:enumeration value="qdef"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines the queueURI type as string with all supported attributes.
-->
<queue>QUEUE@QM</queue>
-->
<xsd:complexType name="queueType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="persistent" type="persistenceType" use="optional"/>
      <xsd:attribute name="eofMarker" type="xsd:boolean" use="optional"/>
      <xsd:attribute name="setMqProps" type="xsd:boolean" use="optional"/>
      <xsd:attribute name="split" type="xsd:boolean" use="optional"/>
      <xsd:attribute name="useGroups" type="xsd:boolean" use="optional"/>
      <xsd:attribute name="delimiter" type="xsd:string" use="optional"/>
      <xsd:attribute name="delimiterType" type="xsd:string" use="optional"/>
      <xsd:attribute name="delimiterPosition" type="delimiterPositionType"
use="optional"/>
      <xsd:attribute name="includeDelimiterInMessage" type="xsd:boolean"
use="optional"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

```

        <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="filesystem" type="filesystemType"/>
      <xsd:element name="queue" type="queueType"/>
    </xsd:choice>
    <xsd:element name="attributes" type="attributeType" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="type" type="DestinationType" use="optional"/>
  <xsd:attribute name="exist" type="existType" use="optional"/>
  <xsd:attribute name="correlationString1" type="xsd:string" use="optional"/>
  <xsd:attribute name="correlationNum1" type="xsd:nonNegativeInteger" use="optional"/>
  <xsd:attribute name="correlationBoolean1" type="xsd:boolean" use="optional"/>
</xsd:complexType>
<!--
    Defines the enumeration values for destination file type
    type="file|directory|queue|dataset|pds|filesystem"
    'dataset' and 'pds' only apply to z/OS environments.
-->
<xsd:simpleType name="DestinationType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="file"/>
    <xsd:enumeration value="directory"/>
    <xsd:enumeration value="queue"/>
    <xsd:enumeration value="dataset"/>
    <xsd:enumeration value="pds"/>
    <xsd:enumeration value="filesystem"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
    Defines the enumerations values for file exists on destination behavior
    exist="error|overwrite"
-->
<xsd:simpleType name="existType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="error"/>
    <xsd:enumeration value="overwrite"/>
  </xsd:restriction>
</xsd:simpleType>

<!--
    Defines one or more file attributes
    <destination encoding=? CRLF=?>
      <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=? CRLF=?>
      <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>2008-12-0816:07</expireTime>
        ..or..
        <expireCount>2</expireCount>
    </repeat>
-->
<xsd:complexType name="repeatType">
    <xsd:sequence>
        <xsd:element name="frequency" type="freqType" maxOccurs="1" minOccurs="1"/>
        <xsd:choice minOccurs="0">
            <xsd:element name="expireTime" type="noZoneTimeType"/>
            <xsd:element name="expireCount" type="positiveIntegerType"/>
        </xsd:choice>
    </xsd:sequence>
</xsd:complexType>
<!--
    <frequency interval="hours">2</frequency>
-->
<xsd:complexType name="freqType">
    <xsd:simpleContent>
        <xsd:extension base="positiveIntegerType">
            <xsd:attribute name="interval" type="intervalType" use="required"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
<!--
    Defines positive integer type
    i.e., 1+
-->
<xsd:simpleType name="positiveIntegerType">
    <xsd:restriction base="xsd:integer">
        <xsd:minInclusive value="1"/>
    </xsd:restriction>
</xsd:simpleType>
<!--
    Defines the interval enumeration values of
    "minutes", "hours", "days", "weeks", "months" or "years"
-->
<xsd:simpleType name="intervalType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="minutes"/>
        <xsd:enumeration value="hours"/>
        <xsd:enumeration value="days"/>
        <xsd:enumeration value="weeks"/>
        <xsd:enumeration value="months"/>
        <xsd:enumeration value="years"/>
    </xsd:restriction>
</xsd:simpleType>
<!--
    Defines the interval of either "source", "admin" or "UTC"
    source = use timezone of the source Agent.
    admin = use timezone of the administrator executing the command script.
    UTC = Timezone is UTC.
-->
<xsd:simpleType name="timebaseType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="source"/>
        <xsd:enumeration value="admin"/>
        <xsd:enumeration value="UTC"/>
    </xsd:restriction>
</xsd:simpleType>
<!--
    Defines a date and time without a time zone (2008-12-08T16:07)
-->

```

```

<xsd:simpleType name="noZoneTimeType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="\n\r\t ]*\d{4}\-(0[1-9]|1[0-2])\-(0[1-9]|[1-2][0-9]|
3[0-1])T([0-1][0-9]|2[0-3]):[0-5][0-9](\+|-)\d{4}|Z)?[\n\r\t ]*" />
  </xsd:restriction>
</xsd:simpleType>
<!--
  -->
  Defines the ID element, e.g. 56
  -->
<xsd:simpleType name="idType">
  <xsd:restriction base="xsd:string" />
</xsd:simpleType>
<!--
  Defines the resultCode type -2 - 9999
  <status resultCode="8011">
  -->
<xsd:simpleType name="resultCodeType">
  <xsd:restriction base="xsd:int">
    <xsd:minInclusive value="-2" />
    <xsd:maxInclusive value="9999" />
  </xsd:restriction>
</xsd:simpleType>
<!--
  Define the metaDataSet type comprising one or more key value pairs
  <metaDataSet>
    <metaData key="name">value</metaData>
    <metaData key="name">value</metaData>
  </metaDataSet>
  -->
<xsd:complexType name="metaDataSetType">
  <xsd:sequence>
    <xsd:element name="metaData" type="metaDataType" maxOccurs="unbounded"
minOccurs="1" />
  </xsd:sequence>
</xsd:complexType>
<!--
  Define the metaData type which is made up of a key and a value
  <metaData key="name">value</metaData>
  -->
<xsd:complexType name="metaDataType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="key" type="xsd:string" use="required" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<!--
  Defines containing element for triggers
  <trigger log="yes">
    <fileExist comparison="=" value="Exist">file1</fileExist>
    <fileSize comparison=">=" value="1GB">file1</fileSize>
  </trigger>
  -->
<xsd:complexType name="triggerType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="fileExist" type="fileExistTriggerType" maxOccurs="unbounded"
minOccurs="1" />
    <xsd:element name="fileSize" type="fileSizeTriggerType" maxOccurs="unbounded"
minOccurs="1" />
  </xsd:choice>
  <xsd:attribute name="log" type="logEnabledType" use="required" />
</xsd:complexType>
<!--
  Defines the file exists trigger type
  <fileExist comparison="=" value="Exist">file1</trigger>
  -->
<xsd:complexType name="fileExistTriggerType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="comparison" type="comparisonFileExistTriggerType"
use="required" />
      <xsd:attribute name="value" type="valueFileExistTriggerType" use="required" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<!--
  Defines file size trigger type
  <fileSize comparison="=" value="1GB">file1,file2,file3</trigger>
  -->
<xsd:complexType name="fileSizeTriggerType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">

```

```

        <xsd:attribute name="comparison" type="comparisonFileSizeTriggerType"
use="required"/>
        <xsd:attribute name="value" type="valueFileSizeTriggerType" use="required"/>
    </xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<!--
    Defines the enumeration values for file exists trigger conditions
    valueFileExistTriggerType="exist|noexist"
-->
<xsd:simpleType name="valueFileExistTriggerType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="exist"/>
        <xsd:enumeration value="noexist"/>
    </xsd:restriction>
</xsd:simpleType>
<!--
    Defines the enumeration values for file exists trigger comparison operator
    comparisonFileExistTriggerType="="|"!="
-->
<xsd:simpleType name="comparisonFileExistTriggerType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="="/>
        <xsd:enumeration value="!="/>
    </xsd:restriction>
</xsd:simpleType>
<!--
    Defines the enumeration values for file size trigger comparison operator
    comparisonFileSizeTriggerType=">="
-->
<xsd:simpleType name="comparisonFileSizeTriggerType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="&gt;="/>
    </xsd:restriction>
</xsd:simpleType>
<!--
    Defines the file size value pattern
    <fileSize comparison=">=" value="10|10B|10KB|10MB|10GB">file1</fileSize>
-->
<xsd:simpleType name="valueFileSizeTriggerType">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="[0123456789]+([bB]|[kK][bB]|[mM][bB]|[gG][bB])"/>
    </xsd:restriction>
</xsd:simpleType>
<!--
    Defines the enumeration values for trigger logging enabled flag
    <trigger log="yes|no">
-->
<xsd:simpleType name="logEnabledType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="yes"/>
        <xsd:enumeration value="no"/>
    </xsd:restriction>
</xsd:simpleType>
<!--
    Defines the reply type
    <reply QMGR="QMGR name" persistent="true">Queue Name</reply>
-->
<xsd:complexType name="replyType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute name="QMGR" type="xsd:string" use="required"/>
            <xsd:attribute name="persistent" type="persistenceType" use="optional"/>
            <xsd:attribute name="detailed" type="detailedType"
use="optional"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<!--
    Defines the accepted choices for the detailed attribute.
-->
<xsd:simpleType name="detailedType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="true"/>
        <xsd:enumeration value="false"/>
    </xsd:restriction>
</xsd:simpleType>

<!--
    Defines the priority type
    <transferset priority="1">

```

```

-->
<xsd:simpleType name="priorityType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[0123456789]"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Define the job information element
  <job>
    <name>JOBNAME</name>
  </job>
-->
<xsd:complexType name="jobType">
  <xsd:sequence>
    <xsd:element name="name" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<!--
  Defines an action
  <action>
    <runCommand name="myCommand.sh"/>
  </action>
-->
<xsd:complexType name="commandActionType">
  <xsd:choice>
    <xsd:element name="command" type="commandType" maxOccurs="1" minOccurs="0"/>
  </xsd:choice>
</xsd:complexType>
<!--
  Defines a command
  <command name="runme" successRC="0" maxReplyLength="1024">
    <argument>firstArg</argument>
    <argument>secondArg</argument>
  </command>
-->
<xsd:complexType name="commandType">
  <xsd:sequence>
    <xsd:element name="argument" type="xsd:string" maxOccurs="unbounded" minOccurs="0"/>
    <xsd:element name="target" type="xsd:string" maxOccurs="unbounded" minOccurs="0"/>
    <xsd:element name="property" type="propertyType" maxOccurs="unbounded"
minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="name" type="xsd:string" use="required"/>
  <xsd:attribute name="successRC" type="xsd:string" use="optional"/>
  <xsd:attribute name="retryCount" type="nonNegativeIntegerType" use="optional"/>
  <xsd:attribute name="retryWait" type="nonNegativeIntegerType" use="optional"/>
  <xsd:attribute name="type" type="callTypeType" use="optional"/>
  <xsd:attribute name="priority" type="commandPriorityType" use="optional"/>
  <xsd:attribute name="message" type="xsd:string" use="optional"/>
</xsd:complexType>
<!--
  Defines the enumeration values for the type of a command
  type="executable|antscript|jcl"
-->
<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>

```

註: 從 IBM MQ 9.0 開始, Managed File Transfer 不支援「Web 閘道」或 Web 代理程式。

相關概念

第 624 頁的『MFT 所使用的 XML 訊息格式』

Managed File Transfer 使用 XML 格式的訊息有許多用途: 用來指令代理程式; 記載監視器、排程及傳送的相關資訊; 以及定義用於配置的資訊。XML 格式在這些用途下的邏輯結構, 由 XML 綱目進行說明。

MFT 開發應用程式參照

在檔案傳送之前或之後執行程式

使用 *fteCreateTransfer* 啟動程式的範例

您可以使用 **fteCreateTransfer** 指令, 指定程式在傳送之前或之後執行。

除了使用 **fteCreateTransfer** 以外, 還有其他方法也可在傳送之前或之後呼叫程式。如需相關資訊, 請參閱第 209 頁的『指定要使用 MFT 執行的程式』。

所有範例都可使用下列語法來指定程式:

```
[type:]commandspec[, [retrycount][, [retrywait][, successrc]]]
```

如需此語法的相關資訊, 請參閱第 367 頁的『fteCreateTransfer: 啟動新的檔案傳送』。

執行可執行程式

下列範例指定稱為 mycommand 的可執行程式, 並將 a 及 b 這兩個引數傳遞至程式。

```
mycommand(a,b)
```

若要於傳送開始前在來源代理程式 AGENT1 上執行此程式, 請使用下列指令:


```
fteCreateTransfer -sa AGENT1 -da AGENT2 -presrc mycommand(a,b)
destinationSpecification sourceSpecification
```

執行及重試可執行程式

下列範例指定未使用任何引數的可執行程式 `simple`。 `retrycount` 的指定值是 1， `retrywait` 的指定值是 5。這些值表示如果程式未傳回成功回覆碼，則會在等待五秒後重試一次。 `successsrc` 未指定任何值，因此唯一的成功回覆碼是預設值 0。

```
executable:simple,1,5
```

若要於傳送完成後在來源代理程式 AGENT1 上執行此程式，請使用下列指令：

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -postsrc executable:simple,1,5
destinationSpecification sourceSpecification
```

執行 Ant Script 並指定成功回覆碼

下列範例指定稱為 `myscript` 的 Ant Script，並將兩個內容傳遞至 Script。請使用 `fteAnt` 指令來執行此 Script。 `successsrc` 的值指定為 `>2<7&!5|0|14`，這指定回覆碼 0、3、4、6 及 14 皆指出作業成功。

```
antscript:myscript(prop1=fred,prop2=bob),,,>2<7&!5|0|14
```

若要於傳送開始前在目的地代理程式 AGENT2 上執行此程式，請使用下列指令：

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -predst
"antscript:myscript(prop1=fred,prop2=bob),,,>2<7&!5|0|14"destinationSpecification sourceSpecification
```

執行 Ant Script 並指定要呼叫的目標

下列範例指定稱為 `script2` 的 Ant Script，以及兩個要呼叫的目標 `target1` 及 `target2`。同時也會傳入內容 `prop1`，其值為 `recmf(F,B)`。此值中的逗點 (,) 及括弧皆以反斜線字元 (\) 跳出。

```
antscript:script2(target1,target2,prop1=recmf\F,B),,,>2<7&!5|0|14
```

若要於傳送完成後在目的地代理程式 AGENT2 上執行此程式，請使用下列指令：

```
fteCreateTransfer -sa AGENT1 -da AGENT2
-postdst "antscript:script2(target1,target2,prop1=recmf\F,B),,,>2<7&!5|0|14"
destinationSpecification sourceSpecification
```

在 Ant Script 中使用 meta 資料

您可以指定 Ant 作業作為下列任何呼叫以進行傳送：

- pre source
- post source
- predestination
- post destination

執行 Ant 作業時，會使用環境變數來令傳送的使用者 meta 資料變得可用。例如，您可以使用下列程式碼來存取此資料：

```
<property environment="environment"/>
<echo>${environment.mymetadata}</echo>
```

其中 `mymetadata` 是插入傳送中的某個 meta 資料的名稱。

執行 JCL Script

下列範例指定稱為 ZOSBATCH 的 JCL Script。retrycount 的指定值是 3、retrywait 的指定值是 30，successrc 的指定值是 0。這些值表示如果 Script 未傳回成功回覆碼 0，則會重試三次，每次嘗試之間會等待 30 秒。

```
jcl:ZOSBATCH,3,30,0
```

其中，ZOSBATCH 為呼叫 MYSYS.JCL 的 PDS 成員，agent.properties 檔包含 commandPath=...:/'MYSYS.JCL':... 行

若要於傳送完成後在來源代理程式 AGENT1 上執行此程式，請使用下列指令：

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -postsrc jcl:ZOSBATCH,3,30,0  
destinationSpecification sourceSpecification
```

相關概念

第 209 頁的『指定要使用 MFT 執行的程式』

您可以在執行 Managed File Transfer Agent 的系統上執行程式。在檔案傳送要求過程中，您可以指定程式在傳送開始前或結束後執行。此外，您也可以提交受管理的呼叫要求，以啟動不在檔案傳送要求過程中的程式。

相關參考

第 367 頁的『fteCreateTransfer: 啟動新的檔案傳送』

fteCreateTransfer 指令從指令行建立並啟動新的檔案傳送。此指令可立即啟動檔案傳送、將檔案傳送排定在未來的時間和日期、將排定的傳送重複一次以上，以及根據特定條件觸發檔案傳送。

搭配使用 Apache Ant 與 MFT

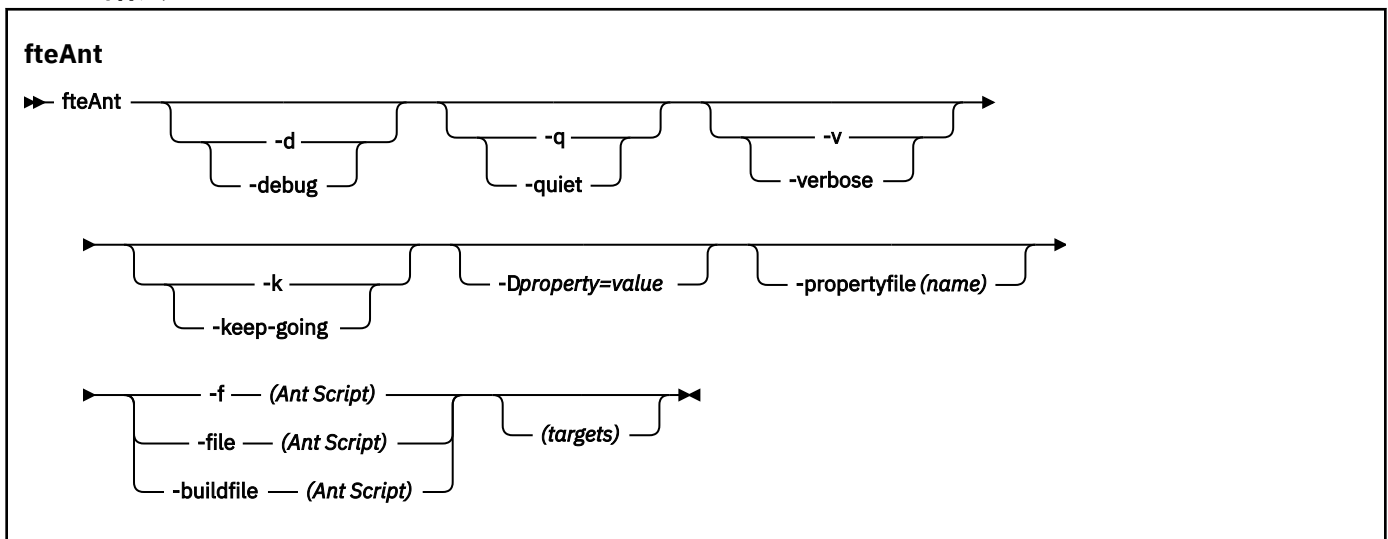
fteAnt: 在 MFT 中執行 Ant 作業

fteAnt 指令可在提供 Managed File Transfer Ant 作業的環境中執行 Ant Script。

用途

使用 **fteAnt** 指令可在具有 Managed File Transfer 的環境中執行 Ant Script。與標準 **ant** 指令不同，**fteAnt** 需要您定義 Script 檔。

語法



參數

-debug 或 -d

選用項目。產生除錯輸出。

-quiet 或 -q

選用項目。產生最小輸出。

-verbose 或 -v

選用項目。產生詳細輸出。

-keep-going 或 -k

選用項目。執行不依賴於失敗目標的所有目標。

-D *property=value*

選用項目。針對給定的 *property* 使用 *value*。設有 **-D** 的內容會優先於內容檔中所設定的內容。

使用 **com.ibm.wmqfte.propertyset** 內容可指定 Ant 作業所要使用的配置選項集。使用非預設協調佇列管理程式的名稱作為此內容的值。Ant 作業會使用與此非預設協調佇列管理程式相關聯的配置選項集。如果未指定此內容，則會使用基於預設協調佇列管理程式的預設配置選項集。如果為 Ant 作業指定 **cmdqm** 屬性，則此屬性優先於為 **fteAnt** 指令指定的配置選項集。無論您是使用預設配置選項集還是使用 **com.ibm.wmqfte.propertyset** 內容指定集，此行為均適用。

-propertyfile (*name*)

選用項目。載入檔案中的所有內容，並優先載入 **-D** 內容。

-f (*Ant script*)、-file (*Ant script*) 或 -buildfile (*Ant script*)

必要項目。指定要執行的 Ant Script 的名稱。

targets

選用項目。要從 Ant Script 執行的一個以上目標的名稱。如果您沒有指定此參數的值，則會執行 Script 的預設目標。

-version

選用項目。顯示 Managed File Transfer 指令及 Ant 版本。

-? 或 -h

選用項目。顯示指令語法。

範例

在此範例中，會執行 Ant Script `fte_script.xml` 中的目標 **copy**，指令會將除錯輸出寫入標準輸出。

```
fteAnt -d -f fte_script.xml copy
```

回覆碼

0

指令已順利完成。

1

指令未順利結束。

也可以從 Ant Script 指定其他狀態回覆碼（例如，透過使用 Ant 失敗作業來指定）。

如需相關資訊，請參閱 [失敗](#)。

MFT 提供的 Ant 作業

Managed File Transfer 提供了許多 Ant 作業，您可用於存取檔案傳送功能。

作業

- [第 732 頁的『fte:awaitoutcome Ant 作業』](#)
- [fte:call](#)
- [fte:cancel](#)
- [fte:filecopy](#)
- [fte:filemove](#)
- [fte:ignoreoutcome](#)
- [fte:ping](#)
- [fte:uuid](#)

巢狀參數

下列巢狀參數說明巢狀元素集，這些元素是數個提供的 Ant 作業共用的：

- [fte:filespec](#)
- [fte:metadata](#)
- [程式呼叫的參數](#)

fte:awaitoutcome Ant 作業

等待 **fte:filecopy**、**fte:filemove** 或 **fte:call** 作業完成。

屬性

id

必要項目。識別等待其結果的傳送。通常，這是由 [fte:filecopy](#)、[fte:filemove](#) 或 [fte:call](#) 作業的 idProperty 屬性設定的內容。

rcproperty

必要項目。將內容命名，以儲存 **fte:awaitoutcome** 作業的回覆碼。

timeout

選用項目。等待作業完成的時間量上限（以秒為單位）。逾時下限為一秒。如果您沒有指定逾時值，**fte:awaitoutcome** 作業會一直等待待決定的作業結果。

範例

在此範例中，啟動檔案複製，其 ID 儲存在 copy.id 內容中。正在複製時，可以執行其他程序。

fte:awaitoutcome 陳述式用於等待到複製作業完成。**fte:awaitoutcome** 陳述式可識別等待使用 copy.id 內容中儲存的 ID 的作業。**fte:awaitoutcome** 會將指示複製作業的結果的回覆碼儲存在稱為 copy.result 的內容中。

```
<-- 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>
```

相關參考

[第 211 頁的『搭配使用 Apache Ant 與 MFT』](#)

您可以利用 Managed File Transfer 提供的作業，將檔案傳送功能整合到 Apache Ant 工具中。

第 731 頁的『MFT 提供的 Ant 作業』

Managed File Transfer 提供了許多 Ant 作業，您可用於存取檔案傳送功能。

fte:call Ant 作業

您可以使用 **fte:call** 作業遠端呼叫 Script 及程式。

此作業可讓您將 **fte:call** 要求傳送至代理程式。代理程式會透過執行 Script 或程式並傳回結果，來處理此要求。要呼叫的指令必須可供代理程式存取。請確保 `agent.properties` 檔案中的 `commandPath` 內容值，包括所要呼叫指令的位置。由指令巢狀元素指定的任何路徑資訊，均必須相對於 `commandPath` 內容所指定的位置。依預設，`commandPath` 為空白，因此代理程式無法呼叫任何指令。如需此內容的相關資訊，請參閱第 306 頁的『`commandPath` MFT 內容』。

如需 `agent.properties` 檔案的相關資訊，請參閱第 463 頁的『MFT `agent.properties` 檔案』。

屬性

代理程式 (`agent`)

必要項目。指定要向其提交 **fte:call** 要求的代理程式。請使用下列格式指定代理程式資訊：

`agentname@qmgrname`，其中 `agentname` 為代理程式名稱，`qmgrname` 為與此代理程式直接連接的佇列管理程式名稱。

`cmdqm`

選用項目。要向其提交要求的指令佇列管理程式。請使用下列格式指定此資訊：

`qmgrname@host@port@channel`，其中：

- `qmgrname` 為佇列管理程式的名稱
- `host` 為執行佇列管理程式的系統的選用主機名稱
- `port` 為佇列管理程式接聽的選用埠號
- `channel` 為要使用的選用 SVRCONN 通道

如果省略指令佇列管理程式的 `host`、`port` 或 `channel` 資訊，則會使用 `command.properties` 檔案中所指定的連線資訊。如需相關資訊，請參閱第 460 頁的『MFT `command.properties` 檔』。

您可以使用 `com.ibm.wmqfte.propertySet` 內容，指定所要使用的 `command.properties` 檔案。如需相關資訊，請參閱 `com.ibm.wmqfte.propertySet`。

如果您不使用 `cmdqm` 屬性，此作業會預設為使用 `com.ibm.wmqfte.ant.commandQueueManager` 內容（如果已設定此內容）。如果未設定 `com.ibm.wmqfte.ant.commandQueueManager` 內容，將會嘗試連線到 `command.properties` 檔案中所定義的預設佇列管理程式。

`com.ibm.wmqfte.ant.commandQueueManager` 內容的格式與 `cmdqm` 屬性相同，即 `qmgrname@host@port@channel`。

`idproperty`

選用項目，除非您已將 `outcome` 指定為 `defer`。指定要為其指派傳送 ID 的內容名稱。提交傳送要求時將會產生傳送 ID，您可以使用傳送 ID 來追蹤傳送進度、診斷傳送問題以及取消傳送。

如果您同時還將 `outcome` 內容指定為 `ignore`，則無法指定此內容。但是，如果您同時還將 `outcome` 內容指定為 `defer`，則必須指定 `idproperty`。

`jobname`

選用項目。指派工作名稱給 **fte:call** 要求。您可以使用工作名稱來建立邏輯傳送群組。使用第 743 頁的『`fte:uuid` Ant 作業』作業可產生虛擬唯一工作名稱。如果您不使用 `jobname` 屬性，此作業會預設為使用 `com.ibm.wmqfte.ant.jobName` 內容值（如果已設定此內容）。如果您不設定此內容，則沒有任何工作名稱與 **fte:call** 要求相關聯。

`origuser`

選用項目。指定要與 **fte:call** 要求相關聯的原始使用者 ID。如果您不使用 `origuser` 屬性，此作業會預設為使用要用來執行 Ant Script 的使用者 ID。

outcome

選用項目。決定在將控制權交還給 Ant Script 之前，此作業是否等待 **fte:call** 作業完成。請指定下列其中一個選項：

await

在交還控制權之前，此作業會等待 **fte:call** 作業完成。如果將 outcome 指定為 await，則 idproperty 為選用屬性。

defer

此作業會在提交 **fte:call** 要求後立即交還控制權，並且假設稍後會使用 [awaitoutcome](#) 或 [ignoreoutcome](#) 作業，來處理呼叫作業的結果。如果將 outcome 指定為 defer，則 idproperty 為必要屬性。

ignore

如果 **fte:call** 作業的結果無關緊要，則可指定 ignore 值。此作業即會在提交 **fte:call** 要求後立即交還控制權，而不配置任何資源來追蹤指令結果。如果將 outcome 指定為 ignore，則無法指定 idproperty 屬性。

如果您沒有指定 outcome 屬性，此作業會預設為使用 await 值。

rcproperty

選用項目。指定要為其指派 **fte:call** 要求的結果碼之內容名稱。結果碼會反映 **fte:call** 要求的整體結果。

如果您同時還將 outcome 內容指定為 ignore 或 defer，則無法指定此內容。但是，如果您已將 outcome 指定為 await，則必須指定 rcproperty。

指定為巢狀元素的參數

fte:command

指定代理程式要呼叫的指令。您只能將單一 fte:command 元素與給定的 **fte:call** 作業相關聯。要呼叫的指令必須位於代理程式 agent.properties 檔案中 commandPath 內容所指定的路徑上。

fte:metadata

您可以指定要與呼叫作業相關聯的 meta 資料。此 meta 資料會記錄在呼叫作業所產生的日誌訊息中。您只能將單一 meta 資料區塊與給定的傳送元素相關聯；但此區塊可以包含許多 meta 資料片段。

範例

此範例顯示如何在佇列管理程式 QM1 上執行的 AGENT1 中呼叫指令。要呼叫的指令為 command.sh Script，而此 Script 是使用單一引數 xyz 來呼叫。command.sh 指令位於代理程式 agent.properties 檔案中 commandPath 內容所指定的路徑上。

```
<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" successsrc="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>
```

相關參考

第 211 頁的『[搭配使用 Apache Ant 與 MFT](#)』

您可以利用 Managed File Transfer 提供的作業，將檔案傳送功能整合到 Apache Ant 工具中。

第 731 頁的『[MFT 提供的 Ant 作業](#)』

Managed File Transfer 提供了許多 Ant 作業，您可用於存取檔案傳送功能。

fte:cancel Ant 作業

取消 Managed File Transfer 受管理傳送或受管理呼叫。受管理傳送可能是使用 **fte:filecopy** 或 **fte:filemove** 作業建立的。受管理呼叫可能是使用 **fte:call** 作業建立的。

屬性

agent

必要項目。指定要向其提交 **fte:cancel** 要求的代理程式。該值的格式為：

agentname@qmgrname，其中 *agentname* 是代理程式的名稱，*qmgrname* 是此代理程式直接連接的佇列管理程式的名稱。

cmdqm

選用項目。要向其提交要求的指令佇列管理程式。請使用下列格式指定此資訊：

qmgrname@host@port@channel，其中：

- *qmgrname* 為佇列管理程式的名稱
- *host* 為執行佇列管理程式的系統的選用主機名稱
- *port* 為佇列管理程式接聽的選用埠號
- *channel* 為要使用的選用 SVRCONN 通道

如果省略指令佇列管理程式的 *host*、*port* 或 *channel* 資訊，則會使用 `command.properties` 檔案中所指定的連線資訊。如需相關資訊，請參閱第 460 頁的『MFT `command.properties` 檔』。

您可以使用 `com.ibm.wmqfte.propertySet` 內容，指定所要使用的 `command.properties` 檔案。如需相關資訊，請參閱 `com.ibm.wmqfte.propertySet`。

如果您不使用 `cmdqm` 屬性，此作業會預設為使用 `com.ibm.wmqfte.ant.commandQueueManager` 內容（如果已設定此內容）。如果未設定 `com.ibm.wmqfte.ant.commandQueueManager` 內容，將會嘗試連線到 `command.properties` 檔案中所定義的預設佇列管理程式。

`com.ibm.wmqfte.ant.commandQueueManager` 內容的格式與 `cmdqm` 屬性相同，即 *qmgrname@host@port@channel*。

id

必要項目。指定要取消的傳送之傳送 ID。透過 [fte:filecopy](#) 及 [fte:filemove](#) 作業提交傳送要求時，將會產生傳送 ID。

origuser

選用項目。指定要與 **cancel** 要求相關聯的原始使用者 ID。如果未使用 `origuser` 屬性，此作業會預設為使用要用來執行 Ant Script 的使用者 ID。

範例

此範例會將 **fte:cancel** 要求傳送至指令佇列管理程式 `qm0`。**fte:cancel** 要求會針對 `transfer.id` 變數所移入的傳送 ID，以佇列管理程式 `qm1` 上的 `agent1` 為目標。執行此要求時所用的使用者 ID 為 "bob"。

```
<fte:cancel cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
  agent="agent1@qm1"
  id="{transfer.id}"
  origuser="bob"/>
```

相關參考

第 211 頁的『[搭配使用 Apache Ant 與 MFT](#)』

您可以利用 Managed File Transfer 提供的作業，將檔案傳送功能整合到 Apache Ant 工具中。

第 731 頁的『[MFT 提供的 Ant 作業](#)』

Managed File Transfer 提供了許多 Ant 作業，您可用於存取檔案傳送功能。

fte:filecopy Ant 作業

fte:filecopy 作業可在 Managed File Transfer 代理程式之間複製檔案。不會從來源代理程式刪除檔案。

屬性

cmdqm

選用項目。要向其提交要求的指令佇列管理程式。請使用下列格式指定此資訊：
`qmgrname@host@port@channel`，其中：

- `qmgrname` 為佇列管理程式的名稱
- `host` 為執行佇列管理程式的系統的選用主機名稱
- `port` 為佇列管理程式接聽的選用埠號
- `channel` 為要使用的選用 SVRCONN 通道

如果省略指令佇列管理程式的 `host`、`port` 或 `channel` 資訊，則會使用 `command.properties` 檔案中所指定的連線資訊。如需相關資訊，請參閱第 460 頁的『MFT `command.properties` 檔』。

您可以使用 `com.ibm.wmqfte.propertySet` 內容，指定所要使用的 `command.properties` 檔案。如需相關資訊，請參閱 `com.ibm.wmqfte.propertySet`。

如果您不使用 `cmdqm` 屬性，此作業會預設為使用 `com.ibm.wmqfte.ant.commandQueueManager` 內容（如果已設定此內容）。如果未設定 `com.ibm.wmqfte.ant.commandQueueManager` 內容，將會嘗試連線到 `command.properties` 檔案中所定義的預設佇列管理程式。

`com.ibm.wmqfte.ant.commandQueueManager` 內容的格式與 `cmdqm` 屬性相同，即 `qmgrname@host@port@channel`。

dst

必要項目。指定複製作業的目的地代理程式。請使用下列格式指定此資訊：`agentname@qmgrname`，其中 `agentname` 為目的地代理程式的名稱，`qmgrname` 為此代理程式直接連接的佇列管理程式的名稱。

idproperty

選用項目，除非您已將 `outcome` 指定為 `defer`。指定要為其指派傳送 ID 的內容名稱。提交傳送要求時將會產生傳送 ID，您可以使用傳送 ID 來追蹤傳送進度、診斷傳送問題以及取消傳送。

如果您同時還將 `outcome` 內容指定為 `ignore`，則無法指定此內容。但是，如果您同時還將 `outcome` 內容指定為 `defer`，則必須指定 `idproperty`。

jobname

選用項目。指派工作名稱給複製要求。您可以使用工作名稱來建立邏輯傳送群組。使用第 743 頁的『`fte:uuid Ant 作業`』作業可產生虛擬唯一工作名稱。如果您不使用 `jobname` 屬性，此作業會預設為使用 `com.ibm.wmqfte.ant.jobName` 內容值（如果已設定此內容）。如果您不設定此內容，則沒有任何工作名稱與複製要求相關聯。

origuser

選用項目。指定要與複製要求相關聯的原始使用者 ID。如果您不使用 `origuser` 屬性，此作業會預設為使用要用來執行 Ant Script 的使用者 ID。

outcome

選用項目。決定在將控制權交還給 Ant Script 之前，此作業是否等待複製作業完成。請指定下列其中一個選項：

await

在交還控制權之前，此作業會等待複製作業完成。如果將 `outcome` 指定為 `await`，則 `idproperty` 為選用屬性。

defer

此作業會在提交複製要求後立即交還控制權，並且假設稍後會使用 `awaitoutcome` 或第 742 頁的『`fte:ignoreoutcome Ant 作業`』作業，來處理複製作業的結果。如果將 `outcome` 指定為 `defer`，則 `idproperty` 為必要屬性。

ignore

如果複製作業的結果無關緊要，則可指定 `ignore` 值。此作業即會在提交複製要求後立即交還控制權，而不配置任何資源來追蹤指令結果。如果將 `outcome` 指定為 `ignore`，則無法指定 `idproperty` 屬性。

如果您沒有指定 `outcome` 屬性，此作業會預設為使用 `await` 值。

priority

選用項目。指定要與複製要求相關聯的優先順序。通常，較高優先順序的傳送要求優先於較低優先順序的要求。優先順序值必須介於 0 - 9（內含）範圍之間。優先順序值 0 為最低優先順序，值 9 為最高優先順序。如果您沒有指定 `priority` 屬性，傳送預設使用優先順序 0。

rcproperty

選用項目。指定要為其指派複製要求的結果碼之內容名稱。結果碼會反映複製要求的整體結果。如果您同時還將 `outcome` 內容指定為 `ignore` 或 `defer`，則無法指定此內容。但是，如果您將 `outcome` 指定為 `await`，則必須指定 `rcproperty`。

V 9.0.1 transferRecoveryTimeout

選用項目。設定時間量（以秒為單位），在此期間內，來源代理程式會一直嘗試回復已停止的檔案傳送。請指定下列其中一個選項：

-1

代理程式繼續嘗試回復已停止的傳送，直至傳送完成為止。使用此選項相當於代理程式在未設定此內容時的預設行為。

0

一旦進入回復，代理程式即停止檔案傳送。

>0

代理程式繼續嘗試回復已停止的傳送，直至達到已指定的正整數值所設定的時間量（以秒為單位）為止。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>
```

指出代理程式從進入回復的 6 小時內會一直嘗試回復傳送。此屬性的最大值為 999999999。

透過這種方式指定傳送回復逾時值是以每個傳送為基礎來進行設定。若要為 Managed File Transfer 網路中的所有傳送設定廣域值，您可以在 `agent.properties` 檔中新增一個內容。如需相關資訊，請參閱 [第 206 頁的『回復中檔案傳送的逾時選項』](#)。

src

必要項目。指定複製作業的來源代理程式。請使用下列格式指定此資訊：`agentname@qmgrname`，其中 `agentname` 為來源代理程式的名稱，`qmgrname` 為此代理程式直接連接的佇列管理程式的名稱。

指定為巢狀元素的參數

fte:filespec

必要項目。您必須指定至少一個用於識別要複製檔案的檔案規格。如果需要，您可以指定多個檔案規格。如需相關資訊，請參閱 [fte:filespec](#) 主題。

fte:metadata

您可以指定要與複製作業相關聯的 meta 資料。此 meta 資料會附在傳送中，並記錄在傳送所產生的日誌訊息中。您只能將單一 meta 資料區塊與給定的傳送元素相關聯；但此區塊可以包含許多 meta 資料片段。如需相關資訊，請參閱 [fte:metadata](#) 主題。

fte:presrc

指定在傳送開始之前，要在來源代理程式上進行的程式呼叫。您只能將單一 `fte:presrc` 元素與給定的傳送相關聯。如需相關資訊，請參閱 [程式呼叫](#) 主題。

fte:predst

指定在傳送開始之前，要在目的地代理程式上進行的程式呼叫。您只能將單一 `fte:predst` 元素與給定的傳送相關聯。如需相關資訊，請參閱 [程式呼叫](#) 主題。

fte:postsrc

指定在傳送完成後，要在來源代理程式上進行的程式呼叫。您只能將單一 `fte:postsrc` 元素與給定的傳送相關聯。如需相關資訊，請參閱[程式呼叫主題](#)。

fte:postdst

指定在傳送完成後，要在目的地代理程式上進行的程式呼叫。您只能將單一 `fte:postdst` 元素與給定的傳送相關聯。如需相關資訊，請參閱[程式呼叫主題](#)。

如果 `fte:presrc`、`fte:predst`、`fte:postsrc`、`fte:postdst` 及結束程式均未傳回成功狀態，則規則依照如下所示的指定順序：

1. 執行來源啟動結束程式。如果來源啟動結束程式失敗，則傳送失敗，並且不再執行任何作業。
2. 執行前置來源呼叫（如果存在）。如果前置來源呼叫失敗，則傳送失敗，並且不再執行任何作業。
3. 執行目的地啟動結束程式。如果目的地啟動結束程式失敗，則傳送失敗，並且不再執行任何作業。
4. 執行前置目的地呼叫（如果存在）。如果前置目的地呼叫失敗，則傳送失敗，並且不再執行任何作業。
5. 執行檔案傳送。
6. 執行目的地結束結束程式。這些結束程式沒有失敗狀態。
7. 如果傳送成功（如果部分檔案傳送成功，即視為成功），請執行後置目的地呼叫（如果存在）。如果後置目的地呼叫失敗，則傳送失敗。
8. 執行來源結束結束程式。這些結束程式沒有失敗狀態。
9. 如果傳送成功，請執行後置來源呼叫（如果存在）。如果後置來源呼叫失敗，則傳送失敗。

範例

以下範例顯示 `agent1` 與 `agent2` 之間的基本檔案傳送。使用用戶端傳輸模式連線，將啟動檔案傳送的指令傳送至稱為 `qm0` 的佇列管理程式。檔案傳送作業的結果指派給名為 `copy.result` 的內容。

```
<fte:filecopy cmdqmq="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>
```

以下範例顯示相同的檔案傳送，但增加了 `meta` 資料，且在傳送完成後將會在來源代理程式上啟動程式。

```
<fte:filecopy cmdqmq="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>
```

相關概念

第 206 頁的『[回復中檔案傳送的逾時選項](#)』

您可以設定時間量（以秒為單位），在此期間內，來源代理程式會一直嘗試回復已停止的檔案傳送。如果代理程式達到重試間隔的逾時值時傳送未順利完成，則傳送失敗。

相關參考

第 211 頁的『[搭配使用 Apache Ant 與 MFT](#)』

您可以利用 Managed File Transfer 提供的作業，將檔案傳送功能整合到 Apache Ant 工具中。

第 731 頁的『MFT 提供的 Ant 作業』

Managed File Transfer 提供了許多 Ant 作業，您可用於存取檔案傳送功能。

fte:filemove Ant 作業

fte:filemove 作業可在 Managed File Transfer 代理程式之間移動檔案。將檔案從來源代理程式成功傳送至目的地代理程式後，會從來源代理程式刪除檔案。

屬性

cmdqm

選用項目。要向其提交要求的指令佇列管理程式。請使用下列格式指定此資訊：

qmgrname@host@port@channel，其中：

- *qmgrname* 為佇列管理程式的名稱
- *host* 為執行佇列管理程式的系統的選用主機名稱
- *port* 為佇列管理程式接聽的選用埠號
- *channel* 為要使用的選用 SVRCONN 通道

如果省略指令佇列管理程式的 *host*、*port* 或 *channel* 資訊，則會使用 `command.properties` 檔案中所指定的連線資訊。如需相關資訊，請參閱第 460 頁的『MFT `command.properties` 檔』。

您可以使用 `com.ibm.wmqfte.propertySet` 內容，指定所要使用的 `command.properties` 檔案。如需相關資訊，請參閱 `com.ibm.wmqfte.propertySet`。

如果您不使用 `cmdqm` 屬性，此作業會預設為使用 `com.ibm.wmqfte.ant.commandQueueManager` 內容（如果已設定此內容）。如果未設定 `com.ibm.wmqfte.ant.commandQueueManager` 內容，將會嘗試連線到 `command.properties` 檔案中所定義的預設佇列管理程式。

`com.ibm.wmqfte.ant.commandQueueManager` 內容的格式與 `cmdqm` 屬性相同，即 *qmgrname@host@port@channel*。

dst

必要項目。指定複製作業的目的地代理程式。請使用下列格式指定此資訊：*agentname@qmgrname*，其中 *agentname* 為目的地代理程式的名稱，*qmgrname* 為此代理程式直接連接的佇列管理程式的名稱。

idproperty

選用項目，除非您已將 `outcome` 指定為 `defer`。指定要為其指派傳送 ID 的內容名稱。提交傳送要求時將會產生傳送 ID，您可以使用傳送 ID 來追蹤傳送進度、診斷傳送問題以及取消傳送。

如果您同時還將 `outcome` 內容指定為 `ignore`，則無法指定此內容。但是，如果您同時還將 `outcome` 內容指定為 `defer`，則必須指定 `idproperty`。

jobname

選用項目。指派工作名稱給移動要求。您可以使用工作名稱來建立邏輯傳送群組。使用 `fte:uuid` 作業可產生虛擬唯一工作名稱。如果您不使用 `jobname` 屬性，此作業會預設為使用 `com.ibm.wmqfte.ant.jobName` 內容值（如果已設定此內容）。如果您不設定此內容，則沒有任何工作名稱與移動要求相關聯。

origuser

選用項目。指定要與移動要求相關聯的原始使用者 ID。如果您不使用 `origuser` 屬性，此作業會預設為使用要用來執行 Ant Script 的使用者 ID。

outcome

選用項目。決定在將控制權交還給 Ant Script 之前，此作業是否等待移動作業完成。請指定下列其中一個選項：

await

在交還控制權之前，此作業會等待移動作業完成。如果將 `outcome` 指定為 `await`，則 `idproperty` 為選用屬性。

defer

此作業會在提交移動要求後立即交還控制權，並且假設稍後會使用第 732 頁的『[fte:awaitoutcome Ant 作業](#)』或第 742 頁的『[fte:ignoreoutcome Ant 作業](#)』作業，來處理移動作業的結果。如果將 `outcome` 指定為 `defer`，則 `idproperty` 為必要屬性。

ignore

如果移動作業的結果無關緊要，則可指定 `ignore` 值。此作業即會在提交移動要求後立即交還控制權，而不配置任何資源來追蹤指令結果。如果將 `outcome` 指定為 `ignore`，則無法指定 `idproperty` 屬性。

如果您沒有指定 `outcome` 屬性，此作業會預設為使用 `await` 值。

priority

選用項目。指定要與移動要求相關聯的優先順序。通常，較高優先順序的傳送要求優先於較低優先順序的要求。優先順序值必須介於 0 - 9（內含）範圍之間。優先順序值 0 為最低優先順序，值 9 為最高優先順序。如果您沒有指定 `priority` 屬性，傳送預設使用優先順序 0。

rcproperty

選用項目。指定要為其指派移動要求的結果碼之內容名稱。結果碼會反映移動要求的整體結果。

如果您同時還將 `outcome` 內容指定為 `ignore` 或 `defer`，則無法指定此內容。但是，如果您已將 `outcome` 指定為 `await`，則必須指定 `rcproperty`。

V 9.0.1 transferRecoveryTimeout

選用項目。設定時間量（以秒為單位），在此期間內，來源代理程式會一直嘗試回復已停止的檔案傳送。請指定下列其中一個選項：

-1

代理程式繼續嘗試回復已停止的傳送，直至傳送完成為止。使用此選項相當於代理程式在未設定此內容時的預設行為。

0

一旦進入回復，代理程式即停止檔案傳送。

>0

代理程式繼續嘗試回復已停止的傳送，直至達到已指定的正整數值所設定的時間量（以秒為單位）為止。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
```

指出代理程式從進入回復的 6 小時內會一直嘗試回復傳送。此屬性的最大值為 999999999。

透過這種方式指定傳送回復逾時值是以每個傳送為基礎來進行設定。若要為 Managed File Transfer 網路中的所有傳送設定廣域值，您可以在 [agent.properties](#) 檔中新增一個內容。如需相關資訊，請參閱第 206 頁的『[回復中檔案傳送的逾時選項](#)』。

src

必要項目。指定移動作業的來源代理程式。請使用下列格式指定此資訊：`agentname@qmgrname`，其中 `agentname` 為來源代理程式的名稱，`qmgrname` 為此代理程式直接連接的佇列管理程式的名稱。

指定為巢狀元素的參數

fte:filespec

必要項目。您必須指定至少一個用於識別要移動檔案的檔案規格。如果需要，您可以指定多個檔案規格。如需相關資訊，請參閱 [fte:filespec](#) 主題。

fte:metadata

選用項目。您可以指定要與檔案移動作業相關聯的 meta 資料。此 meta 資料會附在傳送中，並記錄在傳送所產生的日誌訊息中。您只能將單一 meta 資料區塊與給定的傳送元素相關聯；但此區塊可以包含許多 meta 資料片段。如需相關資訊，請參閱 [fte:metadata](#) 主題。

fte:presrc

選用項目。指定在傳送開始之前，要在來源代理程式上進行的程式呼叫。您只能將單一 fte:presrc 元素與給定的傳送相關聯。如需相關資訊，請參閱 [程式呼叫](#) 主題。

fte:predst

選用項目。指定在傳送開始之前，要在目的地代理程式上進行的程式呼叫。您只能將單一 fte:predst 元素與給定的傳送相關聯。如需相關資訊，請參閱 [程式呼叫](#) 主題。

fte:postsrc

選用項目。指定在傳送完成後，要在來源代理程式上進行的程式呼叫。您只能將單一 fte:postsrc 元素與給定的傳送相關聯。如需相關資訊，請參閱 [程式呼叫](#) 主題。

fte:postdst

選用項目。指定在傳送完成後，要在目的地代理程式上進行的程式呼叫。您只能將單一 fte:postdst 元素與給定的傳送相關聯。如需相關資訊，請參閱 [程式呼叫](#) 主題。

如果 fte:presrc、fte:predst、fte:postsrc、fte:postdst 及結束程式均未傳回成功狀態，則規則依照如下所示的指定順序：

1. 執行來源啟動結束程式。如果來源啟動結束程式失敗，則傳送失敗，並且不再執行任何作業。
2. 執行前置來源呼叫（如果存在）。如果前置來源呼叫失敗，則傳送失敗，並且不再執行任何作業。
3. 執行目的地啟動結束程式。如果目的地啟動結束程式失敗，則傳送失敗，並且不再執行任何作業。
4. 執行前置目的地呼叫（如果存在）。如果前置目的地呼叫失敗，則傳送失敗，並且不再執行任何作業。
5. 執行檔案傳送。
6. 執行目的地結束結束程式。這些結束程式沒有失敗狀態。
7. 如果傳送成功（如果部分檔案傳送成功，即視為成功），請執行後置目的地呼叫（如果存在）。如果後置目的地呼叫失敗，則傳送失敗。
8. 執行來源結束結束程式。這些結束程式沒有失敗狀態。
9. 如果傳送成功，請執行後置來源呼叫（如果存在）。如果後置來源呼叫失敗，則傳送失敗。

範例

以下範例顯示 agent1 與 agent2 之間的基本檔案移動。使用用戶端傳輸模式連線，將啟動檔案移動的指令傳送至稱為 qm0 的佇列管理程式。檔案傳送作業的結果指派給名為 move.result 的內容。

```
<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>
```

相關概念

第 206 頁的『[回復中檔案傳送的逾時選項](#)』

您可以設定時間量（以秒為單位），在此期間內，來源代理程式會一直嘗試回復已停止的檔案傳送。如果代理程式達到重試間隔的逾時值時傳送未順利完成，則傳送失敗。

相關參考

第 211 頁的『[搭配使用 Apache Ant 與 MFT](#)』

您可以利用 Managed File Transfer 提供的作業，將檔案傳送功能整合到 Apache Ant 工具中。

第 731 頁的『[MFT 提供的 Ant 作業](#)』

Managed File Transfer 提供了許多 Ant 作業，您可用於存取檔案傳送功能。

fte:ignoreoutcome Ant 作業

忽略 **fte:filecopy**、**fte:filemove** 或 **fte:call** 指令的輸出。如果您指定 **fte:filecopy**、**fte:filemove** 或 **fte:call** 作業的輸出為 `defer`，則 Ant 作業會配置資源來追蹤此輸出。如果您對輸出不再感興趣，則可以使用 **fte:ignoreoutcome** 作業來釋放這些資源。

屬性

id

必要項目。識別不再感興趣的輸出。一般而言，您可以使用您使用第 735 頁的『[fte:filecopy Ant 作業](#)』、第 739 頁的『[fte:filemove Ant 作業](#)』或第 733 頁的『[fte:call Ant 作業](#)』作業的 `idproperty` 屬性所設定的內容來指定此 ID。

範例

以下範例顯示如何使用 *fte:ignoreoutcome* 作業，來釋放配置為追蹤之前第 735 頁的『[fte:filecopy Ant 作業](#)』作業輸出的資源。

```
<!-- 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}"/>
```

相關參考

第 211 頁的『[搭配使用 Apache Ant 與 MFT](#)』

您可以利用 Managed File Transfer 提供的作業，將檔案傳送功能整合到 Apache Ant 工具中。

第 731 頁的『[MFT 提供的 Ant 作業](#)』

Managed File Transfer 提供了許多 Ant 作業，您可用於存取檔案傳送功能。

fte:ping Ant 作業

此 IBM MQ Managed File Transfer Ant 作業可對代理程式執行連線測試來導出回應，以判斷代理程式是否能夠處理傳送。

屬性

agent

必要項目。指定代理程式，以接受提交的 **fte:ping** 要求。該值的格式為：*agentname@qmgrname*，其中 *agentname* 是代理程式的名稱，*qmgrname* 是此代理程式直接連接的佇列管理程式的名稱。

cmdqm

選用項目。要向其提交要求的指令佇列管理程式。請使用下列格式指定此資訊：

qmgrname@host@port@channel，其中：

- *qmgrname* 為佇列管理程式的名稱
- *host* 為執行佇列管理程式的系統的選用主機名稱
- *port* 為佇列管理程式接聽的選用埠號
- *channel* 為要使用的選用 SVRCONN 通道

如果省略指令佇列管理程式的 *host*、*port* 或 *channel* 資訊，則會使用 `command.properties` 檔案中所指定的連線資訊。如需相關資訊，請參閱第 460 頁的『[MFT command.properties 檔](#)』。

您可以使用 `com.ibm.wmqfte.propertySet` 內容，指定所要使用的 `command.properties` 檔案。如需相關資訊，請參閱 `com.ibm.wmqfte.propertySet`。

如果您不使用 `cmdqm` 屬性，此作業會預設為使用 `com.ibm.wmqfte.ant.commandQueueManager` 內容（如果已設定此內容）。如果未設定 `com.ibm.wmqfte.ant.commandQueueManager` 內容，將會嘗試連線到 `command.properties` 檔案中所定義的預設佇列管理程式。

com.ibm.wmqfte.ant.commandQueueManager 內容的格式與 cmdqm 屬性相同，即 qmgrname@host@port@channel。

rcproperty

必要項目。指定內容，以儲存 **ping** 作業的回覆碼。

timeout

選用項目。作業等待代理程式回應的最長時間量（以秒為單位）。最短逾時是零秒，不過，也可指定 -1 逾時，讓指令永久等待代理程式回應。如果未指定 timeout 的值，預設等待代理程式回應的最長時間為 5 秒。

範例

此範例將 **fte:ping** 要求傳送至 qm1 所管理的 agent1。此 **fte:ping** 要求等待代理程式回應的時間為 15 秒。**fte:ping** 要求的結果儲存在稱為 ping.rc 的內容中。

```
<fte:ping agent="agent1@qm1" rcproperty="ping.rc" timeout="15"/>
```

回覆碼

0

指令已順利完成。

2

指令逾時。

相關參考

[第 211 頁的『搭配使用 Apache Ant 與 MFT』](#)

您可以利用 Managed File Transfer 提供的作業，將檔案傳送功能整合到 Apache Ant 工具中。

[第 731 頁的『MFT 提供的 Ant 作業』](#)

Managed File Transfer 提供了許多 Ant 作業，您可用於存取檔案傳送功能。

fte:uuid Ant 作業

產生虛擬隨機唯一 ID，並將它指派到給定的內容。例如，您可以使用這個 ID 來產生其他檔案傳送作業的工作名稱。

屬性

長度

必要項目。要產生的 UUID 的數值長度。此長度值不包括 **prefix** 參數指定的任何字首的長度。

內容

必要項目。所產生的 UUID 要指派至其中的內容名稱。

字首

選用項目。要新增至所產生的 UUID 的字首。此字首不算是 **length** 參數指定的 UUID 長度的一部分。

範例

這個範例定義的 UUID 是以字母 ABC 開頭，後面跟著 16 個虛擬隨機十六進位字元。會將 UUID 指派給名為 uuid.property 的內容。

```
<fte:uuid length="16" property="uuid.property" prefix="ABC"/>
```

相關參考



[第 211 頁的『搭配使用 Apache Ant 與 MFT』](#)

您可以利用 Managed File Transfer 提供的作業，將檔案傳送功能整合到 Apache Ant 工具中。

[第 731 頁的『MFT 提供的 Ant 作業』](#)

Managed File Transfer 提供了許多 Ant 作業，您可用於存取檔案傳送功能。

fte:filespec Ant 巢狀元素

fte:filespec 參數在其他作業中用作巢狀元素。使用 **fte:filespec** 說明一個以上來源檔案、目錄  或資料集與目的地之間的對映。表示要移動或複製的一組檔案、目錄  或是資料集時，通常會使用此元素。

由下列作業進行巢狀處理：

- [fte:filecopy](#) 作業
- [fte:filemove](#) 作業

來源規格屬性

您必須指定 `srcfilespec` 或 `srcqueue` 其中一個。

srcfilespec

指定檔案作業的來源。此屬性的值可以包含萬用字元。

srcqueue

指定傳送的來源為佇列。傳送會移動此屬性所指定佇列上儲存的訊息中的資料。如果 **fte:filespec** 作業是以巢狀方式置於 **fte:filecopy** 作業中，則您不能指定此屬性。

如果來源代理程式為通訊協定橋接器代理程式，則不支援 `srcqueue` 屬性。

目的地規格屬性

您必須指定 `dstdir`、`dstds`、`dstfilespace`、`dstfile`、`dstqueue` 或 `dstpds` 的其中一個。

dstdir

將目錄指定為檔案作業的目的地。

dstds

將資料集指定為檔案作業的目的地。

唯有當目的地代理程式是在 z/OS 平台上執行時，才支援此屬性。

dstfile

將檔案指定為檔案作業的目的地。

dstfilespace

將檔案空間指定為檔案作業的目的地。

僅當目的地代理程式是具有 Web 閘道檔案空間存取權的 IBM MQ 8.0 Web 代理程式時，此屬性才適用。

dstpds

將分割的資料集指定為檔案作業的目的地。

唯有當目的地代理程式是在 z/OS 平台上執行時，才支援此屬性。

dstqueue

將佇列指定為檔案轉為訊息作業的目的地。您可以選擇性地使用 `QUEUE@QUEUEMANAGER` 格式，在此規格中包含佇列管理程式名稱。如果未指定佇列管理程式名稱，而且未將 `enableClusterQueueInputOutput` 代理程式內容設為 `true`，則會使用目的地代理程式佇列管理程式。如果將 `enableClusterQueueInputOutput` 內容設為 `true`，則目的地代理程式將使用標準 IBM MQ 程序來判定放置佇列的位置。您必須指定佇列管理程式上存在的有效佇列名稱。

如果您指定 `dstqueue` 屬性，則不能指定 `srcqueue` 屬性，因為這些屬性互斥。

如果目的地代理程式為通訊協定橋接器代理程式，則不支援 `dstqueue` 屬性。

來源選項屬性

srcencoding

選用項目。要傳送的檔案所使用的字集編碼。

只有在 `conversion` 屬性設為值 `text` 時，才能指定這個屬性

如果您沒有指定 `srcencoding` 屬性，則會將來源系統的字集用於文字傳送。

srceol

選用項目。正在傳送的檔案所使用的行尾定界字元。有效值如下所示：

- CRLF - 將後接換行字元的歸位字元用作行尾定界字元。Windows 系統一般使用此慣例。
- LF - 將換行字元用作行尾定界字元。UNIX 系統一般使用此慣例。

僅當 `conversion` 屬性設定為 `text` 值時，您才可以指定此屬性。如果您沒有指定 `srceol` 屬性，文字傳送會根據來源代理程式的作業系統自動決定正確的值。

srckeeptrailingspaces

選用項目。採用文字模式進行傳送時，決定是否保留來源記錄（從固定長度格式的資料集讀取）中的尾端空格。有效值如下所示：

- `true` - 保留尾端空格。
- `false` - 移除尾端空格。

如果您沒有指定 `srckeeptrailingspaces` 屬性，則會指定預設值 `false`。

只有在同時指定 `srcfilespec` 屬性並將 `conversion` 屬性設為值 `text` 時，才能指定這個屬性。

srcmsgdelimbytes

選用項目。指定在將多則訊息附加到二進位檔時，要作為定界字元插入的一個以上位元組值。每一個值必須指定為兩個十六進位數字，範圍介於 00 - FF 之間，並以 `x` 作為字首。若有多個位元組，必須以逗點區隔。例如，`srcmsgdelimbytes="x08,xA4"`。僅當您同時指定 `srcqueue` 屬性時，才可以指定 `srcmsgdelimbytes` 屬性。如果您已為 `conversion` 屬性指定 `text` 值，則不能指定 `srcmsgdelimbytes` 屬性。

srcmsgdelimtext

選用項目。指定在將多則訊息附加到文字檔時，要作為定界字元插入的文字序列。您可以在定界字元中針對字串文字併入 Java ESC 序列。例如，`srcmsgdelimtext="\u007d\n"`。來源代理程式會在每則訊息的後面插入文字定界字元。使用傳送的來源編碼將文字定界字元編碼為二進位格式。每則訊息都以二進位格式讀取，以二進位格式將已編碼的定界字元附加到訊息，並且以二進位格式將結果傳送至目的地代理程式。如果來源代理程式字碼頁包括移入及移出狀態，則該代理程式會假設每則訊息的訊息結尾都為移出狀態。在目的地代理程式上，會以與檔案相同的方式，將二進位資料轉換為檔案文字傳送。如果您也已指定 `srcqueue` 屬性，並已為 `conversion` 屬性指定 `text` 值，則只能指定 `srcmsgdelimtext` 屬性。

srcmsgdelimposition

選用項目。指定文字或二進位定界字元的插入位置。有效值如下所示：

- `prefix` - 將定界字元插入到目的地檔案每則訊息中資料的前面。
- `postfix` - 將定界字元插入到目的地檔案每則訊息中資料的後面。

僅當您也指定 `srcmsgdelimbytes` 或 `srcmsgdelimtext` 屬性其中一個時，才可以指定 `srcmsgdelimposition` 屬性。

srcmsggroups

選用項目。指定訊息依 IBM MQ 群組 ID 進行分組。第一個完整群組會寫入目的地檔案。如果未指定此屬性，則會將來源佇列上的所有訊息寫入目的地檔案。僅當您同時指定 `srcqueue` 屬性時，才可以指定 `srcmsggroups` 屬性。

srcqueuetimeout

選用項目。指定等待下列其中一個條件成立的時間（以秒為單位）：

- 新訊息寫入佇列。
- 如果已指定 `srcmsggroups` 屬性，則條件為完整群組寫入佇列。

如果在 `srcqueuetimeout` 值指定的時間內，這兩個條件都不符合，來源代理程式會停止從佇列進行讀取，並完成傳送。如果未指定 `srcqueuetimeout` 屬性，則在以下情況下來源代理程式會立即停止從來源佇列進行讀取：來源佇列為空白或者已指定 `srcmsggroups` 屬性，佇列上沒有完整群組。僅當您同時指定 `srcqueue` 屬性時，才可以指定 `srcqueuetimeout` 屬性。

如需設定 `srcqueuetimeout` 值的相關資訊，請參閱第 574 頁的『指定「訊息轉為檔案」傳送等待時間的指引』。

z/OS `srcrcdelimbytes`

選用項目。指定在將記錄導向來源檔案中的多筆記錄附加到二進位檔時，要作為定界字元插入的一個以上位元組值。您必須將每一個值指定為兩個十六進位數字，範圍介於 00-FF 之間，並以 x 作為字首。若有多個位元組，必須以逗點區隔。例如：

```
srcrcdelimbytes="x08,xA4"
```

唯有當傳送來源檔案為記錄導向（例如 z/OS 資料集），目的地檔案為一般的非記錄導向檔案時，您可以指定 `srcrcdelimbytes` 屬性。如果您也已為 `conversion` 屬性指定 `text` 值，則不能指定 `srcrcdelimbytes` 屬性。

`srcrcdelimpos`

選用項目。指定二進位定界字元的插入位置。有效值如下所示：

- `prefix` - 將定界字元插入到目的地檔案每筆來源記錄導向檔案記錄中資料的前面。
- `postfix` - 將定界字元插入到目的地檔案每筆來源記錄導向檔案記錄中資料的後面。

僅當您同時指定 `srcrcdelimbytes` 屬性時，才可以指定 `srcrcdelimpos` 屬性。

目的地選項屬性

`dstencoding`

選用項目。要用於已傳送檔案的字集編碼。

只有在 `conversion` 屬性設為值 `text` 時，才能指定這個屬性

如果未指定 `dstencoding` 屬性，則會將目的地系統的字集用於文字傳送。

`dsteol`

選用項目。要用於已傳送檔案的行尾定界字元。有效值如下所示：

- `CRLF` - 將後接換行字元的歸位字元用作行尾定界字元。Windows 系統一般使用此慣例。
- `LF` - 將換行字元用作行尾定界字元。UNIX 系統一般使用此慣例。

只有在 `conversion` 屬性設為值 `text` 時，才能指定這個屬性

如果您沒有指定 `dsteol` 屬性，文字傳送會根據目的地代理程式的作業系統自動決定正確的值。

`dstmsgdelimbytes`

選用項目。指定將一個二進位檔分割為多則訊息時要使用的十六進位定界字元。所有訊息都具有相同的 IBM MQ 群組 ID；群組中的最後一則訊息已設定 IBM MQ `LAST_MSG_IN_GROUP` 旗標。將十六進位位元組指定為定界字元所採用的格式為 `xNN`，其中 N 為範圍介於 0-9 或 a-f 的字元。透過指定以逗點區隔的十六進位位元組清單（例如：`x3e,x20,x20,xbf`），您可以將十六進位位元組的序列指定為定界字元。

僅當您也已指定 `dstqueue` 屬性並且傳送採用二進位模式時，才可以指定 `dstmsgdelimbytes` 屬性。您只能指定 `dstmsgsize`、`dstmsgdelimbytes` 及 `dstmsgdelimpattern` 屬性的其中一個。

`dstmsgdelimpattern`

選用項目。指定將一個文字檔分割為多則訊息時要使用的 Java 正規表示式。所有訊息都具有相同的 IBM MQ 群組 ID；群組中的最後一則訊息已設定 IBM MQ `LAST_MSG_IN_GROUP` 旗標。將正規表示式

指定為定界字元的格式是用括弧 (*regular_expression*) 括住的正規表示式，或用雙引號 "*regular_expression*" 括住的正規表示式。如需相關資訊，請參閱 [第 554 頁的『MFT 所使用的正規表示式』](#)。

依預設，目的地代理程式限制正規表示式可以比對的字串長度為五個字元。您可以使用 **maxDelimiterMatchLength** 代理程式內容變更此行為。如需相關資訊，請參閱 [第 464 頁的『進階代理程式內容』](#)。

僅當您也已指定 `dstqueue` 屬性並且傳送採用文字模式時，才可以指定 `dstmsgdelimpattern` 屬性。您只能指定 `dstmsgsize`、`dstmsgdelimbytes` 及 `dstmsgdelimpattern` 屬性的其中一個。

dstmsgdelimposition

選用項目。指定文字或二進位定界字元預期所在的位置。有效值如下所示：

- `prefix` - 定界字元預期位於每一行的開始。
- `postfix` - 定界字元預期位於每一行的結尾。

僅當您也指定 `dstmsgdelimpattern` 屬性時，才可以指定 `dstmsgdelimposition` 屬性。

dstmsgincludedelim

選用項目。指定是否要在訊息中包含將檔案分割為多則訊息所使用的定界字元。如果已指定 `dstmsgincludedelim` 屬性，則會在訊息（包含定界字元之前的檔案資料）的結尾包含定界字元。依預設，訊息不包括定界字元。僅當您也已指定 `dstmsgdelimpattern` 及 `dstmsgdelimbytes` 屬性其中一個時，才可以指定 `dstmsgincludedelim` 屬性。

dstmsgpersist

選用項目。指定寫入目的地佇列的訊息是否持續保存。有效值如下所示：

- `true` - 將持續訊息寫入目的地佇列。這是預設值。
- `false` - 將非持續性訊息寫入目的地佇列。
- `qdef` - 持續性值取自目的地佇列的 `DefPersistence` 屬性。

僅當您也已指定 `dstqueue` 屬性時，您才可以指定此屬性。

dstmsgprops

選用項目。指定由傳送寫入目的地佇列的第一則訊息是否已設定 IBM MQ 訊息內容。可能的值為：

- `true` - 在由傳送建立的第一則訊息上設定訊息內容。
- `false` - 不在由傳送建立的第一則訊息上設定訊息內容。這是預設值。

如需相關資訊，請參閱 [第 569 頁的『寫入目的地佇列的訊息上由 MFT 設定的 MQ 訊息內容』](#)。

僅當您也已指定 `dstqueue` 屬性時，您才可以指定此屬性。

dstmsgsize

選用項目。指定是否要將檔案分割為多則固定長度的訊息。所有訊息都具有相同的 IBM MQ 群組 ID；群組中的最後一則訊息已設定 IBM MQ `LAST_MSG_IN_GROUP` 旗標。訊息的大小由 `dstmsgsize` 值指定。`dstmsgsize` 的格式為 *lengthunits*，其中 *length* 為正整數值，*units* 為下列其中一個值：

- B - 位元組。容許的最小值為目的地訊息字碼頁的每個字元佔用的位元組數最大值的兩倍。
- K - KB。這相等於 1024 個位元組。
- M - MB。這相等於 1024KB。

如果檔案以文字模式傳送，且為雙位元組字集或多位元組字集，則會依指定的訊息大小，在最近字元界限上將檔案分割成多個訊息。

僅當您也已指定 `dstqueue` 屬性時，才可以指定 `dstmsgsize` 屬性。您只能指定 `dstmsgsize`、`dstmsgdelimbytes` 及 `dstmsgdelimpattern` 屬性的其中一個。

dstunsupportedcodepage

選用項目。指定在 `dstqueue` 屬性所指定的目的地佇列管理程式，不支援將檔案資料作為文字傳送傳送至佇列時所使用的字碼頁情況下，要採取的動作。此屬性的有效值如下所示：

- `binary` - 繼續進行傳送，但不將字碼頁轉換套用至正在傳送的資料。指定此值相等於不將 `conversion` 屬性設定為 `text`。

- `fail` - 不繼續進行傳送作業。該檔案會記錄為傳送失敗。這是預設值。

如果您也已指定 `dstqueue` 屬性，並已為 `conversion` 屬性指定 `text` 值，則只能指定 `dstunsupportedcodepage` 屬性。

dsttruncaterecords

選用項目。指定要將長度超過 `LRECL` 資料集屬性的目的地記錄截斷。如果設為 `true`，則會將記錄截斷。如果設為 `false`，則會將記錄換行。預設值為 `false`。此參數僅適用於目的地為資料集的文字模式傳送。

其他屬性

checksum

選用項目。決定對已傳送檔案進行總和檢查所使用的演算法。

- `MD5` - 使用 MD5 雜湊演算法。
- `NONE` - 不使用總和檢查演算法。

如果您沒有指定 `checksum` 屬性，則會使用預設值 `MD5`。

conversion

選用項目。指定在傳送檔案時，要套用至檔案的轉換類型。可能的值為：

- `binary` - 不套用轉換。
- `text` - 在來源及目的地系統間套用字碼頁轉換。同時套用行定界字元的轉換。`srcencoding`、`dstencoding`、`srceol` 及 `dsteol` 屬性會影響所套用的轉換。

如果您沒有指定 `conversion` 屬性，則會指定預設值 `binary`。

overwrite

選用項目。決定作業是否可以改寫現有的目的地檔案 `z/OS` 或資料集。如果您指定 `true` 值，則會改寫任何現有的目的地檔案 `z/OS` 或資料集。如果您指定 `false` 值，則目的地中存在的重複檔案 `z/OS` 或資料集會導致作業失敗。如果未指定 `overwrite` 屬性，則會指定預設值 `false`。

recurse

選用項目。決定檔案傳送是否遞迴到子目錄。如果您指定 `true` 值，則傳送會遞迴到子目錄。如果您指定 `false` 值，則傳送不會遞迴到子目錄。如果未指定 `recurse` 屬性，則會指定預設值 `false`。

範例

此範例指定 `fte: filespec`，其來源檔為 `file1.bin`，目的地檔為 `file2.bin`。

```
<fte:filespec srcfilespec="/home/fteuser/file1.bin" dstfile="/home/fteuser/file2.bin"/>
```

相關參考

第 211 頁的『[搭配使用 Apache Ant 與 MFT](#)』

您可以利用 Managed File Transfer 提供的作業，將檔案傳送功能整合到 Apache Ant 工具中。

第 731 頁的『[MFT 提供的 Ant 作業](#)』

Managed File Transfer 提供了許多 Ant 作業，您可用於存取檔案傳送功能。

fte:metadata Ant nested element

`meta` 資料用於容納其他使用者定義的資訊及檔案傳送作業。

請參閱第 752 頁的『[MFT 使用者結束程式的 meta 資料](#)』，以取得有關 Managed File Transfer 如何使用 `meta` 資料的相關資訊。

由下列作業進行巢狀處理：

- [fte:filecopy](#) 作業

- [fte:filemove](#) 作業
- [fte:call](#) 作業

指定為巢狀元素的參數

fte:entry

您必須在 `fte:metadata` 巢狀元素內至少指定一個項目。您可以選擇指定多個項目。項目會將鍵名稱與值關聯。索引鍵在 `fte:metadata` 區塊中必須是唯一的

項目屬性

name

必要項目。屬於此項目的鍵名稱。此名稱在 `fte:metadata` 元素內巢狀的所有 **entry** 參數中必須是唯一的。

value

必要項目。指派給此項目的值。

範例

此範例顯示包含兩個項目的 `fte:metadata` 定義。

```
<fte:metadata>
  <fte:entry name="org.foo.partColor" value="red"/>
  <fte:entry name="org.foo.partSize" value="medium"/>
</fte:metadata>
```

相關參考

第 211 頁的『[搭配使用 Apache Ant 與 MFT](#)』

您可以利用 Managed File Transfer 提供的作業，將檔案傳送功能整合到 Apache Ant 工具中。

第 731 頁的『[MFT 提供的 Ant 作業](#)』

Managed File Transfer 提供了許多 Ant 作業，您可用於存取檔案傳送功能。

程式呼叫巢狀元素

您可以使用下列五個巢狀元素的其中一個來啟動程式：`fte:presrc`、`fte:predst`、`fte:postdst`、`fte:postsrc` 及 `fte:command`。這些巢狀元素會指示代理程式在其處理過序中呼叫外部程式。您必須先確定指令位於要執行此指令之代理程式的 `agent.properties` 檔案中 `commandPath` 內容所指定的位置中，才可以啟動程式。

雖然每一個程式呼叫元素的名稱都不同，但它們共用相同的屬性集及相同的巢狀元素集。程式可以由 **fte:filecopy**、**fte:filemove** 和 **fte:command** Ant 作業啟動。

您無法從 Connect:Direct 橋接器代理程式呼叫程式。

可呼叫程式的 Ant 作業：

- [fte:filecopy](#) 作業使用 `fte:predst`、`fte:postdst`、`fte:presrc` 及 `fte:postsrc` 巢狀元素來內嵌程式呼叫參數。
- [fte:filemove](#) 作業使用 `fte:predst`、`fte:postdst`、`fte:presrc` 及 `fte:postsrc` 巢狀元素來內嵌程式呼叫參數。
- [fte:call](#) 作業使用 `fte:command` 巢狀元素來內嵌程式呼叫參數。

屬性

指令

必要項目。指定要呼叫的程式。為了讓代理程式能夠執行指令，指令必須位於代理程式的 `agent.properties` 檔案中的 `commandPath` 內容所指定的位置中。如需相關資訊，請參閱第 306 頁的『[commandPath MFT 內容](#)』。`command` 屬性中指定的任何路徑資訊，都會被視為相對於

commandPath 內容所指定的位置。若 type 是 executable，則預期為可執行程式，否則預期為適用於呼叫類型的 Script。

retrycount

選用項目。在程式未傳回成功回覆碼的情況下，重試呼叫程式的次數。command 屬性所指定的程式最多只會被呼叫這個次數。指派給此屬性的值必須是非負數。如果未指定 retrycount 屬性，則會使用預設值零。

retrywait

選用項目。重新嘗試呼叫程式之前等待的時間（以秒為單位）。如果 command 屬性所指定的程式未傳回成功回覆碼，而 retrycount 屬性指定了非零值，此參數即會決定重試之間的等待時間。指派給此屬性的值必須是非負數。如果未指定 retrywait 屬性，則會使用預設值零。

successrc

選用項目。此屬性值用來判定程式呼叫順利執行的時機。指令的程序回覆碼會使用此表示式進行評估。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 AND. 每一個表示式可以是下列其中一種表示式類型：

- 一個數字，表示程序回覆碼與此數之間的「相等」測試。
- 一個以 ">" 字元為開頭的數字，表示此數字與程序回覆碼之間的「大於」測試。
- 以 "<" 字元為字首的數字，表示數字與處理程序回覆碼之間的小於測試。
- 以 "!" 為字首的數字 字元，指出數字與處理程序回覆碼之間的不等於測試。

例如：>2&<7&!5|0|14 將下列回覆碼解譯為成功回覆碼：0、3、4、6、14。其餘全都解譯為不成功回覆碼。如果未指定 successrc 屬性，則會使用預設值零。這表示只有在回覆碼為零時，才會判定指令已順利執行。

類型

選用項目。此屬性的值會指定所呼叫的程式類型。請指定下列其中一個選項：

executable

作業會呼叫可執行程式。可以使用 arg 巢狀元素指定其他引數。預期可從 commandPath 上存取程式，並於適當時設定執行權限。只要指定 Shell 程式，就可以呼叫 UNIX Script (例如，Shell Script 檔的第一行是: #!/bin/sh)。寫入 stderr 或 stdout 的指令輸出會傳送至呼叫的 Managed File Transfer 日誌。不過，資料輸出量會受到代理程式配置的限制。預設資料量是 10K 位元組，但您可以使用代理程式內容 maxCommandOutput 來置換此預設值。

antscript

作業可使用 **fteAnt** 指令來執行指定的 Ant Script。內容可使用 property 巢狀元素來指定。Ant 目標可使用 target 巢狀元素來指定。預期可在 commandPath 上存取 Ant Script。寫入至 stderr 或 stdout 的 Ant 輸出，會傳送至 Managed File Transfer 的呼叫日誌。不過，資料輸出量會受到代理程式配置的限制。預設資料量是 10K 位元組，但您可以使用代理程式內容 maxCommandOutput 來置換此預設值。

jcl

僅 z/OS 支援 jcl 值，此值執行指定的 z/OS JCL Script。JCL 會以工作形式提交，且必須有工作卡存在。工作順利提交後，JCL 指令輸出即會寫入 Managed File Transfer 日誌，其中包含下列文字：JOB *job_name(job_id)*，其中：

- *job_name* 是 JCL 中的工作卡所識別的工作名稱。
- *job_id* 是 z/OS 系統產生的工作 ID。

如果工作無法順利提交，JCL Script 指令即會失敗，並將一則訊息寫入日誌中以指出作業失敗的原因（例如，沒有工作卡存在）。若想要瞭解工作是否已執行或順利完成，請使用 SDSF 之類的系統服務。Managed File Transfer 只會提交工作，因此不提供資訊；然後，系統會決定要在何時執行工作以及如何顯示工作輸出。因為 JCL Script 是以批次工作形式提交，所以不建議您為 presrc 或 predst 巢狀元素指定 jcl，因為您只知道工作已順利提交，但不知道它是否在傳送開始之前是否已順利完成執行。沒有 jcl 類型的有效巢狀元素。

下列範例說明 JCL 工作：

```
//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
```

指定為巢狀元素的參數

fte:arg

只有在 `type` 屬性的值是 `executable` 時，才有效用。請使用巢狀 `fte:arg` 元素，為在程式呼叫過程中所呼叫的程式指定引數。程式引數是使用 `fte:arg` 元素所指定的值，依據 `fte:arg` 元素出現的順序而建置的。您可以選擇將零個以上 `fte:arg` 元素指定為程式呼叫的巢狀元素。

fte:property

只有在 `type` 屬性的值是 `antscript` 時，才有效用。請使用巢狀 `fte:property` 元素的 `name` 及 `value` 屬性，將「名稱/值配對」傳入 Ant Script 中。您可以選擇將零個以上 `fte:property` 元素指定為程式呼叫的巢狀元素。

fte:target

只有在 `type` 屬性的值是 `antscript` 時，才有效用。請在 Ant Script 中指定要呼叫的目標。您可以選擇將零個以上 `fte:target` 元素指定為程式呼叫的巢狀元素。

引數屬性

value

必要項目。傳遞給所呼叫程式的引數值。

內容屬性

名

必要項目。傳遞給 Ant Script 的內容名稱。

value

必要項目。要傳遞給 Ant Script 的內容名稱相關聯的值。

範例

此範例顯示在 `fte:filecopy` 作業過程中指定 `fte:postsrc` 程式呼叫。程式呼叫適用於稱為 `post.sh` 的程式，並提供單一引數 `/home/fteuser2/file.bin`。

```
<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>
```

此範例顯示在 `fte:call` 作業中指定的 `fte:command` 程式呼叫。這是執行檔 `command.sh` 的程式呼叫，其中未傳入任何指令引數。如果 `command.sh` 未傳回成功回覆碼 1，將在 30 秒後重新嘗試指令。

```
<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>
```

此範例顯示在 `fte:call` 作業中指定的 `fte:command` 程式呼叫。這是針對 Ant Script `script.xml` 中的複製及壓縮目標而執行的程式呼叫，其中傳入了兩個內容。

```
<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>
```

相關概念

第 209 頁的『指定要使用 MFT 執行的程式』

您可以在執行 Managed File Transfer Agent 的系統上執行程式。在檔案傳送要求過程中，您可以指定程式在傳送開始前或結束後執行。此外，您也可以提交受管理的呼叫要求，以啟動不在檔案傳送要求過程中的程式。

相關參考

第 211 頁的『搭配使用 Apache Ant 與 MFT』

您可以利用 Managed File Transfer 提供的作業，將檔案傳送功能整合到 Apache Ant 工具中。

第 731 頁的『MFT 提供的 Ant 作業』

Managed File Transfer 提供了許多 Ant 作業，您可用於存取檔案傳送功能。

使用 MFT 使用者結束程式進行自訂

MFT 使用者結束程式的 meta 資料

有三種不同類型的 meta 資料可提供給 Managed File Transfer 的使用者結束常式使用：環境、傳送及檔案 meta 資料。此 meta 資料可顯示為 Java 鍵值組的對映。

環境 meta 資料

環境 meta 資料可傳遞至所有使用者結束常式，並可說明呼叫使用者結束常式的代理程式執行時期環境。此 meta 資料是唯讀項目，任何使用者結束常式都無法加以更新。

鍵	說明
AGENT_CONFIGURATION_DIRECTORY_KEY	包含代理程式配置資訊的目錄名稱。
AGENT_PRODUCT_DIRECTORY_KEY	代理程式碼安裝所在的目錄名稱。
AGENT_VERSION_KEY	呼叫結束常式的代理程式執行時期的版本號碼。

在表格 1 中指定的索引鍵名稱及值名稱，皆為定義於 `EnvironmentMetaDataConstants` 介面中的常數。

傳送 meta 資料

傳送 meta 資料可傳遞至所有的使用者結束常式。此 meta 資料由系統提供的值及使用者提供的值組成。您對系統提供的值所做的任何變更都會被忽略。來源傳送開始使用者結束程式的起始使用者提供的值，基於您在定義傳送時所提供的值。來源代理程式可在處理來源傳送開始使用者結束程式的過程中，變更使用者提供的值。此使用者結束程式是在整個檔案傳送開始之前進行呼叫。後續呼叫其他與該傳送相關的結束常式時，即會使用這些變更。傳送 meta 資料會套用至整個傳送。

雖然所有的使用者結束程式皆可讀取傳送 meta 資料中的值，但只有來源傳送開始使用者結束程式可變更傳送 meta 資料。

您無法使用傳送 meta 資料在不同的檔案傳送之間傳輸資訊。

系統提供的傳送 meta 資料詳述於表格 2 中：

鍵	說明
DESTINATION_AGENT_KEY	作為傳送目的地的代理程式名稱。
JOB_NAME_KEY	與傳送要求相關聯的工作名稱
MQMD_USER_KEY	訊息中用來提交傳送要求的 MQMD 使用者欄位
ORIGINATING_HOST_KEY	在傳送要求中指定為原始主機名稱的主機名稱
ORIGINATING_USER_KEY	在傳送要求中指定為原始使用者 ID 的使用者名稱
SOURCE_AGENT_KEY	作為傳送來源的代理程式名稱
TRANSFER_ID_KEY	傳送的 ID

在表格 2 中指定的索引鍵名稱及值名稱，皆為定義於 TransferMetaDataConstants 介面中的常數。

檔案 meta 資料

檔案 meta 資料可在檔案指定過程中，傳遞至來源傳送開始結束程式。來源與目的地檔案分別有其檔案 meta 資料。

您無法使用檔案 meta 資料在不同的檔案傳送之間傳輸資訊。

鍵	允許值	說明
CONVERT_LINE_SEPARATORS		此索引鍵值可用於文字傳送，指出來源資料中的 CRLF（換行-換行）或 LF（換行）行分隔字元順序，是否要轉換為目的地的行分隔字元順序。
DELIMITER_KEY		此索引鍵值可定義傳送記錄導向的資料到一般檔案時，用以分隔記錄資料的定界字元。 此外也可用於「訊息轉為檔案」及「檔案轉為訊息」的傳送。
DELIMITER_POSITION_KEY	DELIMITER_POSITION_PREFIX_VALUE DELIMITER_POSITION_POSTFIX_VALUE	與 DELIMITER_KEY 併用時可以定義定界字元的位置；只可在開頭或結尾。
DELIMITER_TYPE_KEY	DELIMITER_TYPE_BINARY_VALUE DELIMITER_TYPE_TEXT_VALUE DELIMITER_TYPE_SIZE_VALUE	與 DELIMITER_KEY 並用時可以定義定界字元的類型。
DESTINATION_EXIST_KEY	DESTINATION_EXIST_KEY_ERROR_VALUE DESTINATION_EXIST_KEY_OVERWRITE_VALUE	決定目的地檔案存在時的檔案傳送行為。
FILE_ALIAS_KEY		此索引鍵值可定義所要傳送之檔案的別名。
FILE_CHECKSUM_METHOD_KEY	FILE_CHECKSUM_METHOD_NONE_VALUE FILE_CHECKSUM_METHOD_MD5_VALUE	決定傳送檔案時所使用的總和檢查方法。
FILE_CONVERSION_KEY	FILE_CONVERSION_TEXT_VALUE FILE_CONVERSION_BINARY_VALUE	決定要對檔案內容套用的轉換類型。
FILE_ENCODING_KEY		決定文字檔所使用的編碼。
FILE_END_OF_LINE_KEY	FILE_END_OF_LINE_LF_VALUE FILE_END_OF_LINE_CRLF_VALUE	決定用來表示行尾的字元順序： <LF> 或 <CR><LF>。

表 73: 檔案 meta 資料 (繼續)		
鍵	允許值	說明
FILE_SPACE_ALIAS		決定一個檔案在檔案空間中的別名。 註: 只有在 FILE_TYPE_KEY 是 FILE_TYPE_FILE_SPACE_VALUE 時, 才可以使用此 meta 資料。
FILE_SPACE_NAME		決定檔案空間的名稱。 註: 只有在 FILE_TYPE_KEY 是 FILE_TYPE_FILE_SPACE_VALUE 時, 才可以使用此 meta 資料。
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	決定目的地檔案、佇列或檔案空間的規格。
GROUP_ID_KEY		此索引鍵值可用於「訊息轉為檔案」的傳送, 指定讀取來源佇列的訊息群組。僅當 USE_GROUPS_KEY 的值為 USE_GROUPS_TRUE_VALUE 時, 此屬性才有效。
INCLUDE_DELIMITER_IN_MESSAGE_KEY	INCLUDE_DELIMITER_IN_MESSAGE_TRUE_VALUE INCLUDE_DELIMITER_IN_MESSAGE_FALSE_VALUE	此索引鍵值可用於「檔案轉為訊息」的傳送, 指定是否要在每則訊息結尾加入定界字元, 以將檔案分割為多則訊息。僅當 DELIMITER_TYPE_KEY 的值為 DELIMITER_TYPE_BINARY_VALUE DELIMITER_TYPE_TEXT_VALUE 時, 此屬性才有效。
INSERT_RECORD_LINE_SEPARATOR_KEY		此索引鍵值可用於來自記錄導向之檔案的文字傳送, 指定是否要在每筆記錄後的資料中插入行分隔字元。
KEEP_TRAILING_SPACES_KEY	KEEP_TRAILING_SPACES_TRUE_VALUE KEEP_TRAILING_SPACES_FALSE_VALUE	此索引鍵值可指定在從固定長度格式的資料集讀取記錄時, 是否要移除尾端空格。
NEW_RECORD_ON_LINE_SEPARATOR_KEY		此索引鍵值可用於對記錄導向之檔案的文字傳送, 指定資料中的行分隔字元要加入記錄資料中, 或用於新建記錄 (而不予寫入)。
PERSISTENT_KEY	PERSISTENT_TRUE_VALUE PERSISTENT_FALSE_VALUE PERSISTENT_QDEF_VALUE	此索引鍵值可用於「檔案轉為訊息」的傳送, 指定是否持續保存訊息。
SET_MQ_PROPS_KEY	SET_MQ_PROPS_TRUE_VALUE SET_MQ_PROPS_FALSE_VALUE	此索引鍵值可用於檔案轉為訊息的傳送, 指定是否要對檔案的第一則訊息設定 IBM MQ 訊息內容, 以及當發生錯誤時, 是否要將所有訊息寫入佇列。
UNRECOGNISED_CODE_PAGE_KEY	UNRECOGNISED_CODE_PAGE_FAIL_VALUE UNRECOGNISED_CODE_PAGE_BINARY_VALUE	此索引鍵值可用於「檔案轉為訊息」的傳送, 指定當目的地佇列管理程式無法辨識資料的字碼頁時, 文字模式傳送失敗或執行轉換。
USE_GROUPS_KEY	USE_GROUPS_TRUE_VALUE USE_GROUPS_FALSE_VALUE	此索引鍵值可用於「訊息轉為檔案」的傳送, 指定是否只傳送來源佇列中的整個訊息群組。

表 73: 檔案 meta 資料 (繼續)		
鍵	允許值	說明
WAIT_TIME_KEY		<p>此索引鍵值可用於「訊息轉為檔案」的傳送，指定來源代理程式對於下列其中一個狀況的等待時間（以秒為單位）：</p> <ul style="list-style-type: none"> • 訊息出現在來源佇列中；佇列為空的或佇列變為空的； USE_GROUPS_KEY 的值為 FALSE。 • 整個群組出現在來源佇列； USE_GROUPS_KEY 的值為 TRUE。

在表格 3 中指定的索引鍵名稱及值名稱，皆為定義於 FileMetaDataConstants 介面中的常數。

MFT 資源監視器使用者結束程式

資源監視器使用者結束程式可讓您配置在滿足監視器的觸發條件時，可在關聯作業啟動之前執行的自訂程式碼。

建議直接從使用者結束程式碼呼叫新傳送。在某些情況下，檔案會因為使用者結束程式無法觸發代理程式重新啟動而出現多次傳送的狀況。

資源監視器使用者結束程式使用使用者結束程式的現有基礎架構。呼叫監視器使用者結束程式的時間點，是在觸發監視器之後、且監視器的作業執行對應作業之前。這讓使用者結束程式得以修改要執行的作業，並決定作業是否應該繼續。您可以更新監視器 meta 資料以修改監視器作業；監視器 meta 資料後續用於建立原始監視器時所建立的作業文件中的變數替代。或者，監視器結束程式也可取代或更新傳入作為參數的作業定義 XML 字串。監視器結束程式可為作業傳回 'proceed' 或 'cancel' 結果碼。如果傳回 'cancel'，在受監視資源符合觸發條件之前，作業不會啟動，監視器也不會重新啟動。如果資源未變更，觸發將不會啟動。如同其他使用者結束程式，您可以將監視器結束程式鏈結在一起。如果其中一個結束程式傳回 'cancel' 結果碼，整體結果即會被取消，作業也不會啟動。

- 環境 meta 資料的對映（與其他使用者結束程式相同）
- 包括不變的系統 meta 資料及多變的使用者 meta 資料的監視器 meta 資料對映。不變的系統 meta 資料如下所示：
 - FILENAME - 滿足觸發條件的檔案名稱
 - FILEPATH - 滿足觸發條件的檔案路徑
 - FILESIZE（以位元組為單位 - 此 meta 資料可能不存在） - 滿足觸發條件的檔案大小
 - LASTMODIFIEDDATE（本端） - 滿足觸發條件之檔案的前次變更日期。此日期以代理程式執行所在之時區的當地日期表示，並且格式化為 ISO 8601 日期。
 - LASTMODIFIEDTIME（本端） - 滿足觸發條件之檔案的前次變更的本端格式時間。此時間以代理程式執行所在之時區的當地時間表示，並且格式化為 ISO 8601 時間。
 - LASTMODIFIEDDATEUTC - 滿足觸發條件之檔案的前次變更的通用格式日期。此日期以轉換為世界標準時間時區的當地日期表示，並且格式化為 ISO 8601 日期。
 - LASTMODIFIEDTIMEUTC - 滿足觸發條件之檔案的前次變更的通用格式時間。此時間以轉換為世界標準時間時區的當地時間表示，並且格式化為 ISO 8601 時間。
 - AGENTNAME - 監視器代理程式名稱
- XML 字串，代表監視器觸發後所要執行的作業。

監視器結束程式會傳回下列資料：

- 指定是否要繼續執行（繼續或取消）的指示器
- 要在「滿足觸發條件」日誌訊息中插入的字串

在執行監視器結束程式碼後，最初傳入作為參數的監視器 meta 資料及作業定義 XML 字串也可能已更新。

代理程式內容 monitorExitClasses（位於 agent.properties 檔中）的值會指定要載入的監視器結束程式類別（每一個結束程式類別會以逗點區隔）。例如：

```
monitorExitClasses=testExits.TestExit1,testExits.testExit2
```

監視器使用者結束程式的介面是：

```
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);
}

```

在監視器 meta 資料中，IBM 保留值的常數如下所示：

```
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";
}

```

監視器使用者結束程式範例

此範例類別會實作 MonitorExit 介面。在此範例中，會在填入 LONDON 值（如果小時數是奇數）或 PARIS 值（如果小時數是偶數）的監視器 meta 資料 *REDIRECTEDAGENT* 中加入自訂替代變數。監視器結束程式碼設為一律傳回 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* 替代變數的監視器對應作業可能如下：

```

<?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>

```

Before this transfer is started, the value of the `<destinationAgent>` element's 代理人 attribute is replaced with either LONDON or PARIS.

監視器結束程式類別及作業定義 XML 中的替代變數必須以大寫指定。

相關概念

第 214 頁的『[利用使用者結束程式自訂 MFT](#)』

您可以使用專屬程式（即所謂的使用者結束常式）來自訂 Managed File Transfer 的特性。

第 752 頁的『[MFT 使用者結束程式的 meta 資料](#)』

有三種不同類型的 meta 資料可提供給 Managed File Transfer 的使用者結束常式使用：環境、傳送及檔案 meta 資料。此 meta 資料可顯示為 Java 鍵值組的對映。

第 760 頁的『[MFT 使用者結束程式的 Java 介面](#)』

本節的主題為您提供使用者結束常式的 Java 介面的相關參照資訊。

相關參考

第 759 頁的『[使用者結束程式的 MFT 代理程式內容](#)』

除了 `agent.properties` 檔案中的標準內容外，還具有數個專用於使用者結束程式常式的進階內容。依預設，並不包含這些內容，因此，如果要使用其中任何內容，您必須手動編輯 `agent.properties` 檔案。如果在該代理程式正在執行時對 `agent.properties` 檔案進行變更，請停止並重新啟動代理程式以取得變更。

使用者結束程式的 MFT 代理程式內容

除了 `agent.properties` 檔案中的標準內容外，還具有數個專用於使用者結束程式常式的進階內容。依預設，並不包含這些內容，因此，如果要使用其中任何內容，您必須手動編輯 `agent.properties` 檔案。如果在該代理程式正在執行時對 `agent.properties` 檔案進行變更，請停止並重新啟動代理程式以取得變更。

若為 IBM WebSphere MQ 7.5 或更新版本，可在某些代表檔案或目錄位置的 Managed File Transfer 內容中使用環境變數。這可讓執行產品的某部分時所使用的檔案或目錄，隨著環境變更（例如執行程序的使用者為何）而改變其所在位置。如需相關資訊，請參閱第 452 頁的『在 MFT 內容中使用環境變數』。

使用者結束常式內容

使用者結束常式會依照下表列出的順序來呼叫。如需 `agent.properties` 檔案的相關資訊，請參閱 [進階代理程式內容: 使用者結束常式](#)。

內容名稱	說明
<code>sourceTransferEndExitClasses</code>	指定以逗點區隔的類別清單，這些類別實作來源傳送結束使用者結束程式。
<code>sourceTransferStartExitClasses</code>	指定以逗點區隔的類別清單，這些類別實作來源傳送開始使用者結束程式。
<code>destinationTransferStartExitClasses</code>	指定以逗點區隔的類別清單，這些類別實作目的地傳送開始使用者結束程式常式。
<code>destinationTransferEndExitClasses</code>	指定以逗點區隔的類別清單，這些類別實作目的地傳送使用者結束程式常式。
<code>exitClassPath</code>	<p>指定平台專用的、字元定界的目錄清單，這些目錄充當使用者結束程式常式的類別路徑。</p> <p>在此類別路徑中的任何項目之前，會先搜尋代理程式結束程式目錄。</p> <p>如果您在 Windows 上使用此內容，請使用正斜線字元 (/) 作為路徑定界字元，而不是反斜線字元 (\)。例如：</p> <pre>exitClassPath=C:/mycomp/mqft/exits/encryptFileExit.jar; C:/mycomp/mqft/exits/fileFilter.jar.</pre> <p>若為 IBM WebSphere MQ 7.5 或更新版本，此內容值可包含環境變數。</p>
<code>exitNativeLibraryPath</code>	<p>指定平台專用的、字元定界的目錄清單，這些目錄充當使用者結束程式常式的原生程式庫路徑。</p> <p>若為 IBM WebSphere MQ 7.5 或更新版本，此內容值可包含環境變數。</p>
<code>monitorExitClasses</code>	指定以逗點區隔的類別清單，這些類別實作監視器結束常式。如需相關資訊，請參閱第 755 頁的『MFT 資源監視器使用者結束程式』。
<code>protocolBridgeCredentialExitClasses</code>	指定以逗點區隔的類別清單，這些類別實作通訊協定橋接器認證使用者結束常式。如需相關資訊，請參閱第 188 頁的『使用結束類別對映檔案伺服器的認證』。
<code>protocolBridgePropertiesExitClasses</code>	<p>指定以逗點區隔的類別清單，這些類別實作通訊協定橋接器伺服器內容使用者結束常式。</p> <p>如需相關資訊，請參閱第 183 頁的『ProtocolBridgePropertiesExit2: 查閱通訊協定檔案伺服器內容』。</p>
<code>IOExitClasses</code>	指定以逗點區隔的類別清單，這些類別實作 I/O 使用者結束常式。僅列出實作 IOExit 介面的類別，亦即，不列出實作其他 I/O 使用者結束程式介面的類別，例如 <code>IOExitResourcePath</code> 及 <code>IOExitChannel</code> 。如需相關資訊，請參閱第 218 頁的『使用 MFT 傳送 I/O 使用者結束程式』。

結束程式呼叫的順序

按下列順序呼叫來源及目的地結束程式：

1. `SourceTransferStartExit`
2. `DestinationTransferStartExit`
3. `DestinationTransferEndExit`

鏈結來源及目的地結束程式

如果您指定多個結束程式，請先呼叫清單中的第一個結束程式，然後呼叫第二個結束程式等等。第一個結束程式所做的任何變更會傳遞為後續呼叫的結束程式的輸入等等。比方說，例如有兩個「來源傳送啟動」結束程式，則第一個結束程式對傳送 meta 資料所做的任何變更會輸入至第二個結束程式。每一個結束程式會傳回自己的結果。如果給定類型的所有結束程式傳回 PROCEED 作為傳送結果碼，則整體結果是 PROCEED。如果一個以上的結束程式傳回 CANCEL_TRANSFER，則整體結果是 CANCEL_TRANSFER。結束程式傳回的所有結果碼及字串會輸出在傳送日誌中。

如果「來源傳送啟動」結束程式的整體結果是 PROCEED，則傳送會繼續使用結束程式所做的任何變更。如果整體結果是 CANCEL_TRANSFER，則會呼叫「來源傳送結束」結束程式，然後取消此傳送。傳送日誌中的完成狀態為「已取消」。

如果「目的地傳送啟動」結束程式的整體結果是 PROCEED，則傳送會繼續使用結束程式所做的任何變更。如果整體結果是 CANCEL_TRANSFER，則會呼叫「目的地傳送結束」結束程式，然後呼叫「來源傳送結束」結束程式。最後，會取消傳送。傳送日誌中的完成狀態為「已取消」。

如果來源或目的地結束程式需要按鏈結或執行順序將資訊傳遞至下列結束程式，則必須透過更新傳送 meta 資料來執行。傳送 meta 資料的用法是結束程式實作特性。比方說，例如結束程式將傳回結果設定為 CANCEL_TRANSFER，且需要與下列已取消傳送的結束程式進行通訊，則必須以其他結束程式理解的方式設定傳送 meta 資料值來執行。

範例

```
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
```

相關概念

[第 214 頁的『利用使用者結束程式自訂 MFT』](#)

您可以使用專屬程式（即所謂的使用者結束常式）來自訂 Managed File Transfer 的特性。

[第 752 頁的『MFT 使用者結束程式的 meta 資料』](#)

有三種不同類型的 meta 資料可提供給 Managed File Transfer 的使用者結束常式使用：環境、傳送及檔案 meta 資料。此 meta 資料可顯示為 Java 鍵值組的對映。

[第 760 頁的『MFT 使用者結束程式的 Java 介面』](#)

本節的主題為您提供使用者結束常式的 Java 介面的相關參照資訊。

相關參考

[第 755 頁的『MFT 資源監視器使用者結束程式』](#)

資源監視器使用者結束程式可讓您配置在滿足監視器的觸發條件時，可在關聯作業啟動之前執行的自訂程式碼。

[第 452 頁的『在 MFT 內容中使用環境變數』](#)

從 IBM WebSphere MQ 7.5 開始，可能會在代表檔案或目錄位置的 Managed File Transfer 內容中使用環境變數。這可讓執行產品的某部分時所使用的檔案或目錄，隨著環境變更而改變其所在位置。例如，執行程序的使用者為何。

[第 463 頁的『MFT agent.properties 檔案』](#)

每一個 Managed File Transfer Agent 都有自己的內容檔 agent.properties，其中必須包含代理程式用來連接至其佇列管理程式的資訊。agent.properties 檔案也可以包含變更代理程式行為的內容。

MFT 使用者結束程式的 Java 介面

本節的主題為您提供使用者結束常式的 Java 介面的相關參照資訊。

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

```
/*
 * 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;
    }
}
```

相關參考

[第 790 頁的『SourceTransferStartExit.java 介面』](#)

[第 764 頁的『DestinationTransferStartExit.java 介面』](#)

[第 763 頁的『DestinationTransferEndExit.java 介面』](#)

[第 783 頁的『MonitorExit.java 介面』](#)

[第 784 頁的『ProtocolBridgeCredentialExit.java 介面』](#)

相關資訊

[利用使用者結束程式自訂 MFT](#)

DestinationTransferEndExit.java 介面

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 null 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);
}
```

相關概念

第 214 頁的『利用使用者結束程式自訂 MFT』

您可以使用專屬程式（即所謂的使用者結束常式）來自訂 Managed File Transfer 的特性。

相關參考

第 790 頁的『SourceTransferStartExit.java 介面』

第 789 頁的『SourceTransferEndExit.java 介面』

第 764 頁的『DestinationTransferStartExit.java 介面』

第 783 頁的『MonitorExit.java 介面』

第 784 頁的『ProtocolBridgeCredentialExit.java 介面』

DestinationTransferStartExit.java 介面

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 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 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);

```

相關概念

第 214 頁的『[利用使用者結束程式自訂 MFT](#)』

您可以使用專屬程式（即所謂的使用者結束常式）來自訂 Managed File Transfer 的特性。

相關參考

第 790 頁的『[SourceTransferStartExit.java 介面](#)』

第 789 頁的『[SourceTransferEndExit.java 介面](#)』

第 763 頁的『[DestinationTransferEndExit.java 介面](#)』

第 783 頁的『[MonitorExit.java 介面](#)』

第 784 頁的『[ProtocolBridgeCredentialExit.java 介面](#)』

[FileTransferResult.java 介面](#)

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();
}

```

相關參考

[第 790 頁的『SourceTransferStartExit.java 介面』](#)

[第 764 頁的『DestinationTransferStartExit.java 介面』](#)

[第 763 頁的『DestinationTransferEndExit.java 介面』](#)

[第 783 頁的『MonitorExit.java 介面』](#)

[第 784 頁的『ProtocolBridgeCredentialExit.java 介面』](#)

相關資訊

[利用使用者結束程式自訂 MFT](#)

[IOExit.java 介面](#)

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;

```

相關概念

第 214 頁的『利用使用者結束程式自訂 MFT』

您可以使用專屬程式（即所謂的使用者結束常式）來自訂 Managed File Transfer 的特性。

相關參考

第 218 頁的『使用 MFT 傳送 I/O 使用者結束程式』

您可以使用 Managed File Transfer 傳送 I/O 使用者結束程式，配置自訂程式碼來執行 Managed File Transfer 傳送的基礎檔案系統 I/O 工作。

IOExitChannel.java 介面

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;
}

```

相關概念

第 214 頁的『[利用使用者結束程式自訂 MFT](#)』

您可以使用專屬程式（即所謂的使用者結束常式）來自訂 Managed File Transfer 的特性。

相關參考

第 218 頁的『[使用 MFT 傳送 I/O 使用者結束程式](#)』

您可以使用 Managed File Transfer 傳送 I/O 使用者結束程式，配置自訂程式碼來執行 Managed File Transfer 傳送的基礎檔案系統 I/O 工作。

IOExitLock.java 介面

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();
}

```

相關概念

第 214 頁的『利用使用者結束程式自訂 MFT』

您可以使用專屬程式（即所謂的使用者結束常式）來自訂 Managed File Transfer 的特性。

相關參考

第 218 頁的『使用 MFT 傳送 I/O 使用者結束程式』

您可以使用 Managed File Transfer 傳送 I/O 使用者結束程式，配置自訂程式碼來執行 Managed File Transfer 傳送的基礎檔案系統 I/O 工作。

IOExitPath.java 介面

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/ftuser/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();
}

```

相關概念

第 214 頁的『[利用使用者結束程式自訂 MFT](#)』

您可以使用專屬程式（即所謂的使用者結束常式）來自訂 Managed File Transfer 的特性。

相關參考

第 218 頁的『[使用 MFT 傳送 I/O 使用者結束程式](#)』

您可以使用 Managed File Transfer 傳送 I/O 使用者結束程式，配置自訂程式碼來執行 Managed File Transfer 傳送的基礎檔案系統 I/O 工作。

IOExitProperties.java 介面

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;
}
}

```

相關概念

第 214 頁的『利用使用者結束程式自訂 MFT』

您可以使用專屬程式（即所謂的使用者結束常式）來自訂 Managed File Transfer 的特性。

相關參考

第 218 頁的『使用 MFT 傳送 I/O 使用者結束程式』

您可以使用 Managed File Transfer 傳送 I/O 使用者結束程式，配置自訂程式碼來執行 Managed File Transfer 傳送的基礎檔案系統 I/O 工作。

IOExitRecordChannel.java 介面

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;
}

```

相關概念

第 214 頁的『[利用使用者結束程式自訂 MFT](#)』

您可以使用專屬程式（即所謂的使用者結束常式）來自訂 Managed File Transfer 的特性。

相關參考

第 218 頁的『[使用 MFT 傳送 I/O 使用者結束程式](#)』

您可以使用 Managed File Transfer 傳送 I/O 使用者結束程式，配置自訂程式碼來執行 Managed File Transfer 傳送的基礎檔案系統 I/O 工作。

IOExitRecordResourcePath.java 介面

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
* {@code 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;
}

```

相關概念

第 214 頁的『[利用使用者結束程式自訂 MFT](#)』

您可以使用專屬程式（即所謂的使用者結束常式）來自訂 Managed File Transfer 的特性。

相關參考

第 218 頁的『[使用 MFT 傳送 I/O 使用者結束程式](#)』

您可以使用 Managed File Transfer 傳送 I/O 使用者結束程式，配置自訂程式碼來執行 Managed File Transfer 傳送的基礎檔案系統 I/O 工作。

IOExitResourcePath.java 介面

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(IOExitResourcePath destination) throws IOException;

/**
 * Creates a new path to use for writing to a temporary resource that did
 * not previously exist.
 * <p>
 * The implementation can choose the abstract path name for the temporary
 * resource. However, for clarity and problem diagnosis, the abstract path
 * name for the temporary resource should be based on this abstract path
 * name with the specified suffix appended and additional characters to make
 * the path unique (for example, sequence numbers), as required.
 * <p>
 * When WMQFTE transfers data to a destination it normally attempts to first
 * write to a temporary resource then on transfer completion renames the
 * temporary resource to the required destination. This method is called by
 * WMQFTE to create a new temporary resource path. The returned path should
 * be new and the resource should not previously exist.
 *
 * @param suffix
 *        Recommended suffix to use for the generated temporary path.
 *
 * @return A new {@link IOExitResourcePath} instance for the temporary
 *         resource path, that did not previously exist.
 * @throws RecoverableIOException
 *         If a recoverable problem occurs whilst attempting to create
 *         the temporary resource. This means that WMQFTE can attempt to
 *         recover the transfer.
 * @throws IOException
 *         If some other I/O problem occurs.
 */
IOExitResourcePath createTempPath(String suffix)
    throws RecoverableIOException, IOException;

/**
 * Opens a {@link IOExitChannel} instance for reading data from the resource
 * denoted by this abstract path. The current data byte position for the
 * resource is expected to be the passed position value, such that when
 * {@link IOExitChannel#read(java.nio.ByteBuffer)} is called, data starting
 * from that position is read.
 *
 * @param position

```

```

*           The required data byte read position.
* @return A new {@link IOExitChannel} instance allowing data to be read
*         from the resource denoted by this abstract path.
* @throws RecoverableIOException
*         If a recoverable problem occurs while attempting to open the
*         resource for reading. This means that WMQFTE can attempt to
*         recover the transfer.
* @throws IOException
*         If some other I/O problem occurs.
*/
IOExitChannel openForRead(long position) throws RecoverableIOException,
    IOException;

/**
* Opens a {@link IOExitChannel} instance for writing data to the resource
* denoted by this abstract path. Writing of data, using the
* {@link IOExitChannel#write(java.nio.ByteBuffer)} method, starts at either
* the beginning of the resource or end of the current data for the
* resource, depending on the specified append parameter.
*
* @param append
*       When {@code true} indicates that data written to the resource
*       should be appended to the end of the current data. When
*       {@code false} indicates that writing of data is to start at
*       the beginning of the resource; any existing data is lost.
* @return A new {@link IOExitChannel} instance allowing data to be written
*         to the resource denoted by this abstract path.
* @throws RecoverableIOException
*         If a recoverable problem occurs whilst attempting to open the
*         resource for writing. This means that WMQFTE can attempt to
*         recover the transfer.
* @throws IOException
*         If some other I/O problem occurs.
*/
IOExitChannel openForWrite(boolean append) throws RecoverableIOException,
    IOException;

/**
* Tests if the resource denoted by this abstract path is in use by another
* application. Typically, this is because another application has a lock on
* the resource either for shared or exclusive access.
*
* @return {@code true} if resource denoted by this abstract path is in use
*         by another application, {@code false} otherwise.
*/
boolean inUse();

/**
* Obtains a {@link IOExitProperties} instance for properties associated
* with the resource denoted by this abstract path.
*
* <p>
* WMQFTE will read these properties to govern how a transfer behaves when
* interacting with the resource.
*
* @return A {@link IOExitProperties} instance for properties associated
*         with the resource denoted by this abstract path.
*/
IOExitProperties getProperties();
}

```

相關概念

[第 214 頁的『利用使用者結束程式自訂 MFT』](#)

您可以使用專屬程式（即所謂的使用者結束常式）來自訂 Managed File Transfer 的特性。

相關參考

[第 218 頁的『使用 MFT 傳送 I/O 使用者結束程式』](#)

您可以使用 Managed File Transfer 傳送 I/O 使用者結束程式，配置自訂程式碼來執行 Managed File Transfer 傳送的基礎檔案系統 I/O 工作。

[IOExitWildcardPath.java 介面](#)

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 {

```

相關概念

第 214 頁的『[利用使用者結束程式自訂 MFT](#)』

您可以使用專屬程式（即所謂的使用者結束帶式）來自訂 Managed File Transfer 的特性。

相關參考

第 218 頁的『[使用 MFT 傳送 I/O 使用者結束程式](#)』

您可以使用 Managed File Transfer 傳送 I/O 使用者結束程式，配置自訂程式碼來執行 Managed File Transfer 傳送的基礎檔案系統 I/O 工作。

MonitorExit.java 介面

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 EnvironmentMetaDataConstants 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 MonitorMetaDataConstants 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);
}

```

相關概念

[第 214 頁的『利用使用者結束程式自訂 MFT』](#)

您可以使用專屬程式（即所謂的使用者結束常式）來自訂 Managed File Transfer 的特性。

相關工作

[第 141 頁的『監視 MFT 資源』](#)

您可以監視 Managed File Transfer 資源；例如，佇列或目錄。滿足此資源的條件時，資源監視器會啟動作業，例如檔案傳送。您可以在 IBM MQ Explorer 的 Managed File Transfer 外掛程式中使用 **fteCreateMonitor** 指令或「監視器」視圖來建立資源監視器。

相關參考

[第 790 頁的『SourceTransferStartExit.java 介面』](#)

[第 789 頁的『SourceTransferEndExit.java 介面』](#)

[第 764 頁的『DestinationTransferStartExit.java 介面』](#)

[第 763 頁的『DestinationTransferEndExit.java 介面』](#)

[第 784 頁的『ProtocolBridgeCredentialExit.java 介面』](#)

[ProtocolBridgeCredentialExit.java 介面](#)

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);
}

```

相關概念

第 214 頁的『利用使用者結束程式自訂 MFT』

您可以使用專屬程式（即所謂的使用者結束常式）來自訂 Managed File Transfer 的特性。

相關工作

第 188 頁的『使用結束類別對映檔案伺服器的認證』

如果您不想要使用通訊協定橋接器代理程式的預設認證對映功能，您可以撰寫專屬的使用者結束程式，將 Managed File Transfer 中的使用者認證對映至檔案伺服器上的使用者認證。Managed File Transfer 提供一個使用者結束程式範例，可執行使用者認證對映。如果您配置認證對映使用者結束程式，它們會取代預設認證對映功能。

ProtocolBridgeCredentialExit2.java 介面

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);
    }

```

相關概念

第 214 頁的『[利用使用者結束程式自訂 MFT](#)』

您可以使用專屬程式（即所謂的使用者結束常式）來自訂 Managed File Transfer 的特性。

相關工作

第 188 頁的『[使用結束類別對映檔案伺服器的認證](#)』

如果您不想要使用通訊協定橋接器代理程式的預設認證對映功能，您可以撰寫專屬的使用者結束程式，將 Managed File Transfer 中的使用者認證對映至檔案伺服器上的使用者認證。Managed File Transfer 提供一個使用者結束程式範例，可執行使用者認證對映。如果您配置認證對映使用者結束程式，它們會取代預設認證對映功能。

ProtocolBridgePropertiesExit2.java 介面

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);
}

```

相關概念

第 184 頁的『[ProtocolBridgePropertiesExit: 查閱通訊協定檔案伺服器內容](#)』

如果您有大量通訊協定檔案伺服器，則可以實作

`com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` 介面來查閱在傳送時參照的通訊協定檔案伺服器內容。您可以實作此介面，這更勝於維護 `ProtocolBridgeProperties.xml` 檔。建議您使用 `ProtocolBridgePropertiesExit2.java` 介面，但同時也支援 `ProtocolBridgePropertiesExit.java` 介面。如果您已從 IBM WebSphere MQ File Transfer Edition 實作 `ProtocolBridgePropertiesExit.java` 介面，則可以在 IBM WebSphere MQ 7.5 或更新版本中使用該介面。`ProtocolBridgePropertiesExit2.java` 中的新 `getCredentialLocation` 方法會使用 `ProtocolBridgeCredentials.xml` 檔案的預設位置，即您的起始目錄。

第 214 頁的『[利用使用者結束程式自訂 MFT](#)』

您可以使用專屬程式（即所謂的使用者結束常式）來自訂 Managed File Transfer 的特性。

相關工作

第 188 頁的『[使用結束類別對映檔案伺服器的認證](#)』

如果您不想要使用通訊協定橋接器代理程式的預設認證對映功能，您可以撰寫專屬的使用者結束程式，將 Managed File Transfer 中的使用者認證對映至檔案伺服器上的使用者認證。Managed File Transfer 提供一個使用者結束程式範例，可執行使用者認證對映。如果您配置認證對映使用者結束程式，它們會取代預設認證對映功能。

`SourceFileExitFileSpecification.java` 類別

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;
    }
}

```

相關概念

第 752 頁的『MFT 使用者結束程式的 meta 資料』

有三種不同類型的 meta 資料可提供給 Managed File Transfer 的使用者結束常式使用：環境、傳送及檔案 meta 資料。此 meta 資料可顯示為 Java 鍵值組的對映。

SourceTransferEndExit.java 介面

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
 * a 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);
}

```

相關概念

第 214 頁的『利用使用者結束程式自訂 MFT』

您可以使用專屬程式（即所謂的使用者結束常式）來自訂 Managed File Transfer 的特性。

相關參考

[第 790 頁的『SourceTransferStartExit.java 介面』](#)

[第 764 頁的『DestinationTransferStartExit.java 介面』](#)

[第 763 頁的『DestinationTransferEndExit.java 介面』](#)

[第 783 頁的『MonitorExit.java 介面』](#)

[第 784 頁的『ProtocolBridgeCredentialExit.java 介面』](#)

[SourceTransferStartExit.java 介面](#)

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);
}

```

相關概念

[第 214 頁的『利用使用者結束程式自訂 MFT』](#)

您可以使用專屬程式（即所謂的使用者結束常式）來自訂 Managed File Transfer 的特性。

相關參考

[第 787 頁的『SourceFileExitFileSpecification.java 類別』](#)

[第 789 頁的『SourceTransferEndExit.java 介面』](#)

[第 764 頁的『DestinationTransferStartExit.java 介面』](#)

[第 763 頁的『DestinationTransferEndExit.java 介面』](#)

[第 783 頁的『MonitorExit.java 介面』](#)

[第 784 頁的『ProtocolBridgeCredentialExit.java 介面』](#)

[TransferExitResult.java 介面](#)

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;
}
}

```

相關參考

[第 790 頁的『SourceTransferStartExit.java 介面』](#)

[第 764 頁的『DestinationTransferStartExit.java 介面』](#)

[第 763 頁的『DestinationTransferEndExit.java 介面』](#)

[第 783 頁的『MonitorExit.java 介面』](#)

[第 784 頁的『ProtocolBridgeCredentialExit.java 介面』](#)

相關資訊

[利用使用者結束程式自訂 MFT](#)

可以放置在 MFT 代理程式指令佇列上的訊息所適用的訊息格式

下列 XML 綱目會定義可以放置在代理程式指令佇列上以要求代理程式執行動作的訊息格式。XML 訊息可透過指令行指令或應用程式，放置在代理程式指令佇列上。

檔案傳送要求訊息格式

檔案傳送是由送達代理程式指令佇列的 XML 訊息起始，通常是由使用者發出檔案傳送指令或使用「IBM MQ Explorer」外掛程式來進行。The transfer request XML must conform to the FileTransfer.xsd schema and have the <request> element as the root element. FileTransfer.xsd 綱目文件位於 `MQ_INSTALLATION_PATH/mqft/samples/schema` 目錄中。FileTransfer.xsd 綱目會匯入 `fteutils.xsd`，它位於相同的目錄中。

檔案傳送訊息可具有下列三個根元素之一：

- <request> - 用於新的檔案傳送要求、受管理的呼叫要求或刪除擱置中的排定傳送
- <cancel> - 用於取消正在進行的檔案傳送
- <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 [使用傳送定義檔](#).

綱目

下列綱目說明哪些元素在傳送要求 XML 訊息中是有效的。

```
<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>
```

```

    </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"/>
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:sequence>

```

```

        <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>

```

瞭解傳送要求訊息

下列清單中說明在傳送要求訊息中使用的元素及屬性：

元素說明

<request>

群組元素，包含指定檔案傳送要求需要的所有元素。

屬性	說明
version	指定此元素的版本，由 Managed File Transfer 提供。

<managedTransfer>

群組元素，包含單一檔案傳送或單一檔案傳送群組需要的所有元素。

<deleteScheduledTransfer>

群組元素，包含用來取消排程傳送的發送端及 ID 資訊。

<managedCall>

群組元素，包含程式或執行檔的單一受管理呼叫需要的所有元素。

<ID>

唯一 ID，指定要從擱置的排定傳送清單中刪除的傳送要求。

<originator>

群組元素，包含指定要求發送端的元素。

<hostName>

來源檔案所在系統的主機名稱。

<userID>

發出檔案傳送的使用者 ID。

<mqmdUserID>

選用項目。在訊息描述子 (MQMD) 中提供的 IBM MQ 使用者 ID。

<schedule>

群組元素，說明檔案傳送的排定時間、重複行為及下次發生時間。

<submit>

指定排定傳送應該開始的日期和時間。

屬性	說明
timebase	指定要使用的時區。此屬性可具有下列其中一個值： <ul style="list-style-type: none"> • source - 使用來源代理程式的時區 • admin - 使用發出此指令的管理者的時區 • UTC - 使用世界標準時間 (UTC)
時區	根據 timebase 值的時區說明

<repeat>

群組元素，包含排定傳送多久重複一次、排定傳送重複的次數及排定傳送何時停止重複的相關詳細資料。

<frequency>

在重複傳送之前必須經歷的時段。

屬性	說明
區間	間隔單位，必須是下列其中一個值： <ul style="list-style-type: none"> • 分鐘 • 時

屬性	說明
	<ul style="list-style-type: none"> • 日 • 週 • 月 • 年

<expireTime>

選用元素，指定排定傳送停止重複的日期和時間。此元素與 <expireCount> 元素互斥。

<expireCount>

選用元素，指定排定檔案傳送在停止之前發生的次數。此元素與 <expireTime> 元素互斥。

<sourceAgent>

指定來源檔案所在系統上的代理程式名稱。

屬性	說明
代理程式 (agent)	指定代理程式的名稱。
QMgr	代理程式佇列管理程式的名稱。

<destinationAgent>

指定系統上要做為檔案傳送目的地的代理程式名稱。

屬性	說明
代理程式 (agent)	指定代理程式的名稱。
QMgr	代理程式佇列管理程式的名稱。
hostName	代理程式佇列管理程式的主機名稱或 IP 位址。
portNumber	用於與目的地代理程式佇列管理程式進行用戶端連線的埠號。
channel	用來連接至目的地代理程式佇列管理程式的通道名稱。

<trigger>

選用元素，指定要進行檔案傳送時必須具備的條件。

屬性	說明
log	指出是否記載觸發失敗的旗標。有效值如下所示： <ul style="list-style-type: none"> • yes - 會針對失敗的觸發傳送建立日誌項目 • no - 不會針對失敗的觸發傳送建立日誌項目

<fileExist>

指定以逗點區隔的檔案名稱清單，這些檔案名稱位於與來源代理程式相同的系統上。如果此名稱清單中的檔案滿足觸發的條件，就會進行傳送。此元素與 <fileSize> 元素互斥。

屬性	說明
comparison	指出如何根據名稱清單評估來源檔案名稱。有效值如下所示： <ul style="list-style-type: none"> • = 名稱清單中至少有一個檔案名稱必須符合 • != 名稱清單中至少有一個檔案不存在
value	指出比較類型：

屬性	說明
	<ul style="list-style-type: none"> • exist: 檔案必須存在

<fileSize>

指定以逗點區隔的檔案名稱清單，這些檔案名稱位於與來源代理程式相同的系統上。如果此名稱清單中的檔案滿足觸發的條件，就會進行傳送。此元素與 <fileExist> 元素互斥。

屬性	說明
comparison	指出如何根據名稱清單評估來源檔案名稱。有效值如下所示： <ul style="list-style-type: none"> • >= 名稱清單中有一個檔案名稱存在，且檔案大小最少為 value 屬性指定的大小
value	指定為整數值的檔案大小，其單位指定為下列其中一項： <ul style="list-style-type: none"> • B - 位元組 • KB - 千位元組 • MB - 百萬位元組 • GB - 十億位元組 (單位值不區分大小寫)

<reply>

指定對同步檔案傳送產生的暫時回覆佇列名稱（在指令行上使用 **-w** 參數指定）。佇列名稱是由 `command.properties` 配置檔中的索引鍵 **dynamicQueuePrefix** 或預設值 `WMQFTE.*` 所定義。（如果未指定的話）。

屬性	說明
詳細資料	回覆訊息中是否應包含詳細的傳輸結果資訊。可以針對每個傳送產生多個回覆訊息。有效值如下所示： <ul style="list-style-type: none"> • true - 必須包含詳細回覆資訊。此資訊格式與進度訊息中發佈到傳送日誌的格式相同，亦即 <transferSet> 元素。如需相關資訊，請參閱第 637 頁的『檔案傳送日誌訊息格式』。僅當傳輸來源代理程式的 <code>enableDetailedReplyMessages</code> 內容設定為 true 時，才會顯示詳細回覆資訊。 • false - 無須包含詳細回覆資訊。 預設值是 false。
QMGR	產生暫時動態佇列以接收回覆的指令佇列管理程式的名稱。
persistent	寫入回覆佇列的訊息是否具有持續性。有效值如下所示： <ul style="list-style-type: none"> • true - 訊息具有持續性 • false - 訊息不具有持續性 • qdef - 訊息的持續性由回覆佇列的內容定義 預設值是 false。

<transferSet>

指定您想要一起執行的檔案傳送群組，或您想要一起執行的受管理呼叫群組。在傳輸期間，<transferSet> 是包含 <item> 元素的群組元素。

屬性	說明
priority	傳送的優先順序層次。優先順序是範圍 0-9 中的值，其中 0 是最低優先順序。預設優先順序層次是 0，依預設，傳送時會使用來源代理程式的優先順序層次。

<metaDataSet>

選用群組元素，包含一個以上 meta 資料項目。

<metaData>

指定使用者定義的 meta 資料，其傳送至代理程式呼叫的結束點。此元素包含字串形式的 meta 資料值。

屬性	說明
key	字串形式的 meta 資料名稱

<call>

群組元素，包含指定要呼叫的程式或執行檔的 <command> 元素。

<command>

指定要呼叫的程式或執行檔。此指令必須位於代理程式指令路徑中。如需相關資訊，請參閱第 464 頁的表 45。此元素可以包含選用的 <argument> 元素。

屬性	說明
名	指令的名稱。
successRC	此指令傳回的成功回覆碼。預設值是 0。
retryCount	指令失敗時重試指令的次數。
retryWait	指令重試之間要等待的時間（以秒為單位）。
類型	要呼叫的程式類型。有效值是 antscript、jcl 或 executable。

<argument>

指定要傳遞至指令的引數。

<item>

群組元素，包含指定來源及目的地檔案名稱和位置的元素。


屬性	說明
mode	指定傳送模式為二進位或文字模式。
checksumMethod	指定產生訊息摘要以建立數位簽章的雜湊演算法類型。有效值是 MD5 或 none。



<source>

群組元素，指定來源系統上的檔案以及傳送完成之後是否移除這些檔案

屬性	說明
recursive	指定當 <source> 元素是目錄或包含萬用字元時，在子目錄中遞迴地傳送檔案。
disposition	指定當 <source> 已順利傳送至其目的地時，對 <source> 元素採取的動作。有效值如下所示： <ul style="list-style-type: none"> • leave - 來源檔案維持不變。 • delete - 順利傳送來源檔案之後，即從來源系統中刪除來源檔案。

<file>

指定傳送來源。  若為 Multiplatforms，傳送來源可以是檔案或目錄名稱。對於 z/OS 平台，傳送來源可以是檔案、目錄、資料集或 PDS 名稱。請使用完整路徑，其格式與作業系統一致，例如 C:/from/here.txt。請不要使用檔案 URI。

屬性	說明
別名 (alias)	指定來源檔案的別名。此別名是來源檔案的名稱，不包括針對傳送所指定的任何目錄路徑。
EOL	指定文字傳送的行尾標記。有效值為： <ul style="list-style-type: none"> • LF - 僅限換行字元 • CRLF - 換行 (carriage return/line feed) 字元序列
encoding	文字檔傳送的來源檔案編碼。
 delimiter	指定在記錄導向的來源檔案（例如，z/OS 資料集）中，記錄之間所內含的定界字元。將定界字元值指定為介於 00-FF 範圍內的兩個十六進位數字，並以 x 作為字首。例如，x12 或 x03,x7F。
delimiterType	指定包括在目的地檔案中的個別訊息資料之後的定界字元類型。有效值如下所示： <ul style="list-style-type: none"> • binary - 十六進位定界字元 僅當您已啟用 7.0.4.1 功能時，此屬性才可用。
delimiterPosition	指定將記錄導向的來源檔案記錄寫入一般檔案時，要插入定界字元的位置。有效值如下所示： <ul style="list-style-type: none"> • prefix - 將定界字元插入到目的地檔案中，每一個來源記錄導向檔案記錄的資料之前。 • postfix - 將定界字元插入到目的地檔案中，每一個來源記錄導向檔案記錄的資料之後。
includeDelimiterInFile	指定是否在記錄導向的來源檔案記錄之間併入定界字元。
 keepTrailingSpaces	指定在文字模式傳送時，是否在從固定長度格式資料集讀取的來源記錄中，保留尾端空格。預設值是刪除尾端空格。有效值如下所示： <ul style="list-style-type: none"> • true - 在從固定長度格式資料集讀取的來源記錄中，保留尾端空格。 • false - 在從固定長度格式資料集讀取的來源記錄中，刪除尾端空格。

<queue>

與 <source> 元素一起使用時，指定要從中傳送的佇列名稱，該佇列必須位於來源代理程式佇列管理程式上。請使用 *QUEUE* 格式。請不要併入佇列管理程式名稱，該佇列必須在來源代理程式佇列管理程式上。如果您在 <destination> 元素內使用 <queue> 元素，就不能在 <source> 元素內使用它。



屬性	說明
useGroups	指定是否只從來源佇列傳送第一個完整訊息群組。有效值如下所示： <ul style="list-style-type: none"> • true - 只傳送第一個完整訊息群組 • false - 傳送來源佇列上的所有訊息
groupId	指定要從來源佇列讀取的訊息群組。只有在 useGroups 屬性的值為 true 時，此屬性才有效。
delimiterType	指定包括在目的地檔案中的個別訊息資料之後的定界字元類型。有效值如下所示： <ul style="list-style-type: none"> • text - 文字或 Java 文字定界字元 • binary - 十六進位定界字元
delimiter	指定包括在目的地檔案中的個別訊息資料之間的定界字元。

屬性	說明
delimiterPosition	指定將定界字元併入在目的地檔案中的個別訊息資料之前或之後。有效值如下所示： <ul style="list-style-type: none"> • prefix - 將定界字元併入在資料之前 • postfix - 將定界字元併入在資料之後
encoding	指定來源佇列編碼。
waitTime	指定來源代理程式等待下列情況的時間（以秒為單位）： <ul style="list-style-type: none"> • 訊息出現在來源佇列上（如果該佇列是空的或已變成空的） • 一個完整群組出現在來源佇列上（如果 useGroups 屬性已設為 true） 如需設定 waitTime 值的相關資訊，請參閱第 574 頁的『指定「訊息轉為檔案」傳送等待時間的指引』。

<destination>

指定目的地及行為的群組元素（如果檔案已存在於目的地代理程式上）。

您只能指定 <file> 和 <queue> 其中之一作為目的地的子元素。

屬性	說明
類型	目的地的類型。有效值如下所示： <ul style="list-style-type: none"> • file - 將檔案指定為目的地 • directory - 將目錄指定為目的地 •  dataset - 指定 z/OS 資料集作為目的地 •  pds - 指定 z/OS 分割的資料集作為目的地 • queue - 指定 IBM MQ 佇列作為目的地 • filespace - 將檔案空間指定為目的地 <p>只有在 <destination> 元素具有 <queue> 的子元素時，值 佇列 才有效。</p> <p>只有在 <destination> 元素具有 <filespace> 的子元素時，值 檔案空間 才有效。</p> <p>只有在 <destination> 元素具有 <file> 的子元素時，其他值才有效。</p>
exist	指定如果目的地系統上存在目的地檔案時要採取的動作。有效值如下所示： <ul style="list-style-type: none"> • error - 報告錯誤，且不傳送檔案。 • overwrite - 改寫現有目的地檔案。 <p>如果 <destination> 元素具有 <queue> 或 <filespace> 的子元素，則此屬性無效。</p>

<file>

指定先前說明的 <destination> 元素的其他設定。請使用完整路徑，其格式與作業系統一致，例如 C:/from/here.txt。請不要使用檔案 URI。

屬性	說明
別名(alias)	指定 <destination> 檔案的別名。此別名是來源檔案的名稱，不包括針對傳送所指定的任何目錄路徑。
encoding	文字檔傳送的 <destination> 檔編碼。
EOL	指定文字傳送的行尾標記。有效值為：

屬性	說明
	<ul style="list-style-type: none"> • LF - 僅限換行字元 • CRLF - 換行 (carriage return/line feed) 字元序列
truncateRecords	選用項目。指定截斷長度超過 LRECL 資料集屬性的 <destination> 記錄。 <ul style="list-style-type: none"> • True - 將記錄截斷 • False - 將記錄換行 預設值為 false。

<queue>

與 **<destination>** 元素一起使用時，指定要傳送至其中的佇列名稱，該佇列可以位於連接至目的地代理程式佇列管理程式的任何佇列管理程式上。請使用 *QUEUE@QM* 格式，其中 *QUEUE* 是放置訊息的佇列名稱，*QM* 是佇列所在的佇列管理程式。如果您在 **<source>** 元素內使用 **<queue>** 元素，就不能在 **<destination>** 元素內使用它。

屬性	說明
delimiter	將檔案分割成多個訊息的定界字元。
delimiterType	指定定界字元的類型。有效值如下所示： <ul style="list-style-type: none"> • text - Java 正規表示式 • binary - 十六進位位元組序列 • size - 位元組、千位元組或百萬位元組的數目。例如，1 B、1 K 或 1 M。
delimiterPosition	指定預期是在併入個別訊息中的資料之前還是之後會有定界字元。有效選項如下所示： <ul style="list-style-type: none"> • prefix - 預期定界字元在資料之前 • postfix - 預期定界字元在資料之後
includeDelimiterInMessage	此布林值指定是否在訊息結尾併入用來將檔案分割成多個訊息的定界字元。
encoding	指定目的地佇列編碼。
persistent	指定訊息是否具有持續性。有效值如下所示： <ul style="list-style-type: none"> • true - 訊息具有持續性 • false - 訊息不具有持續性 • qdef - 訊息的持續性值由目的地佇列的設定定義
setMqProps	此布林值指定是否將 IBM MQ 訊息內容設定在檔案的第一個訊息上，且在發生錯誤時是否將所有訊息寫入佇列中。
unrecognisedCodePage	指定如果目的地佇列管理程式無法辨識資料字碼頁，則文字模式傳送是失敗，還是執行轉換。有效值如下所示： <ul style="list-style-type: none"> • fail - 傳送報告失敗 • binary - 資料會轉換成目的地字碼頁，且說明資料格式的 IBM MQ 訊息標頭會設定為 MQFMT_NONE。 預設行為是 fail。

<filespace>

群組元素，指定要傳送至其中的檔案空間名稱。

<name>

與 <fileSpace> 元素一起使用時，此元素的值指定檔案空間的名稱。

<preSourceCall>

群組元素，指定在傳送開始之前，要在傳送來源呼叫的指令。

<postSourceCall>

群組元素，指定在傳送完成之後，要在傳送來源呼叫的指令。

<preDestinationCall>

群組元素，指定在傳送開始之前，要在傳送目的地呼叫的指令。

<postDestinationCall>

群組元素，指定在傳送完成之後，要在傳送目的地呼叫的指令。

<command>

與 <preSourceCall>、<postSourceCall>、<preDestinationCall> 或 <postDestinationCall> 元素一起使用時，此元素指定要呼叫的指令。此指令必須位於代理程式指令路徑中。如需相關資訊，請參閱第 464 頁的表 45。

屬性	說明
名	要執行的指令名稱。
successRC	指令順利執行時所預期的回覆碼。

<argument>

與 <command> 元素一起使用時，此元素指定要傳入指令中的引數。<command> 元素內可以有任意數量的 <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 637 頁的『檔案傳送日誌訊息格式』](#)。

<name>

與 <job> 元素一起使用時，此元素的值指定工作的名稱。

<transferSpecifications>

群組元素，包含用於多個傳送群組的 <item> 元素。如需如何使用此元素的進一步詳細資料，請參閱[使用傳送定義檔](#)。

<cancel>

群組元素，包含取消進行中的檔案傳送需要的所有元素。

屬性	說明
version	指定此元素的版本，由 Managed File Transfer 提供。

<transfer>

與 <cancel> 元素一起使用時，此元素的值指定要取消的傳送要求 ID。

<job>

包含工作資訊的群組元素。

<jobName>

指定邏輯工作 ID。

檔案傳送取消訊息格式

檔案傳送要求會傳回 48 個字元的 ID，以識別特定代理程式的傳送。這個 ID 可用來取消傳送。

瞭解傳送取消訊息

以下說明在傳送取消訊息中使用的元素及屬性：

<cancel>

群組元素，包含取消進行中的檔案傳送需要的所有元素。

屬性	說明
version	指定此元素的版本，由 Managed File Transfer 提供。

<originator>

群組元素，包含指定要求發送端的元素。

<hostName>

來源檔案所在系統的主機名稱。

<userID>

發出檔案傳送的使用者 ID。

<mqmdUserID>

選用項目。在訊息描述子 (MQMD) 中提供的 IBM MQ 使用者 ID。

<transfer>

與 <cancel> 元素一起使用時，此元素的值指定要取消的傳送要求 ID。

<job>

選用項目。包含工作資訊的群組元素。

<jobName>

指定邏輯工作 ID。

範例

針對下列每一個要求，提供了符合此綱目的 XML 訊息的範例：

- [建立檔案傳送](#)
- [建立非同步檔案傳送要求](#)
- [取消檔案傳送](#)
- [建立排定的傳送](#)
- [刪除排定的傳送](#)
- [建立受管理的呼叫](#)
- [建立包括受管理呼叫的檔案傳送](#)

檔案傳送要求訊息範例

您可以放置在代理程式指令佇列上以要求代理程式建立或取消傳送的訊息範例。

建立傳送要求

```
<?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>

```

建立傳送要求 - 同步

當使用者要求區塊傳輸同步要求時（亦即，他們會等待傳送完成並接收狀態訊息），放置在指令佇列上的訊息會包含一個回覆元素，指定回覆訊息要傳送至其中的佇列。下列範例顯示放置在 FTEAGENT 所使用的指令佇列上的訊息：

```

<?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>

```

在 <reply> 元素中移入了指令佇列管理程式的名稱，此佇列管理程式中已建立暫時動態佇列來接收順利（或未順利）完成傳送的相關回覆。暫時動態佇列的名稱由下列兩個部分組成：

- command.properties 配置檔中金鑰 **dynamicQueuePrefix** 所定義的字首（它是 WMQFTE。依預設）
- IBM MQ 所產生的佇列 ID

取消傳送要求

```

<?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>414D5120553322E42494E44494E47538B0F404D032C0020</transfer>
  <reply QMGR="QM_JUPITER">WMQFTE.4D400F8B20002007</reply>
</cancel>

```

相關參考

第 668 頁的『檔案傳送要求訊息格式』

檔案傳送是由送達代理程式指令佇列的 XML 訊息起始，通常是由使用者發出檔案傳送指令或使用「IBM MQ Explorer」外掛程式來進行。The transfer request XML must conform to the FileTransfer.xsd schema and have the <request> element as the root element. FileTransfer.xsd 綱目文件位於

MQ_INSTALLATION_PATH/mqft/samples/schema 目錄中。FileTransfer.xsd 綱目會匯入 fteutils.xsd，它位於相同的目錄中。

排定的檔案傳送訊息範例

以下範例說明您可以放置在代理程式指令佇列上以要求代理程式建立或刪除排程的訊息。

建立排定的傳送

```
<?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>
```

刪除排定的傳送

```
<?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>
```

相關參考

第 668 頁的『檔案傳送要求訊息格式』

檔案傳送是由送達代理程式指令佇列的 XML 訊息起始，通常是由使用者發出檔案傳送指令或使用「IBM MQ Explorer」外掛程式來進行。The transfer request XML must conform to the FileTransfer.xsd schema and have the <request> element as the root element. FileTransfer.xsd 綱目文件位於 MQ_INSTALLATION_PATH/mqft/samples/schema 目錄中。FileTransfer.xsd 綱目會匯入 fteutils.xsd，它位於相同的目錄中。

MFT 代理程式呼叫要求訊息範例

訊息範例，您可以將訊息放在代理程式指令佇列上，以要求代理程式建立受管理呼叫，或是建立呼叫程式的傳送。

受管理呼叫要求範例

```
<?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>
    </job>
  </managedCall>
</request>
```

具有呼叫的受管理傳送要求範例

```
<?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>
    </job>
  </managedTransfer>
</request>
```

相關概念

第 209 頁的『指定要使用 MFT 執行的程式』

您可以在執行 Managed File Transfer Agent 的系統上執行程式。在檔案傳送要求過程中，您可以指定程式在傳送開始前或結束後執行。此外，您也可以提交受管理的呼叫要求，以啟動不在檔案傳送要求過程中的程式。

相關參考

第 668 頁的『檔案傳送要求訊息格式』

檔案傳送是由送達代理程式指令佇列的 XML 訊息起始，通常是由使用者發出檔案傳送指令或使用「IBM MQ Explorer」外掛程式來進行。The transfer request XML must conform to the FileTransfer.xsd schema and have the <request> element as the root element. FileTransfer.xsd 綱目文件位於 MQ_INSTALLATION_PATH/mqft/samples/schema 目錄中。FileTransfer.xsd 綱目會匯入 fteutils.xsd，它位於相同的目錄中。

MFT 監視器要求訊息格式

當適用的 XML 訊息送達代理程式的指令佇列時，即會建立資源監視器；這通常是因為使用者發出 fteCreateMonitor 指令或使用「IBM MQ Explorer」介面所造成。

監視器 XML 必須符合 Monitor.xsd 綱目，使用 <monitor> 元素作為根元素。

監視器訊息可具有下列其中一個根元素：

- <monitor> - 用於建立及啟動新的資源監視器
- <deleteMonitor> - 用於停止及刪除現有監視器

fteListMonitors 指令沒有指令訊息，因為此指令會直接從 SYSTEM.FTE 主題擷取符合的監視定義。

綱目

下列綱目說明在監視器要求 XML 訊息中有效的元素。

```
<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>

```

瞭解建立監視器訊息

以下說明在建立監視器訊息中使用的元素及屬性：

元素說明

<monitor>

群組元素，包含取消進行中的檔案傳送需要的所有元素。

屬性	說明
version	指定此元素的版本，由 Managed File Transfer 提供。

<name>

監視器的名稱，必須是監視器代理程式內的唯一名稱。

<description>

監視器的說明（目前未使用）。

<pollInterval>

每一次根據觸發條件檢查資源之間的時間間隔。

屬性	說明
單元：	指定輪詢間隔的時間單位。有效值為： <ul style="list-style-type: none"> • 秒 • 分鐘 • 時 • 日 • 週

屬性	說明
	<ul style="list-style-type: none"> • 月 • 年

<agent>

與監視器相關聯的代理程式名稱。

<resources>

群組元素，包含指定監視器資源的元素。

<directory>

完整路徑，指定要在監視器的代理程式機器上監視的目錄。

屬性	說明
recursionLevel	除了指定的目錄外所要監視的子目錄數目。
ID	資源的唯一 ID。

<queue>

佇列名稱，指定要在監視代理程式的佇列管理程式上監視的佇列。

<triggerMatch>

群組元素，包含指定與受監視資源相互比較之觸發條件的元素。

<conditions>

群組元素，包含指定與受監視資源相互比較之條件類型的元素。

<allOf>

指定必須滿足所有內含條件的述詞。

<anyOf>

指定必須滿足任何內含條件的述詞。

<condition>

定義將納入整體監視器觸發條件的比較條件。

<name>

條件的名稱。

<resource>

識別據以比較條件的資源定義。

屬性	說明
ID	資源的唯一 ID。

如果所要監視的資源是目錄，則必須在條件中指定下列三個元素之一：

- fileMatch
- fileNoMatch
- fileSize

如果所要監視的資源是佇列，則必須在條件中指定下列兩個元素之一：

- queueNotEmpty
- completeGroups

<fileMatch>

檔案名稱符合條件的群組元素。

<pattern>

指定檔案名稱符合型樣。資源上的檔案必須符合此型樣，才能滿足條件。預設型樣是 *（任何檔案皆符合）。

<fileNoMatch>

反向檔案名稱符合條件的群組元素。

<pattern>

指定反向檔案名稱符合型樣。如果受監視資源上沒有符合的檔案，則滿足條件。預設型樣是 *（無任何檔案即符合）。

<fileSize>

檔案大小比較的群組元素。

<compare>

指定檔案大小比較。這個值必須是非負整數。

屬性	說明
operator	要使用的比較運算子。僅支援 > = '。
單位:	指定檔案大小的單位，可以是下列其中一項： <ul style="list-style-type: none"> • B - 位元組 • KB - 千位元組 • MB - 百萬位元組 • GB - 十億位元組 單位值不區分大小寫，因此 mb 的作用與 MB 相同。

<pattern>

要比對的檔案名稱型樣。預設值是 *（任何檔案皆符合）。

<queueNotEmpty>

只有在資源是佇列時，才可指定此項目。指定佇列上必須有訊息，才會觸發監視器。

<completeGroups>

只有在資源是佇列時，才可指定此項目。指定佇列上必須有完整訊息群組存在，才會觸發監視器。對於佇列上的每一個完整群組，分別執行單一傳送作業。

<reply>

選用元素，用來指定非同步要求的回覆佇列。

屬性	說明
QMGR	佇列管理程式名稱。

<tasks>

群組元素，包含的元素指定滿足監視器觸發條件時所要呼叫的作業。

<task>

群組元素，定義監視器在滿足觸發條件時所要呼叫的個別作業。目前只能指定一項作業。

<name>

作業的名稱。接受任何英數字元。

<description>

作業的說明。接受任何文字值。

<transfer>

定義傳送作業的群組元素。

<request>

定義作業類型的群組元素。其中必須包含下列其中一個繼承自 FileTransfer.xsd 綱目定義的元素：

- [managedTransfer](#)
- managedCall

屬性	說明
version	由 Managed File Transfer 提供的要求版本。其格式為 n.mm，其中 n 是主要發行版本，mm 是次要版本。例如，1.00。

<originator>

群組元素，包含指定要求發送端的元素。

<hostName>

來源檔案所在系統的主機名稱。

<userID>

發出檔案傳送的使用者 ID。

<mqmdUserID>

選用項目。在訊息描述子 (MQMD) 中提供的 IBM MQ 使用者 ID。

<job>

包含工作資訊的群組元素。

<jobName>

指定邏輯工作 ID。

<defaultVariables>

群組元素，包含個以上 variable 元素。在監視佇列時，將這些變數用於變數替代中。如需變數替代的相關資訊，請參閱第 149 頁的『使用變數替代來自訂 MFT 資源監視器作業』。

<variable>

一個元素，包含與 key 屬性指定的索引鍵相關聯的值。

屬性	說明
key	預設變數的名稱。

瞭解刪除監視器訊息

以下說明在刪除監視器訊息中使用的元素及屬性：

元素說明

<deleteMonitor>

群組元素，包含停止及刪除監視器需要的所有元素。

屬性	說明
version	指定此元素的版本，由 Managed File Transfer 提供。

<name>

要刪除的監視器名稱。

<originator>

群組元素，包含指定要求發送端的元素。

<hostName>

來源檔案所在系統的主機名稱。

<userID>

發出檔案傳送的使用者 ID。

<mqmdUserID>

選用項目。在訊息描述子 (MQMD) 中提供的 IBM MQ 使用者 ID。

<reply>

指定對要求產生的暫時回覆佇列名稱。佇列名稱是由 command.properties 配置檔中的索引鍵 dynamicQueuePrefix 定義。如果未指定此項目，佇列名稱將會使用預設值 WMQFTE。

屬性	說明
QMGR	產生暫時動態佇列以接收回覆的指令佇列管理程式的名稱。

範例

以下範例說明針對下列每一個監視器要求，提供符合此綱目的 XML 訊息：

- [建立監視器](#)
- [刪除監視器](#)

MFT 監視器要求訊息範例

以下範例說明您可以放置在代理程式指令佇列上以要求代理程式建立或刪除監視器的訊息。

建立監視器要求

```
<?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>
```


刪除監視器要求

```
<?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>
```

相關參考

第 684 頁的『MFT 監視器要求訊息格式』

當適用的 XML 訊息送達代理程式的指令佇列時，即會建立資源監視器；這通常是因為使用者發出 `fteCreateMonitor` 指令或使用「IBM MQ Explorer」介面所造成。

連線測試 MFT 代理程式要求訊息格式

您可以發出 `ftePingAgent` 指令或將 XML 訊息放置在代理程式指令佇列上，以對代理程式進行連線測試。連線測試代理程式要求 XML 必須符合 `PingAgent.xsd` 綱目。安裝 Managed File Transfer 之後，您可以在下列目錄中找到 `PingAgent.xsd` 綱目檔：`MQ_INSTALLATION_PATH/mqft/samples/schema`。`PingAgent.xsd` 綱目會匯入 `fteutils.xsd`，它位於相同的目錄。

當代理程式在其指令佇列上收到連線測試代理程式要求訊息時，如果代理程式在作用中，則會傳回 XML 回應訊息給指令，或傳回給將連線測試代理程式要求訊息放入指令佇列中的應用程式。代理程式的回應訊息符合 `Reply.xsd` 所定義的格式。如需此格式的相關資訊，請參閱第 694 頁的『MFT 代理程式回覆訊息格式』。

綱目

下列綱目說明哪些元素在連線測試代理程式要求 XML 訊息中是有效的。

```
<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>
```

瞭解連線測試代理程式要求訊息

下列清單中說明在連線測試代理程式要求訊息中使用的元素及屬性：

<pingAgent>

群組元素，包含指定連線測試代理程式要求需要的所有元素。

<originator>

群組元素，包含指定連線測試要求發送端需要的所有元素。

<hostName>

發出要求的機器的主機名稱。

<userID>

要求發送端的使用者名稱。

<mqmdUserID>

要求發送端的 MQMD 使用者名稱。

<agent>

要進行連線測試的代理程式。

屬性	說明
代理程式 (agent)	必要項目。代理程式的名稱。
QMgr	選用項目。代理程式所連接的佇列管理程式。

<reply>

接收代理程式所傳送回覆訊息的佇列名稱。

屬性	說明
QMGR	必要項目。回覆佇列所在的佇列管理程式的名稱。

範例

此範例顯示傳送至代理程式 AGENT_JUPITER 的連線測試代理程式訊息。如果 AGENT_JUPITER 在作用中且能夠處理代理程式要求，則會將回應訊息傳送至 QM_JUPITER 的佇列 WMQFTE.4D400F8B20003708。

```
<?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 代理程式回覆訊息格式

當代理程式在其代理程式指令佇列上收到 XML 訊息時，如果需要回應，代理程式會將 XML 回覆訊息傳送至原始訊息中定義的回覆佇列。回覆 XML 符合 Reply.xsd 綱目。Reply.xsd 綱目文件位於 MQ_INSTALLATION_PATH/mqft/samples/schema 目錄中。Reply.xsd 綱目會匯入 fteutils.xsd，它位於相同的目錄。

綱目

下列綱目說明在回覆 XML 訊息中有效的元素。

```
<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>
```

瞭解回覆訊息

下列清單中說明回覆訊息中使用的元素及屬性：

<reply>

此元素包含指定回覆資訊的元素。

屬性	說明
ID	回覆的 ID。
version	回覆訊息格式的版本。
detailedReplyMessagesDisabled	這是關於代理程式已停用詳細回覆的通知（enableDetailedReplyMessages 代理程式內容設定為 false）。

<transferSet>

指定要求傳輸的檔案傳輸結果資訊。如需相關資訊，請參閱第 637 頁的『檔案傳送日誌訊息格式』。

<status>

要求代理程式執行的動作的狀態。

屬性	說明
resultCode	代理程式執行的動作所傳回的結果碼。

<supplement>

要求代理程式執行的動作的其他回應資訊。

範例

下一段是回覆訊息範例：

```
<reply version="1.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="Reply.xsd"
      ID="0102020300000000000000000000000000000000000000000000000000000000">
  <status resultCode="65">
    <supplement>Additional reply information</supplement>
  </status>
</reply>
```

MFT 診斷訊息

以下將按數字順序提供診斷訊息，並根據這些診斷訊息所源自的 Managed File Transfer 組件對其進行分組。

如需這些訊息的詳細資料，請參閱 IBM Documentation: https://www.ibm.com/docs/SSFKSJ_9.0.0/com.ibm.wmqfte.doc/messages_main.html

注意事項

本資訊係針對 IBM 在美國所提供之產品與服務所開發。

在其他國家中，IBM 可能不會提供本書中所提的各項產品、服務或功能。請洽當地 IBM 業務代表，以取得當地目前提供的產品和服務之相關資訊。這份文件在提及 IBM 的產品、程式或服務時，不表示或暗示只能使用 IBM 的產品、程式或服務。只要未侵犯 IBM 的智慧財產權，任何功能相當的產品、程式或服務都可以取代 IBM 的產品、程式或服務。不過，任何非 IBM 的產品、程式或服務，使用者必須自行負責作業的評估和驗證責任。

本文件所說明之主題內容，IBM 可能擁有其專利或專利申請案。提供本文件不代表提供這些專利的授權。您可以書面提出授權查詢，來函請寄到：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

如果是有關雙位元組 (DBCS) 資訊的授權查詢，請洽詢所在國的 IBM 智慧財產部門，或書面提出授權查詢，來函請寄到：

智慧財產權授權
法務部與智慧財產權法律
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

下列段落不適用於英國，若與任何其他國家之法律條款抵觸，亦不適用於該國： International Business Machines Corporation 只依 "現況" 提供本出版品，不提供任何明示或默示之保證，其中包括且不限於不侵權、可商用性或特定目的之適用性的隱含保證。有些地區在特定交易上，不允許排除明示或暗示的保證，因此，這項聲明不一定適合您。

這項資訊中可能有技術上或排版印刷上的訛誤。因此，IBM 會定期修訂；並將修訂後的內容納入新版中。IBM 隨時會改進及/或變更本出版品所提及的產品及/或程式，不另行通知。

本資訊中任何對非 IBM 網站的敘述僅供參考，IBM 對該網站並不提供任何保證。這些網站所提供的資料不是 IBM 本產品的資料內容，如果要使用這些網站的資料，您必須自行承擔風險。

IBM 得以各種適當的方式使用或散布由您提供的任何資訊，無需對您負責。

如果本程式的獲授權人為了 (i) 在個別建立的程式和其他程式（包括本程式）之間交換資訊，以及 (ii) 相互使用所交換的資訊，因而需要相關的資訊，請洽詢：

IBM Corporation
軟體交互作業能力協調程式，部門 49XA
3605 公路 52 N
Rochester, MN 55901
U.S.A.

在適當條款與條件之下，包括某些情況下（支付費用），或可使用此類資訊。

IBM 基於雙方之 IBM 客戶合約、IBM 國際程式授權合約或任何同等合約之條款，提供本資訊所提及的授權程式與其所有適用的授權資料。

本文件中所含的任何效能資料都是在受管制的環境下判定。因此不同作業環境之下所得的結果，可能會有很大的差異。有些測定已在開發階段系統上做過，不過這並不保證在一般系統上會出現相同結果。甚至有部分的測量，是利用插補法而得的估計值，實際結果可能有所不同。本書的使用者應依自己的特定環境，查證適用的資料。

本文件所提及之非 IBM 產品資訊，取自產品的供應商，或其發佈的聲明或其他公開管道。IBM 並未測試過這些產品，也無法確認這些非 IBM 產品的執行效能、相容性或任何對產品的其他主張是否完全無誤。有關非 IBM 產品的性能問題應直接洽詢該產品供應商。

有關 IBM 未來方針或目的之所有聲明，僅代表 IBM 的目標與主旨，隨時可能變更或撤銷，不必另行通知。

這份資訊含有日常商業運作所用的資料和報告範例。為了要使它們儘可能完整，範例包括個人、公司、品牌和產品的名稱。這些名稱全屬虛構，如與實際公司的名稱和住址雷同，純屬巧合。

著作權授權：

本資訊含有原始語言之範例應用程式，用以說明各作業平台中之程式設計技術。您可以基於研發、使用、銷售或散布符合作業平台（撰寫範例程式的作業平台）之應用程式介面的應用程式等目的，以任何形式複製、修改及散布這些範例程式，而不必向 IBM 付費。這些範例並未在所有情況下完整測試。因此，IBM 不保證或暗示這些程式的可靠性、有用性或功能。

若貴客戶正在閱讀本項資訊的電子檔，可能不會有照片和彩色說明。

程式設計介面資訊

程式設計介面資訊 (如果有提供的話) 旨在協助您建立與此程式搭配使用的應用軟體。

本書包含預期程式設計介面的相關資訊，可讓客戶撰寫程式以取得 WebSphere MQ 的服務。

不過，本資訊也可能包含診斷、修正和調整資訊。提供診斷、修正和調整資訊，是要協助您進行應用軟體的除錯。

重要：請勿使用此診斷、修改及調整資訊作為程式設計介面，因為它可能會變更。

商標

IBM、IBM 標誌 [ibm.com](http://www.ibm.com) 是 IBM Corporation 在全球許多適用範圍的商標。IBM 商標的最新清單可在 Web 的 "Copyright and trademark information" www.ibm.com/legal/copytrade.shtml 中找到。其他產品和服務名稱，可能是 IBM 或其他公司的商標。

Microsoft 及 Windows 是 Microsoft Corporation 在美國及/或其他國家或地區的商標。

UNIX 是 The Open Group 在美國及/或其他國家/地區的註冊商標。

Linux 是 Linus Torvalds 在美國及/或其他國家或地區的註冊商標。

本產品包含 Eclipse Project (<http://www.eclipse.org/>) 所開發的軟體。

Java 和所有以 Java 為基礎的商標及標誌是 Oracle 及/或其子公司的商標或註冊商標。



產品編號:

(1P) P/N: