

9.0

ファイル転送管理

IBM

注記

本書および本書で紹介する製品をご使用になる前に、[945 ページの『特記事項』](#)に記載されている情報をお読みください。

本書は、IBM® MQ バージョン 9 リリース 0、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

お客様が IBM に情報を送信する場合、お客様は IBM に対し、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で情報を使用または配布する非独占的な権利を付与します。

© Copyright International Business Machines Corporation 2007 年, 2023.

目次

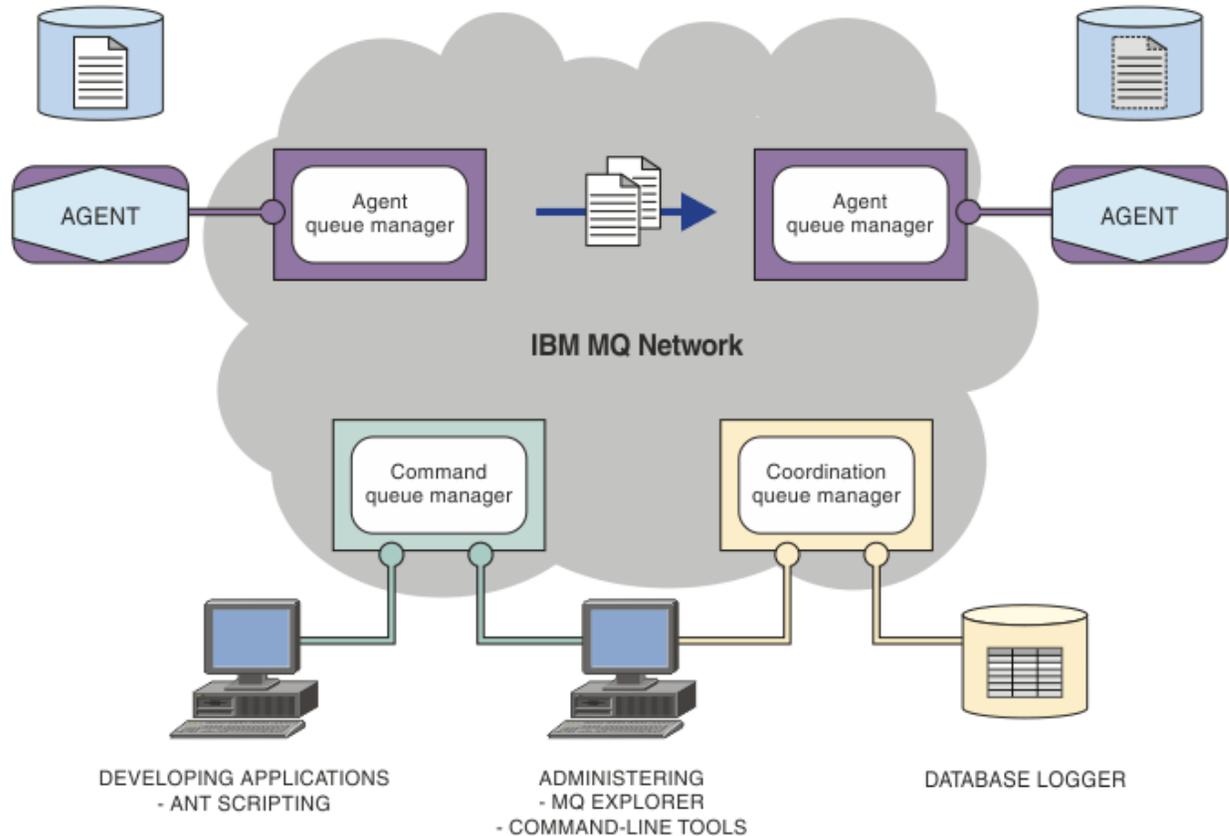
ファイル転送管理	5
MFT の概要.....	7
MFT の紹介.....	7
MFT と IBM MQ の連動について.....	9
MFT トポロジーの概要.....	10
MFTREST API の概要.....	11
IBM MQ 9.0 の MFT の新機能および変更点.....	11
MFT のマイグレーションの考慮事項.....	13
MFT シナリオの概要.....	29
MFT の一般的なトポロジー.....	29
基本サーバーの構成.....	33
MFT の保護.....	41
MFT と IBM MQ の接続認証.....	42
MFT のサンドボックス.....	45
MFT の SSL または TLS 暗号化の構成.....	51
クライアント・モードでチャンネル認証を使用してキュー・マネージャーに接続する操作.....	53
Connect:Direct ブリッジ・エージェントと Connect:Direct ノードの間の SSL または TLS の構成.....	54
MFT の構成.....	57
MFT 構成オプション (Multiplatforms).....	57
z/OS での MFT 構成オプション.....	59
Redistributable Managed File Transfer Agent の構成.....	59
MFT エージェントまたはロガーのコマンド・データ・セットの作成.....	63
Managed File Transfer for z/OS の構成.....	64
IBM i での MFT の構成.....	94
初めて使用する際の MFT の構成.....	95
MFT ロガーの構成.....	106
Connect:Direct ブリッジの構成.....	143
MSCS による MFT エージェントの構成.....	153
管理 MFT.....	154
MFT エージェントの開始.....	155
新規ファイル転送の開始.....	160
スケジュール済みファイル転送の作成.....	163
保留中のファイル転送の処理.....	164
ファイル転送のトリガー.....	165
進行中のファイル転送のモニター.....	166
「転送ログ」のファイル転送の状況の表示.....	168
MFT リソースのモニター.....	170
ファイル転送テンプレートの処理.....	195
ファイルからメッセージへのデータ転送.....	198
メッセージからファイルへのデータ転送.....	206
MFT エージェントのリスト.....	212
MFT エージェントの停止.....	212
プロトコル・ブリッジ.....	214
Connect:Direct ブリッジ.....	230
IBM Integration Bus からの MFT の操作.....	245
MFT のリカバリーと再始動.....	246
ファイル転送のリカバリーのタイムアウト・オプション.....	246
MFT 用のアプリケーションの開発.....	250
MFT で実行するプログラムの指定.....	250
MFT での Apache Ant の使用.....	252
ユーザー出口での MFT のカスタマイズ.....	256
エージェント・コマンド・キューにメッセージを PUT することによる MFT の制御.....	270

トラブルシューティング MFT.....	270
MFT の一般的なトラブルシューティング.....	270
Connect:Direct ブリッジのトラブルシューティング.....	345
MFT リファレンス.....	350
MFT セキュリティー・リファレンス.....	350
MFT のコマンドおよびプロセスと、その接続先のキュー・マネージャー.....	367
MFT コマンド.....	373
MFT 構成リファレンス.....	535
MFT 管理参照.....	624
MFT アプリケーション開発リファレンス.....	844
MFT の診断メッセージ.....	942
特記事項.....	945
プログラミング・インターフェース情報.....	946
商標.....	946

Managed File Transfer

Managed File Transfer は、ファイルのサイズや使用するオペレーティング・システムにかかわらず、システム間のファイル転送を管理下に置いて実行できます。監査も可能です。

Managed File Transfer を使用すれば、ファイル転送を管理し、確認し、保護するために、カスタマイズしたスケーラブルな自動化ソリューションを構築できます。Managed File Transfer によって、コストのかかる冗長性を除去し、保守コストを削減し、既存の IT 投資を最大限に活用することが可能になります。



この図は、単純な Managed File Transfer トポロジーを示しています。2つのエージェントがあり、それぞれが IBM MQ ネットワーク内の独自のエージェント・キュー・マネージャーに接続しています。図の一方にあるエージェントから IBM MQ ネットワークを経由して、図のもう一方にあるエージェントにファイルを転送します。さらに、IBM MQ ネットワークには、調整キュー・マネージャーとコマンド・キュー・マネージャーもあります。アプリケーションおよびツールは、これらのキュー・マネージャーに接続して、Managed File Transfer ネットワーク内の IBM MQ アクティビティを構成し、管理し、操作し、ログに記録します。

Managed File Transfer には、オペレーティング・システムと全体的なセットアップに応じて、4種類のインストール・オプションがあります。これらのオプションは Managed File Transfer Agent、Managed File Transfer Logger、Managed File Transfer Service、または Managed File Transfer Tools です。詳しくは、[Managed File Transfer 製品のオプションを参照してください](#)。

Managed File Transfer を使用して、次のタスクを実行できます。

- 管理対象ファイル転送を作成します。
 - Linux® または Windows プラットフォーム上の IBM MQ Explorer から新規ファイル転送を作成します。
 - サポートされているすべてのオペレーティング・システムで、コマンド・ラインから新しいファイル転送を作成できます。

- ファイル転送機能を Apache Ant ツールに組み込みます。
- エージェント・コマンド・キューにメッセージを PUT することによって、Managed File Transfer を制御するアプリケーションを作成します。
- ファイル転送は、後の時点で実行されるようにスケジュールに入れます。また、スケジュール済みファイル転送を、一定の範囲のファイル・システム・イベント (例えば、新規ファイルの作成など) に基づいてトリガーすることもできます。
- 例えばディレクトリーなどのリソースを継続的にモニターして、そのリソースの内容が事前定義の条件に一致した場合にタスクを開始します。このタスクは、ファイル転送、Ant スクリプト、または JCL ジョブにすることができます。
- IBM MQ キューとの間のファイル転送が可能です。
- FTP サーバー、FTPS サーバー、または SFTP サーバーとの間でファイル転送が可能です。
- Connect:Direct® ノードとの間のファイル転送が可能です。
- テキスト・ファイルとバイナリー・ファイルの両方の転送が可能です。テキスト・ファイルの場合、ソース・システムと宛先システムの間でコード・ページと行の終わり規則が自動的に変換されます。
- 転送は、SSL (Secure Socket Layer) ベース接続の業界規格を使用して保護できます。
- 転送の進行状況を表示することや、ネットワーク内のすべての転送に関する情報をログに記録することが可能です。
 - Linux または Windows のオペレーティング・システムで、IBM MQ Explorer から転送の進行状況を表示できます。
 - Linux または Windows のオペレーティング・システムで、IBM MQ Explorer を使用して、完了した転送の状況を確認できます。
 - Managed File Transfer のデータベース・ロガー機能を使用して、Db2® または Oracle データベースにログ・メッセージを保存します。

Managed File Transfer は、IBM MQ の基盤の上に構築されている製品であり、アプリケーション間の 1 回限りのメッセージ配信を確実に実行できるようになっています。IBM MQ のさまざまなフィーチャーを活用することができます。例えば、チャンネル圧縮を使用して、IBM MQ チャンネルを介してエージェント間で送信するデータを圧縮し、SSL チャンネルを使用して、エージェント間で送信するデータを保護することができます。ファイルは安全に転送され、ファイル転送を行う媒体となるインフラストラクチャーで発生した障害に対処する機能があります。ネットワーク障害が発生した場合、接続が復元されたときに、ファイル転送は中止された位置から再開します。

ファイル転送を既存の IBM MQ ネットワークと統合することにより、2 つの別個のインフラストラクチャーを保守して必要なリソースを浪費するということ避けられます。IBM MQ をまだ使用していないお客様の場合、Managed File Transfer をサポートする IBM MQ ネットワークを作成することにより、将来 SOA をインプリメントするためのバックボーンを構築することになります。IBM MQ を既に使用しているお客様の場合には、Managed File Transfer を使用することにより、既存の IBM MQ インフラストラクチャー (IBM MQ Internet Pass-Thru、IBM Integration Bus を含む) を活用できます。

Managed File Transfer は、以下のように、他の多くの IBM 製品との統合が可能です。

IBM Integration Bus

Managed File Transfer によって転送されたファイルを IBM Integration Bus フローの一部として処理できます。詳しくは、[245 ページの『IBM Integration Bus からの MFT の操作』](#)を参照してください。

IBM Sterling Connect:Direct

Managed File Transfer Connect:Direct ブリッジを使用して、既存の Connect:Direct ネットワークとの間でファイルを転送できます。詳しくは、[230 ページの『Connect:Direct ブリッジ』](#)を参照してください。

IBM Tivoli® Composite Application Manager

IBM Tivoli Composite Application Manager には、調整キュー・マネージャーにパブリッシュされた情報をモニターするために使用できるエージェントが用意されています。

関連概念

10 ページの『MFT トポロジーの概要』

Managed File Transfer エージェントが IBM MQ ネットワーク内の調整キュー・マネージャーとどのように接続されるかについての概要。

9 ページの『MFT と IBM MQ の連動について』

Managed File Transfer は、さまざまな方法で IBM MQ と対話します。

関連情報

[Managed File Transfer 製品のオプション](#)

MFT の概要

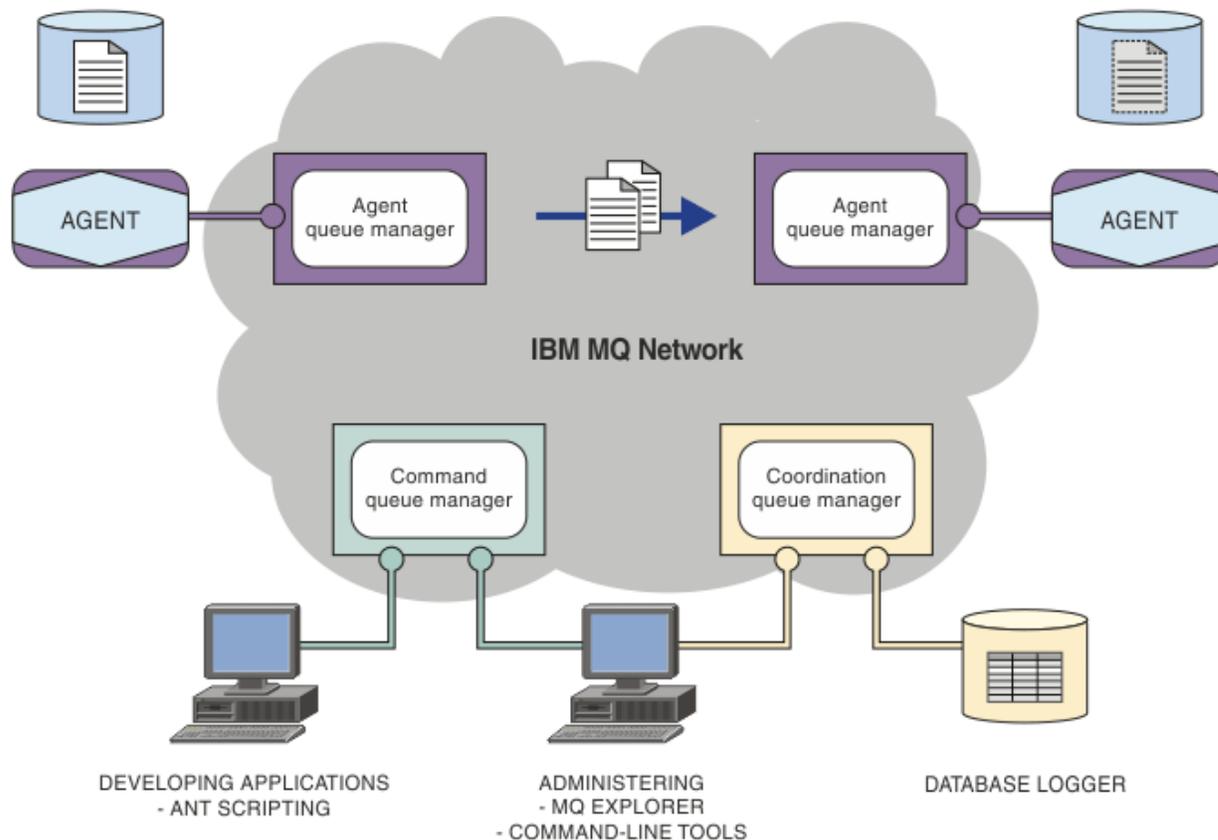
このセクションでは、Managed File Transfer を開始するにあたって使用できる初歩的な情報を提供します。

- [5 ページの『Managed File Transfer』](#)
- [Managed File Transfer 製品のオプション](#)
- [10 ページの『MFT トポロジーの概要』](#)
- [11 ページの『IBM MQ 9.0 の MFT の新機能および変更点』](#)

Managed File Transfer

Managed File Transfer は、ファイルのサイズや使用するオペレーティング・システムにかかわらず、システム間のファイル転送を管理下に置いて実行できます。監査も可能です。

Managed File Transfer を使用すれば、ファイル転送を管理し、確認し、保護するために、カスタマイズしたスケーラブルな自動化ソリューションを構築できます。Managed File Transfer によって、コストのかかる冗長性を除去し、保守コストを削減し、既存の IT 投資を最大限に活用することが可能になります。



この図は、単純な Managed File Transfer トポロジーを示しています。2つのエージェントがあり、それぞれが IBM MQ ネットワーク内の独自のエージェント・キュー・マネージャーに接続しています。図の一方にあるエージェントから IBM MQ ネットワークを経由して、図のもう一方にあるエージェントにファイルを転送します。さらに、IBM MQ ネットワークには、調整キュー・マネージャーとコマンド・キュー・マネージャーもあります。アプリケーションおよびツールは、これらのキュー・マネージャーに接続して、Managed File Transfer ネットワーク内の IBM MQ アクティビティを構成し、管理し、操作し、ログに記録します。

Managed File Transfer には、オペレーティング・システムと全体的なセットアップに応じて、4種類のインストール・オプションがあります。これらのオプションは Managed File Transfer Agent、Managed File Transfer Logger、Managed File Transfer Service、または Managed File Transfer Tools です。詳しくは、[Managed File Transfer 製品のオプション](#)を参照してください。

Managed File Transfer を使用して、次のタスクを実行できます。

- 管理対象ファイル転送を作成します。
 - Linux または Windows プラットフォーム上の IBM MQ Explorer から新規ファイル転送を作成します。
 - サポートされているすべてのオペレーティング・システムで、コマンド・ラインから新しいファイル転送を作成できます。
 - ファイル転送機能を Apache Ant ツールに組み込みます。
 - エージェント・コマンド・キューにメッセージを PUT することによって、Managed File Transfer を制御するアプリケーションを作成します。
 - ファイル転送は、後の時点で実行されるようにスケジュールに入れます。また、スケジュール済みファイル転送を、一定の範囲のファイル・システム・イベント (例えば、新規ファイルの作成など) に基づいてトリガーすることもできます。
 - 例えばディレクトリーなどのリソースを継続的にモニターして、そのリソースの内容が事前定義の条件に一致した場合にタスクを開始します。このタスクは、ファイル転送、Ant スクリプト、または JCL ジョブにすることができます。
 - IBM MQ キューとの間のファイル転送が可能です。
 - FTP サーバー、FTPS サーバー、または SFTP サーバーとの間でファイル転送が可能です。
 - Connect:Direct ノードとの間のファイル転送が可能です。
 - テキスト・ファイルとバイナリー・ファイルの両方の転送が可能です。テキスト・ファイルの場合、ソース・システムと宛先システムの間でコード・ページと行の終わり規則が自動的に変換されます。
 - 転送は、SSL (Secure Socket Layer) ベース接続の業界規格を使用して保護できます。
- 転送の進行状況を表示することや、ネットワーク内のすべての転送に関する情報をログに記録することが可能です。
 - Linux または Windows のオペレーティング・システムで、IBM MQ Explorer から転送の進行状況を表示できます。
 - Linux または Windows のオペレーティング・システムで、IBM MQ Explorer を使用して、完了した転送の状況を確認できます。
 - Managed File Transfer のデータベース・ロガー機能を使用して、Db2 または Oracle データベースにログ・メッセージを保存します。

Managed File Transfer は、IBM MQ の基盤の上に構築されている製品であり、アプリケーション間の 1 回限りのメッセージ配信を確実に実行できるようになっています。IBM MQ のさまざまなフィーチャーを活用することができます。例えば、チャンネル圧縮を使用して、IBM MQ チャンネルを介してエージェント間で送信するデータを圧縮し、SSL チャンネルを使用して、エージェント間で送信するデータを保護することができます。ファイルは安全に転送され、ファイル転送を行う媒体となるインフラストラクチャーで発生した障害に対処する機能があります。ネットワーク障害が発生した場合、接続が復元されたときに、ファイル転送は中止された位置から再開します。

ファイル転送を既存の IBM MQ ネットワークと統合することにより、2つの別個のインフラストラクチャーを保守して必要なリソースを浪費するということを避けられます。IBM MQ をまだ使用していないお客様の場合、Managed File Transfer をサポートする IBM MQ ネットワークを作成することにより、将来 SOA をインプリメントするためのバックボーンを構築することになります。IBM MQ を既に使用しているお客

様の場合には、Managed File Transfer を使用することにより、既存の IBM MQ インフラストラクチャー (IBM MQ Internet Pass-Thru、IBM Integration Bus を含む) を活用できます。

Managed File Transfer は、以下のように、他の多くの IBM 製品との統合が可能です。

IBM Integration Bus

Managed File Transfer によって転送されたファイルを IBM Integration Bus フローの一部として処理できます。詳しくは、[245 ページの『IBM Integration Bus からの MFT の操作』](#)を参照してください。

IBM Sterling Connect:Direct

Managed File Transfer Connect:Direct ブリッジを使用して、既存の Connect:Direct ネットワークとの間でファイルを転送できます。詳しくは、[230 ページの『Connect:Direct ブリッジ』](#)を参照してください。

IBM Tivoli Composite Application Manager

IBM Tivoli Composite Application Manager には、調整キュー・マネージャーにパブリッシュされた情報をモニターするために使用できるエージェントが用意されています。

関連概念

[10 ページの『MFT トポロジーの概要』](#)

Managed File Transfer エージェントが IBM MQ ネットワーク内の調整キュー・マネージャーとどのように接続されるかについての概要。

[9 ページの『MFT と IBM MQ の連動について』](#)

Managed File Transfer は、さまざまな方法で IBM MQ と対話します。

関連情報

[Managed File Transfer 製品のオプション](#)

MFT と IBM MQ の連動について

Managed File Transfer は、さまざまな方法で IBM MQ と対話します。

- Managed File Transfer は、各ファイルを 1 つ以上のメッセージに分割し、それらメッセージを IBM MQ ネットワークを介して送信することにより、エージェント・プロセス間でファイルを転送します。
- エージェント・プロセスは、IBM MQ ログに対する影響を最小化するために、非永続メッセージを使用してファイル・データを移動します。エージェント・プロセスは、相互にやり取りすることにより、ファイル・データが含まれるメッセージのフローを調整します。このようにして、ファイル・データが含まれているメッセージが IBM MQ 伝送キューに蓄積される状況が回避され、いずれかの非永続メッセージが送信されなかった場合にファイル・データが確実に再送信されるようになります。
- Managed File Transfer エージェントは、いくつかの IBM MQ キューを使用します。詳細については [621 ページの『MFT システム・キューおよびシステム・トピック』](#)を参照してください。
- これらのキューの一部は内部使用に限られていますが、エージェントは、読み取り先の特定のキューに送信される特殊形式のコマンド・メッセージの形で要求を受け入れることができます。コマンド行コマンドおよび IBM MQ Explorer・プラグインの両方は、IBM MQ メッセージをエージェントに送信し、対象となるアクションを実行するようにエージェントに指示します。このような方法でエージェントと対話する IBM MQ アプリケーションを作成できます。詳細については [270 ページの『エージェント・コマンド・キューにメッセージを PUT することによる MFT の制御』](#)を参照してください。
- Managed File Transfer エージェントは、その状態と、転送の進行状況と結果に関する情報を、調整キュー・マネージャーとして指定されている MQ キュー・マネージャーに送信します。この情報は、調整キュー・マネージャーによりパブリッシュされ、転送の進行状況のモニターまたは発生した転送の記録を行うアプリケーションによってサブスクライブできます。コマンド行コマンドおよび IBM MQ Explorer・プラグインの両方で、パブリッシュされた情報を利用できます。この情報を使用する IBM MQ アプリケーションを作成できます。情報のパブリッシュ先になるトピックの詳細については、[619 ページの『SYSTEM.FTE トピック』](#)を参照してください。
- Managed File Transfer のキー・コンポーネントは、IBM MQ キュー・マネージャーの機能を利用してメッセージのストア・アンド・フォワード処理を行います。これは、故障が発生した場合、インフラストラクチャーの中で影響を受けていない部分はファイルの転送を続行できることを意味します。このことは

調整キュー・マネージャーにも当てはまります。ストア・アンド・フォワードと永続サブスクリプションの組み合わせにより、調整キュー・マネージャーは、使用不可状態になっても行われたファイル転送に関する主要な情報を失うことなく対処できます。

MFT トポロジーの概要

Managed File Transfer エージェントが IBM MQ ネットワーク内の調整キュー・マネージャーとどのように接続されるかについての概要。

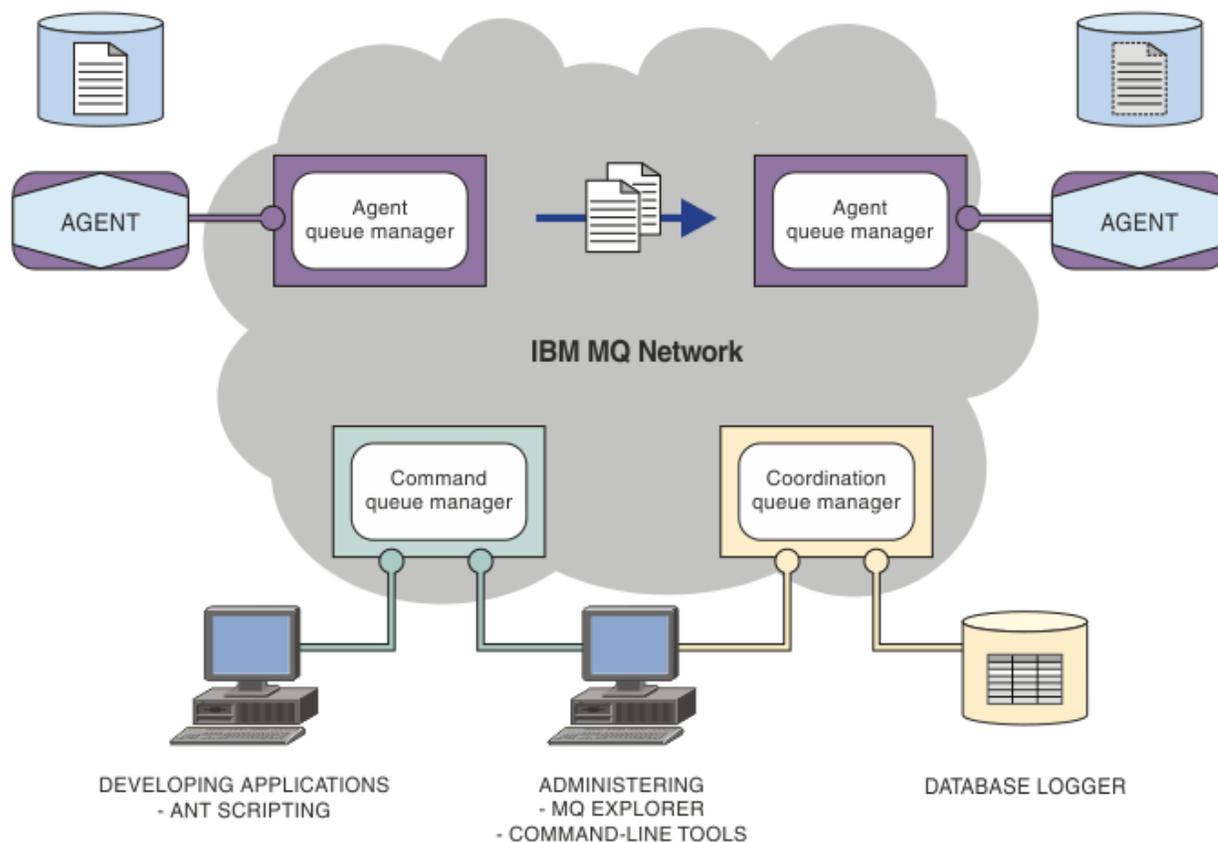
Managed File Transfer エージェントは、転送されるファイルを送受信します。エージェントはそれぞれ、関連付けられているキュー・マネージャーに対して独自の一連のキューを持ちます。エージェントはバインディング・モードまたはクライアント・モードのいずれかでキュー・マネージャーに接続されます。また、エージェントはそのキュー・マネージャーとして調整キュー・マネージャーを使用することもできます。

調整キュー・マネージャーは、監査およびファイル転送情報をブロードキャストします。調整キュー・マネージャーは、エージェント、転送状況、および転送監査の情報を収集するためのシングル・ポイントの役割を果たします。調整キュー・マネージャーが使用可能な状態になっていることは、転送を実行するための必要条件ではありません。調整キュー・マネージャーが一時的に使用できなくなった場合でも、転送処理は、通常どおり続行されます。監査メッセージと状況メッセージは、調整キュー・マネージャーが使用できるようになって、通常の処理が可能になるまで、エージェント・キュー・マネージャーに格納されます。

エージェントは調整キュー・マネージャーに登録され、詳細をそのキュー・マネージャーにパブリッシュします。このエージェント情報は、IBM MQ Explorer からの転送を開始できるようにするために Managed File Transfer プラグインが使用します。各種のコマンドも、エージェントの情報と状況を表示するために、調整キュー・マネージャーで収集されるエージェント情報を使用します。

転送状況と転送監査の情報は、調整キュー・マネージャーでパブリッシュされます。Managed File Transfer プラグインはその転送状況と転送監査の情報を使用して、IBM MQ Explorer から転送の進行状況をモニターします。監査能力を確保するために、調整キュー・マネージャーに格納される転送監査の情報を保存することもできます。

IBM MQ ネットワークに接続する場合にはこのコマンド・キュー・マネージャーが使用され、このコマンド・キュー・マネージャーが Managed File Transfer のコマンドを発行するときに接続されるキュー・マネージャーとなります。



関連概念

5 ページの『[Managed File Transfer](#)』

Managed File Transfer は、ファイルのサイズや使用するオペレーティング・システムにかかわらず、システム間のファイル転送を管理下に置いて実行できます。監査も可能です。

9 ページの『[MFT と IBM MQ の連動について](#)』

Managed File Transfer は、さまざまな方法で IBM MQ と対話します。

29 ページの『[MFT シナリオの概要](#)』

このセクションでは、一般的な Managed File Transfer トポロジーを、システムをセットアップしてテスト・メッセージを転送するシナリオとともに紹介します。

V 9.0.5 MFTREST API の概要

Managed File Transfer に対応した REST API の機能強化の概要。

IBM MQ 9.0.5 より、REST API に Managed File Transfer の一部のコマンド (転送のリスト表示や、ファイル転送エージェントの詳細情報の表示など) のサポートが追加されました。

詳細については、[REST API の機能強化](#)を参照してください。

IBM MQ 9.0 の MFT の新機能および変更点

IBM MQ 9.0 の主な新規および変更された Managed File Transfer 機能に関する情報へのリンク。

IBM MQ 9.0.0

V 9.0.0

- [V 9.0.0](#) Managed File Transfer のプロトコル・ブリッジ・エージェントのロギング機能拡張

- ▶ **V 9.0.0** [Managed File Transfer Web Gateway サポートの除去](#)

IBM MQ 9.0.x Continuous Delivery リリース

CD

- ▶ **Windows** ▶ **V 9.0.1** ▶ **Linux** [Managed File Transfer で構成可能な、停止した転送のリカバリー・タイムアウト](#)
- ▶ **Windows** ▶ **V 9.0.1** ▶ **Linux** [Managed File Transfer Agent 再配布可能パッケージ](#)
- ▶ **Windows** ▶ **V 9.0.1** ▶ **Linux** [リモート Managed File Transfer エージェントの **DISPLAY** CHSTATUS 出力値の変更](#)
- ▶ **Windows** ▶ **V 9.0.1** ▶ **Linux** [Managed File Transfer のメッセージからファイルへの転送の変更点](#)
- ▶ **V 9.0.1** [MFT の新しいエージェント・プロパティ additionalWildcardSandboxChecking](#)
- ▶ **Windows** ▶ **V 9.0.2** ▶ **Linux** [IBM MQ Explorer MFT プラグインで作成されるサブスクリプションの命名の改善](#)
- ▶ **Windows** ▶ **V 9.0.2** ▶ **Linux** [最後に報告された状況が UNKNOWN である場合の MFT エージェントに関する情報の改善](#)
- ▶ **z/OS** ▶ **V 9.0.2** ▶ **MQ Adv. VUE** [z/OS® 上の MFT エージェントで使用するための新しいエージェント・プロパティ **adminGroup**](#)
- ▶ **Windows** ▶ **z/OS** ▶ **Linux** ▶ **V 9.0.3** [Managed File Transfer のリソース・モニター・イベント・ロギング](#)
- ▶ **z/OS** 製品 ID (PID) の記録に関する変更点
- ▶ **z/OS** ▶ **V 9.0.3** ▶ **MQ Adv. VUE** [IBM MQ Advanced for z/OS, Value Unit Edition とのエージェントのリモート接続](#)
- ▶ **Windows** ▶ **z/OS** ▶ **Linux** [Managed File Transfer **fteCleanAgent** コマンドの動作の変更](#)
- ▶ **Windows** ▶ **Linux** [Managed File Transfer インストール・プロパティの **messagePublicationFormat=mixed** の動作の修正](#)
- ▶ **Windows** ▶ **z/OS** ▶ **Linux** ▶ **V 9.0.3** [Managed File Transfer の MQCSP 認証の有効化](#)
- ▶ **Windows** ▶ **Linux** ▶ **AIX** ▶ **V 9.0.4** [Managed File Transfer Logger をクライアント・モードで実行可能](#)
- ▶ **V 9.0.5** ▶ **Multi** [11 ページの『MFTREST API の概要』](#)

IBM MQ 9.0.0.x Long Term Support

LTS

- ▶ **V 9.0.0.1** [fteMigrateAgent コマンドの変更](#)
- ▶ **V 9.0.0.1** [MFT の新しいエージェント・プロパティ addCommandPathToSandbox](#)
- ▶ **V 9.0.0.1** [MFT の新しいエージェント・プロパティ additionalWildcardSandboxChecking](#)
- ▶ **z/OS** ▶ **V 9.0.0.1** [z/OS 上の MFT エージェントで使用するための新しいエージェント・プロパティ **adminGroup**](#)
- ▶ **V 9.0.0.2** [Managed File Transfer **fteCleanAgent** コマンドの動作の変更](#)

- [V 9.0.0.2](#) Managed File Transfer での MQCSP 認証の有効化
- [V 9.0.0.2](#) Managed File Transfer installation.properties ファイルの新規 **messagePublicationFormat** プロパティ
- [V 9.0.0.2](#) メッセージからファイルへの転送のために復元された、転送 ID と groupId 属性の値の比較チェック
- [V 9.0.5](#) fteSetAgentTraceLevel コマンドおよび fteSetLoggerTracelevel コマンドで、コンマ区切りリストを使用できる

関連情報

[V 9.0.0](#) IBM MQ 9.0.0 の新機能と変更点

[CD](#) IBM MQ 9.0.x の Continuous Delivery の新機能と変更点

[LTS](#) IBM MQ 9.0.x の Long Term Support の変更点

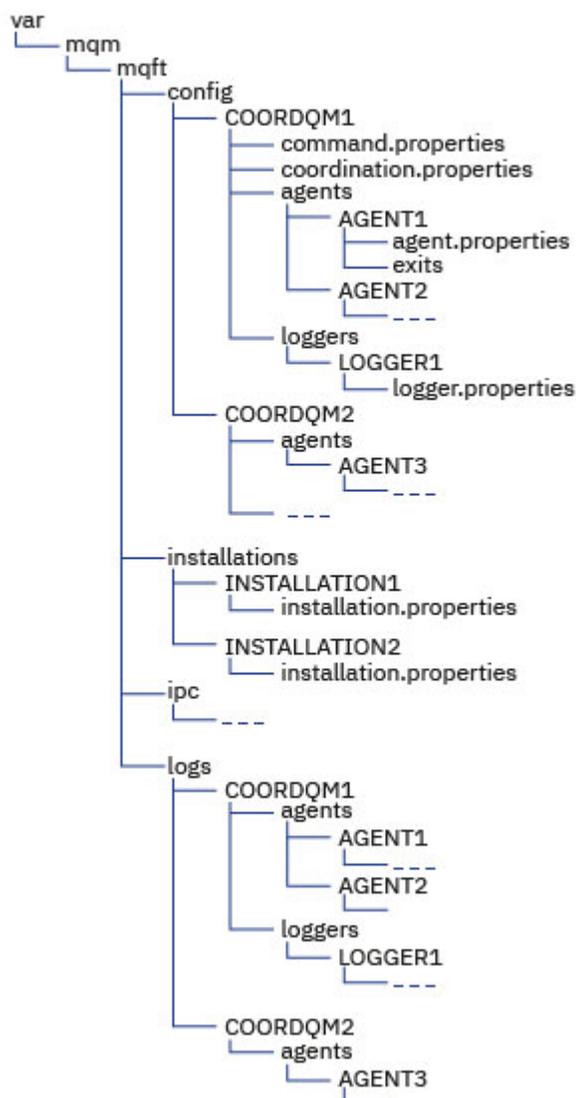
MFT のマイグレーションの考慮事項

Managed File Transfer のマイグレーションを計画している場合は、以下の情報を確認してください。

MFT でのインストール直後の構成レイアウト

この図は、IBM MQ の場合のインストール直後の構成レイアウトを示しています。

IBM MQ Managed File Transfer configuration



デフォルトの構成ディレクトリー:

情報は、config、installations、ipc、および logs の4つの別個のサブディレクトリーに分割されます。

デフォルトの製品ルート・ディレクトリー (MQ_DATA_PATH) は以下のとおりです。

- UNIX システム: /var/mqm
- Linux システム: /var/mqm
- Windows: 構成ディレクトリーの場所は、基本 IBM MQ のインストール済み環境の場所に応じて異なります。基本インストール済み環境のデフォルトの場所は以下のとおりです。
 - 32 ビット: C:\Program Files (x86)\IBM\WebSphere® MQ
 - 64 ビット: C:\Program Files\IBM\MQ

構成サブディレクトリーは以下のとおりです。

- MQ_DATA_PATH/mqft/config ディレクトリーには、Managed File Transfer プロセスに対して読み取り専用の構成部分が含まれています。例えば、agent.properties や command.properties です。

- `MQ_DATA_PATH/mqft/installations` ディレクトリーには、インストール済み環境ごとの構成情報が含まれます。このディレクトリーの内容は、`wmqfte.properties` ファイルの内容に相当します。
- `MQ_DATA_PATH/mqft/ipc` ディレクトリーには、Managed File Transfer コンポーネント間の通信に内部的に使用される IPC リソースが含まれます。UNIX および Linux システムにのみ適用されます。
- `MQ_DATA_PATH/mqft/logs` ディレクトリーには、Managed File Transfer プロセスによる書き込みが行われる構成の部分が含まれます。例えば、トレース情報やログ・ファイルです。

installation.properties ファイル

Linux **UNIX** UNIX and Linux システムの場合、デフォルトの場所は `MQ_DATA_PATH/mqft/installations/installation_name` になります。

Windows Windows では、デフォルトのロケーションは `MQ_DATA_PATH\mqft\installations\installation_name` です。

logger.properties ファイル

このファイルには、スタンドアロン・ファイル・ロガー、スタンドアロン・データベース・ロガー、および Java EE データベース・ロガーのプロパティ情報が組み込まれています。

デフォルトの場所は `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` です。

セキュリティの変更

IBM WebSphere MQ 7.5 以降では、以下のリストにある **fte** コマンドは、管理者 (mqm グループのメンバー) であるユーザーのみが実行できます。

- [383 ページの『fteChangeDefaultConfigurationOptions』](#)
- [388 ページの『fteCreateAgent \(MFT エージェントの作成\)』](#)
- [393 ページの『fteCreateBridgeAgent \(MFT プロトコル・ブリッジ・エージェントの作成および構成\)』](#)
- [402 ページの『fteCreateCDAgent \(Connect:Direct ブリッジ・エージェントの作成\)』](#)
- [408 ページの『fteCreateLogger \(MFT ファイルまたはデータベース・ロガーの作成\)』](#)
- [462 ページの『fteDeleteAgent』](#)
- [465 ページの『fteDeleteLogger』](#)
- [483 ページの『fteMigrateAgent: FTE 7.0 エージェントの MQ 7.5 以降へのマイグレーション』](#)
- [485 ページの『fteMigrateConfigurationOptions: FTE 7.0 構成の MQ 7.5 以降へのマイグレーション』](#)
- [487 ページの『fteMigrate ロガー: FTE 7.0 データベース・ロガーの MQ 7.5 以降へのマイグレーション』](#)
- [490 ページの『fteModifyAgent \(Windows サービスとしての MFT エージェントの実行\)』](#)
- [492 ページの『fteModifyLogger \(Windows サービスとしての MFT ロガーの実行\)』](#)
- [512 ページの『fteSetupCommands: MFT command.properties ファイルの作成』](#)
- [514 ページの『fteSetupCoordination』](#)

Multi マルチプラットフォームで IBM WebSphere MQ 7.5 以降を使用する場合、エージェント・プロセスを実行しているユーザーのみが **fteSetAgentTraceLevel** コマンドを実行できます。

z/OS **V 9.0.2** z/OS の場合、IBM MQ 9.0.2 の Continuous Delivery では、以下のいずれかの方法で **fteSetAgentTraceLevel** コマンドを実行できます。

- エージェント・プロセスを実行しているものと同じユーザー ID。
- エージェント・プロパティ **adminGroup** で指定されたグループのメンバー。

V 9.0.0.1 これは、IBM MQ 9.0.0 Fix Pack 1 以降の Long Term Support リリースにも適用されます。

z/OS の場合、Continuous Delivery from IBM MQ 9.0.2 では、**fteShowAgentDetails** コマンドを実行してローカル・エージェントに関する情報を表示すると、以下のいずれかの方法でコマンドを実行できます。

- エージェント・プロセスを実行しているものと同じユーザー ID。
- エージェント・プロパティ **adminGroup** で指定されたグループのメンバー。

V 9.0.0.1

これは、IBM MQ 9.0.0 Fix Pack 1 以降の Long Term Support リリースにも適用されます。

詳しくは、**adminGroup** プロパティ (550 ページの『MFT agent.properties ファイル』) を参照してください。

IBM MQ 8.0 でのセキュリティーの変更

IBM WebSphere MQ 7.0 上で Managed File Transfer を動作させていて、IBM MQ 8.0 に移行する場合、MQMFTCredentials.xml ファイルのユーザー ID 情報はキュー・マネージャーに渡されますが、アクションは行われません。

ユーザー ID とパスワード情報の受け渡しは、IBM MQ 8.0 だけでサポートされているからです。

commandPath とエージェント・サンドボックス

IBM MQ 8.0 以降では、エージェント・サンドボックスを使用してエージェントが構成されている場合に、エージェント・プロパティ **commandPath** が設定されていると、エージェントの開始時に、**commandPath** で指定したディレクトリーが拒否パスに自動的に追加されます。エージェント・サンドボックスを使用して構成されていないエージェントで **commandPath** プロパティが設定されている場合、新規サンドボックスが自動的にセットアップされ、エージェントの開始時に、**commandPath** で指定したディレクトリーが拒否ディレクトリーに追加されます。

エージェント・サンドボックスを使用して構成されていないエージェントで **commandPath** プロパティが設定されている場合、新規サンドボックスが自動的にセットアップされ、エージェントの開始時に、**commandPath** で指定したディレクトリーが拒否ディレクトリーに追加されます。

commandPath プロパティについて詳しくは、[365 ページの『commandPath MFT プロパティ』](#)および [550 ページの『MFT agent.properties ファイル』](#) を参照してください。

commandPath とユーザー・サンドボックス

IBM MQ 8.0 以降では、エージェントが 1 つ以上のユーザー・サンドボックスを使用して構成されている場合、エージェント・プロパティ **commandPath** が設定されていると、**commandPath** で指定されたディレクトリー (およびそのすべてのサブディレクトリー) が、エージェントの開始時に、**<exclude>** エlement として、各ユーザー・サンドボックスの **<read>** Element と **<write>** Element に自動的に追加されます。

commandPath プロパティについて詳しくは、[365 ページの『commandPath MFT プロパティ』](#)および [550 ページの『MFT agent.properties ファイル』](#) を参照してください。

Windows サービスとして実行される MFT エージェントの IBM WebSphere MQ 7.5 から IBM MQ 9.0 へのマイグレーション

IBM WebSphere MQ 7.5、IBM MQ 8.0、および IBM MQ 9.0 の間では、Windows プラットフォーム上でデフォルトの IBM MQ インストール・パスが変更されています。

キュー・マネージャーを IBM WebSphere MQ 7.5 から IBM MQ 8.0 または IBM MQ 9.0 にマイグレーションする場合、キュー・マネージャーと同じシステムで実行されているすべてのアプリケーションを再構成して、新しいインストール・ロケーションから IBM MQ ライブラリーをロードする必要があります。これには、Windows サービスとして実行されるすべての IBM MQ Managed File Transfer エージェントが含まれます。

以下の手順を実行します。

1. IBM WebSphere MQ 7.5 キュー・マネージャーに関連付けられているエージェントを停止します。詳しくは、[MFT エージェントの停止](#)を参照してください。例えば、以下のコマンドを発行します。

```
fteStopAgent <agent_name>
```

2. エージェントを変更して、エージェントのサービス定義を削除します。詳しくは、[Windows サービスとしての MFT エージェントの実行](#)を参照してください。

例えば、以下のコマンドを発行します。

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

3. 次に、**setmqm** コマンドを使用して、エージェント・キュー・マネージャーを IBM WebSphere MQ 7.5 から新しいバージョンにマイグレーションします。**setmqm** コマンドについて詳しくは、[キュー・マネージャーとインストール済み環境の関連付け](#)を参照してください。

4. 最後に、**fteModifyAgent** コマンドを使用して、エージェントを変更しエージェントが Windows サービスとして再実行されるように再構成します。

以下に例を示します。

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

IBM WebSphere MQ File Transfer Edition 7.0 エージェントの IBM WebSphere MQ 7.5 以降へのマイグレーション

fteMigrateAgent コマンドを使用して、IBM WebSphere MQ File Transfer Edition 7.0 エージェントを IBM WebSphere MQ 7.5 以降にマイグレーションします。エージェントが Windows サービスとして実行されるように構成されている場合、このタスクのオプションのステップを実行する必要があります。

始める前に

V 9.0.0 IBM MQ 9.0 から、Managed File Transfer は Web エージェントをサポートしなくなりました。**fteMigrateAgent** コマンドを使用して Web エージェントを以前のリリースから IBM MQ 9.0 にマイグレーションしようとする、Web エージェントのマイグレーションがサポートされていないことを示すエラー・メッセージが表示されます。

IBM WebSphere MQ File Transfer Edition エージェントの一部としてインストールされている IBM WebSphere MQ File Transfer Edition のインスタンスを識別します。

IBM i このトピックの情報は、IBM i には適用されません。IBM i を使用している場合は、IBM MQ 8.0 で Managed File Transfer を使用する前に、キュー・マネージャーを IBM MQ 8.0 にマイグレーションします。

このタスクについて

IBM WebSphere MQ 7.5 以降にマイグレーションするには、まずエージェントを停止してから、キュー・マネージャー・インストール済み環境を IBM WebSphere MQ 7.5 以降にマイグレーションし、追加で File Transfer コンポーネントを選択します。キュー・マネージャーがマイグレーションされたら、**fteMigrateAgent** コマンドを使用して、エージェントの構成を IBM WebSphere MQ 7.5 以降のインストール済み環境にマイグレーションできます。マイグレーションしたエージェントを IBM WebSphere MQ 7.5 以降のキュー・マネージャーを使用して開始します。

エージェントが IBM WebSphere MQ バインディング・モードを使用してエージェントのキュー・マネージャーに接続しており、そのキュー・マネージャーが IBM WebSphere MQ 7.0.1 Fix Pack 6 以降である場合は、複数のインストール済み環境を使用して以下にリストするステップに従うことで、並列マイグレーションを実行することもできます。並列マイグレーションを実行しない場合は、上記で説明しているステップを実行して、キュー・マネージャーとエージェントをマイグレーションしてください。

エージェントが IBM WebSphere MQ バインディング・モードを使用してエージェントのキュー・マネージャーに接続しており、そのキュー・マネージャーが IBM WebSphere MQ 7.0.1 Fix Pack 5 以前である場合

は、このインストール済み環境を IBM WebSphere MQ 7.0.1 Fix Pack 6 にマイグレーションして並列マイグレーションを実行できるようにするか、あるいはこのインストール済み環境を直接 IBM WebSphere MQ 7.5 にマイグレーションする必要があります。

エージェントがネットワーク経由で IBM WebSphere MQ クライアントとしてキュー・マネージャーに接続している場合は、ステップ 18 ページの『7』を実行するだけで、エージェントを IBM WebSphere MQ 7.5 以降にマイグレーションできます。

エージェントが Windows サービスとして構成されている場合は、**-f** パラメーターを指定してコマンドを実行する必要があります。詳しくは、483 ページの『[fteMigrateAgent: FTE 7.0 エージェントの MQ 7.5 以降へのマイグレーション](#)』を参照してください。

手順

1. MQ Server および File Transfer コンポーネントを選択し、既存のキュー・マネージャー・インストール済み環境と併せて IBM WebSphere MQ 7.5 以降をインストールします。
2. IBM WebSphere MQ File Transfer Edition 7.0.x エージェントを停止します。
3. キュー・マネージャーを IBM WebSphere MQ 7.5 以降のインストール済み環境に移動します。
4. **fteMigrateAgent** コマンドを使用して、エージェント構成を IBM WebSphere MQ File Transfer Edition 7.0.x インストール済み環境から IBM WebSphere MQ 7.5 以降に統合されたエージェント機能にマイグレーションします。
5. オプション: IBM WebSphere MQ File Transfer Edition 7.0.x エージェントが Windows サービスとして構成されている場合は、以下のオプションのステップを実行します。
 - a) **fteModifyAgent** コマンドを使用して、エージェントが Windows サービスではなくなるように再構成します。
 - b) IBM WebSphere MQ 7.5 以降のインストール済み環境から **fteModifyAgent** コマンドを使用して、IBM WebSphere MQ 7.5 以降のバージョンのエージェントを Windows サービスとして構成します。
6. IBM WebSphere MQ 7.5 以降のエージェントを開始します。
7. オプション: ネットワーク経由でキュー・マネージャーに接続するエージェントをクライアントとしてマイグレーションするには、以下のステップを実行します。
 - a) IBM WebSphere MQ 7.5 以降をシステムにインストールします。
 - b) **fteMigrateAgent** コマンドを使用して、エージェント構成を WMQFTE 7.0.x インストール済み環境から IBM WebSphere MQ 7.5 以降に統合されたエージェント機能にマイグレーションします。
 - c) IBM WebSphere MQ File Transfer Edition 7.0.x エージェントを停止します。
 - d) IBM WebSphere MQ File Transfer Edition 7.0.x エージェントが Windows サービスとして構成されている場合は、IBM WebSphere MQ File Transfer Edition 7.0.x インストール済み環境から **fteModifyAgent** コマンドを使用して、Windows サービスとして実行されないようにエージェントを再構成します。
 - e) IBM WebSphere MQ File Transfer Edition 7.0.x エージェントが Windows サービスとして構成されている場合は、IBM WebSphere MQ 7.5 以降のインストール済み環境から **fteModifyAgent** コマンドを使用して、IBM WebSphere MQ 7.5 以降のエージェントを Windows サービスとして構成します。
 - f) IBM WebSphere MQ 7.5 以降のエージェントを開始します。

FTE 7.0 データベース・ロガーの MQ 7.5 以降へのマイグレーション

fteMigrateLogger コマンドを使用して、スタンドアロン IBM WebSphere MQ File Transfer Edition 7.0 データベース・ロガーを IBM WebSphere MQ 7.5 以降にマイグレーションします。データベース・ロガーを Windows サービスとして実行するように構成した場合は、追加のマイグレーション・ステップを実行する必要があります。

始める前に

スタンドアロン・データベース・ロガーのインスタンスを識別します。データベース・ロガー・インスタンスが、受信した最新バージョンの転送ログ・メッセージを正しく処理できるようにするために、Managed File Transfer ネットワークの他の部分より先にデータベース・ロガー・インスタンスを更新する必要があります。

このタスクについて

直接 IBM WebSphere MQ 7.5 以降にマイグレーションするには、まずロガーを停止してから、このインストール済み環境を必要なバージョンにマイグレーションします。このマイグレーションが完了した後に、**fteMigrateLogger** コマンドを使用してデータベース・ロガー構成を IBM MQ の必要なバージョンにマイグレーションします。

キュー・マネージャーが IBM WebSphere MQ File Transfer Edition 7.0.1 Fix Pack 6 以降である場合は、以下の手順で説明するように、複数のインストール済み環境を使用して並列マイグレーションを実行することもできます。並列マイグレーションを実行したくない場合は、上記で説明しているステップを実行して、キュー・マネージャーとロガーをマイグレーションしてください。

データベース・ロガーが IBM WebSphere MQ File Transfer Edition 7.0.1 Fix Pack 5 以前のキュー・マネージャーに接続している場合は、最初にこのインストール済み環境を IBM WebSphere MQ File Transfer Edition 7.0.1 Fix Pack 6 にマイグレーションして並列マイグレーションを実行できるようにするか、あるいはこのインストール済み環境を直接 IBM WebSphere MQ 7.5 以降にマイグレーションしてから、追加で File Transfer コンポーネントを選択する必要があります。

データベース・ロガーが Windows サービスとして構成されている場合は、**-f** パラメーターを指定して **fteMigrateLogger** コマンドを実行する必要があります。詳しくは、[487 ページの『fteMigrate ロガー: FTE 7.0 データベース・ロガーの MQ 7.5 以降へのマイグレーション』](#)を参照してください。

手順

1. MQ Server および Managed File Transfer Service コンポーネントを選択し、既存の IBM WebSphere MQ File Transfer Edition 7.0 インストール済み環境と併せて IBM WebSphere MQ 7.5 以降をインストールします。
2. IBM WebSphere MQ File Transfer Edition 7.0 データベース・ロガーを停止します。
3. キュー・マネージャーを IBM WebSphere MQ 7.5 以降のインストール済み環境に移動します。
4. **fteMigrateLogger** コマンドを使用して、データベース・ロガー構成を IBM WebSphere MQ File Transfer Edition 7.0 インストール済み環境から IBM WebSphere MQ 7.5 以降のデータベース・ロガーにマイグレーションします。
5. データベース・ロガーの開始に必要な表を作成するため、`MQ_INSTALLATION_PATH/mqft/sql` 内の SQL スクリプトを実行します。製品レベルが上がるたびに 1 つずつスクリプトがあり、適切なスクリプトをすべて順番どおりに実行します。

アップグレード sql スクリプトをバージョン順に実行する必要があります。まず、IBM WebSphere MQ File Transfer Edition または Managed File Transfer の現行レベルから開始します。*** が Db2、Oracle、または z/OS の場合、使用可能なスクリプトは次のようになります。

- `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`

例えば、IBM WebSphere MQ File Transfer Edition 7.0.3 を使用していて、IBM MQ 8.0.0.0 にマイグレーションする場合は、703-704、704-750、750-7502、および 7502-800 を実行します。

注: z/OS z/OS 上で、ftelog_tables_zos_704-800.sql を使用して直接 704 から 800 へアップグレード可能です。

ftelog_tables_***_704-750.sql までのスクリプトは、IBM MQ 8.0.0.0 に含まれており、MQ_INSTALL_ATIION_PATH/mqft/sql ディレクトリー内にあります。750 より後のアップグレードは IBM MQ 8.0.0.0 に含まれていません。これらがディレクトリー内に存在しない場合は、[APAR IT01841](#) で指定されているリンクからダウンロードできます。

6. オプション: IBM WebSphere MQ File Transfer Edition 7.0 データベース・ロガーが Windows サービスとして構成されている場合は、以下のステップを実行します。
 - a) [fteModifyDatabaseLogger](#) コマンドを使用して、データベース・ロガーが Windows サービスでなくなるように再構成します。
 - b) [fteModifyLogger](#) コマンドを使用して、IBM WebSphere MQ 7.5 以降のロガーで Managed File Transfer を再構成し、Windows サービスになるようにします。
7. IBM WebSphere MQ 7.5 以降のデータベース・ロガーで Managed File Transfer を開始します。

タスクの結果

これで、データベース・ロガーが IBM WebSphere MQ File Transfer Edition 7.0 から IBM WebSphere MQ 7.5 以降にマイグレーションされました。

Windows、UNIX、Linux で使用する Db2 のログ・データベースのページ・サイズの拡張

ご使用のデータベースが Db2 Windows、UNIX、または Linux システム上にあり、8 KB 未満のページ・サイズでログ・データベースを作成した場合は、IBM WebSphere MQ File Transfer Edition 7.0.3 以降の表にマイグレーションする前に、データベースのページ・サイズを増やす必要があります。

手順

1. まだデータベース・ロガーを停止していない場合は、[fteStopDatabaseLogger](#) コマンドを使用してデータベース・ロガーを停止します。
2. Db2 で提供されているツールを使用してログ・データベースをバックアップします。
3. Db2 **export** コマンドを使用して、データをログ・データベース表からディスク上のファイルに転送します。

注: ラージ・オブジェクトを含む表の場合は、ラージ・オブジェクト・ファイルを指定する必要があります。該当する表は、CALL_RESULT および METADATA です。
4. ログ・データベース表をドロップします。
5. ページ・サイズが少なくとも 8 KB のテーブル・スペースと、それに関連する、ページ・サイズが少なくとも 8 KB のバッファ・プールを作成します。

新しいテーブル・スペースに名前を指定します。例: FTE8KSPACE。
6. コマンドが新しいテーブル・スペースに表を作成するように [ftelog_tables_db2.sql](#) ファイルを編集します。

[ftelog_tables_db2.sql](#) ファイルでテキスト IN "USERSPACE1" の出現箇所をすべて IN "new_tablespace_name" に変更します。例えば、IN "USERSPACE1" を IN "FTE8KSPACE" に変更します。
7. ご使用のデータベースに対して [ftelog_tables_db2.sql](#) ファイルで SQL コマンドを実行します。
8. Db2 **load** コマンドを使用して、エクスポートされたデータを新しい表に転送します。

注:

- 入力ファイルにある列名に基づいて列名をマップします。「入力列名」と「ターゲット列名」が、構造を変更したこれらの表で一致していることを確認します。

- MONITOR と TRANSFER を除くすべての表の ID 列で IDENTITY OVERRIDE 動作を指定する必要があります。この動作を指定することにより、ロード操作中に行 ID が再生成されないようになります。
9. 以下の表に対し、この順番で、保全性状況値 **immediate** および **checked** を指定して Db2 **set integrity** コマンドを実行します。
- CALL_ARGUMENT
 - MONITOR
 - MONITOR_ACTION
 - MONITOR_EXIT_RESULT
 - MONITOR_METADATA
 - SCHEDULE_ACTION
 - SCHEDULE
 - SCHEDULE_ITEM
 - TRANSFER
 - TRANSFER_CALLS
 - TRANSFER_EVENT
 - TRANSFER_ITEM
 - TRANSFER_STATS
 - TRIGGER_CONDITION
10. 生成される ID 列を含む表で、既存の最大 ID 値より 1 だけ大きい値から開始するように ID 生成機能を設定します。
- 以下の表には、生成される ID 列が含まれます。
- AUTH_EVENT
 - CALL
 - CALL_ARGUMENT
 - CALL_RESULT
 - FILE_SPACE_ENTRY
 - METADATA
 - MONITOR_ACTION
 - MONITOR_EXIT_RESULT
 - MONITOR_METADATA
 - SCHEDULE
 - SCHEDULE_ACTION
 - SCHEDULE_ITEM
 - SCHEDULE_SPEC
 - TRANSFER_CALLS
 - TRANSFER_CD_NODE
 - TRANSFER_CORRELATOR
 - TRANSFER_EVENT
 - TRANSFER_EXIT
 - TRANSFER_ITEM
 - TRANSFER_ITEM_ATTRIBUTES
 - TRANSFER_STATS
 - TRIGGER_CONDITION

これらの表の生成される ID の値を正しい値にするには、各表について以下のステップを実行します。

- a) 既存データ内の最大 ID 値を判別します。

次の SQL 文を実行してこの値を見つけることができます。

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

このコマンドで戻される値が、指定された表の中で最大の既存 ID です。

- b) 表を変更して、前のステップで戻された値より 1 だけ大きい新規の値から開始するように、ID 生成機能を設定します。

次の SQL 文を実行して、この値を設定することができます。

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

関連タスク

[142 ページの『スタンドアロン・データベース・ロガーから MFT 用の Java EE データベース・ロガーへのマイグレーション』](#)

スタンドアロン・データベース・ロガーから Java EE データベース・ロガーにマイグレーションできます。スタンドアロン・データベース・ロガーを停止して、Java EE データベース・ロガーをインストールする必要があります。ログ項目の消失または重複を回避するには、スタンドアロン・データベース・ロガーを停止する前に、SYSTEM.FTE トピックへのメッセージのパブリッシュを停止し、JEE データベース・ロガーをインストールした後にこれを再始動する必要があります。マイグレーション前にデータベースをバックアップしてください。

[22 ページの『z/OS 上の Db2 上のデータベース表の IBM MQ 8.0 へのマイグレーション』](#)

ご使用のデータベースが z/OS システム上の Db2 である場合は、以下のステップを実行して、IBM WebSphere MQ File Transfer Edition 7.0.3 から IBM WebSphere MQ File Transfer Edition 7.0.4 にマイグレーションし、IBM WebSphere MQ File Transfer Edition 7.0.4 から IBM MQ 8.0 の Managed File Transfer にマイグレーションする必要があります。Db2 表は以前のリリースと構造が異なっています。例えば、一部の表には新しい列が含まれており、一部の可変文字列はより大きくすることができるため、以前のリリースからの表を Db2 形式にマイグレーションする必要があります。

z/OS 上の Db2 上のデータベース表の IBM MQ 8.0 へのマイグレーション

ご使用のデータベースが z/OS システム上の Db2 である場合は、以下のステップを実行して、IBM WebSphere MQ File Transfer Edition 7.0.3 から IBM WebSphere MQ File Transfer Edition 7.0.4 にマイグレーションし、IBM WebSphere MQ File Transfer Edition 7.0.4 から IBM MQ 8.0 の Managed File Transfer にマイグレーションする必要があります。Db2 表は以前のリリースと構造が異なっています。例えば、一部の表には新しい列が含まれており、一部の可変文字列はより大きくすることができるため、以前のリリースからの表を Db2 形式にマイグレーションする必要があります。

このタスクについて

IBM MQ 8.0 の Managed File Transfer は、非常に大きなファイルの転送をサポートします。この場合、ファイルのサイズは整数 (32 ビット) で保管できるサイズより大きくなります。BIGINT (64 ビット) 数値を使用する定義があります。prod/mqf/sql ディレクトリーにある ftelog_tables_zos.sql という整数定義か、ftelog_tables_zos_bigint.sql という BIGINT 定義のいずれかを選んで使用することができます。

BIGINT データ・タイプは、IBM WebSphere MQ File Transfer Edition 7.0.4 Fix Pack 3、または IBM WebSphere MQ 7.5.0 Fix Pack 1 以降の Managed File Transfer で使用できます。

BIGINT データ・タイプを使用可能にするには、Db2 9.1 以降を使用している必要があります。INTEGER データ型は、転送されるファイルのサイズおよび各転送に関連付けられているテーブル ID を示すフィールドに使用します。サイズが 2 GB を超えるファイルの転送をログに記録するには、またはデータベースに 2,147,483,648 回を超える転送を保管するには、BIGINT SQL ファイルを使用する必要があります。

従う必要があるプロセスの概要を以下のリストに示します。

1. IBM WebSphere MQ 7.0 の既存の表があります。これらの表にはスキーマ (FTELOG など) があります。
2. 別のスキーマ名 (FTEV8 など) を使用して IBM MQ 8.0 表を作成します。これにより、FTELOG.table から FTE8.table にデータをコピーできます。
3. 新しい表にデータをコピーします。
4. 新しい表に生成 ID 値を設定します。
5. **fteMigrateLogger** コマンドを実行して、プロパティー・ファイルをディレクトリー構造の新しい場所に移動します。
6. ロガー・プロパティー・ファイルを編集して、新しいスキーマ (FTEV8) を指定します。
7. 既存のロガー JCL を編集して、IBM MQ 8.0 Managed File Transfer ライブラリーを使用します。
8. ロガーを開始します。
9. ロガーが機能するようになると、FTELOG 表を削除できます。

以下の説明では、Managed File Transfer 製品が USS の /HMF8800 ディレクトリーにインストールされています。

手順

1. データベース・ロガーをまだ停止していない場合は、USS または **P loggerjob** で **fteStopDatabaseLogger** コマンドを使用してデータベース・ロガーを停止します。
2. コマンド **ls /HMF8800/mqft/sql** を発行して、ディレクトリー内の SQL ファイルをリストします。BIGINT 数値を使用する場合は、**ftelog_tables_zos_bigint.sql** をホーム・ディレクトリーにコピーします。使用しない場合は、**ftelog_tables_zos.sql** をホーム・ディレクトリーにコピーします。
3. ホーム・ディレクトリーに移動したファイルを以下のように編集します。
 - a) **ftelog** を新しい表のスキーマ名に変更します。
 - b) 各索引の名前が固有であることを確認します。
これを行うには、編集セッションで以下のようにします。
 - i) すべての行を除外します。
 - ii) 'CREATE UNIQUE INDEX ' ALL を検索します。
 - iii) **_KEY _K8Y ALL NX** を変更します。
4. ファイルを確認して、すべてのステートメントが 71 桁目までに入っていることを確認します。ステートメントが 71 桁目までに入っていない場合、71 桁目より前で行を分割します。
5. このファイルを SQL への入力として使用したり、PDS にコピーしたりすることもできます。これを行うには、PDS を編集して **copy** コマンドを使用し、ディレクトリーとファイル名を指定します。
6. 従うべきサイトの標準があるかもしれないので、Db2 管理者とともに定義を確認します。
7. 以下を実行します。
 - a) /HMF8800/mqft/sql/ftelog_tables ディレクトリーにある **_zos_704-800.sql** ファイルをホーム・ディレクトリーにコピーします。
 - b) このファイルを編集します。FTESRC を既存のスキーマ (FTELOG) に変更し、FTEDEST を新しいスキーマ (FTEV8) に変更します。
 - c) ファイルを確認して、すべてのステートメントが 71 桁目までに入っていることを確認します。ステートメントが 71 桁目までに入っていない場合、71 桁目より前で行を分割します。
 - d) Managed File Transfer 表の **DB2 RUNSTATS jcl** がある場合は、新しいスキーマと表を指定して、新しいジョブを作成します。
8. 一部の表では生成 ID で各行に固有 ID を強制しており、これらの ID を設定する必要があります。以下の表には、生成される ID 列が含まれます。

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

これらの表の生成される ID の値を正しい値にするには、各表について以下のステップを実行します。

- a) 既存データ内の最大 ID 値を判別します。

次の SQL 文を実行してこの値を見つけることができます。

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

このコマンドで戻される値が、指定された表の中で最大の既存 ID です。

- b) 表を変更して、前のステップで戻された値より 1 だけ大きい新規の値から開始するように、ID 生成機能を設定します。

次の SQL 文を実行して、この値を設定することができます。

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

9. データベース・プロパティ・ファイルを編集して、新しいスキーマ名を指定します。

- a) Managed File Transfer 構成ディレクトリーが /u/userid/fteconfig の場合は、USS コマンド **find /u/userid/fteconfig -name databaselogger.properties** を使用して、ロガーのプロパティ・ファイルを見つけることができます。

- b) このファイルを編集し、wmqfte.database.schema を新しいスキーマ値に変更します。

10. ロガーを使用する前に、以下のコマンドを発行して、ディレクトリー・ツリー構造を 8.0.0 形式に変換します。

- a) **fteMigrateConfigurationOptions**

- b) **fteMigrateLogger**

これにより、databaselogger.properties が logger.properties にコピーされます。

11. 8.0.0 Managed File Transfer ライブラリーを使用するように既存のロガー JCL を編集します。

12. ロガーを開始します。

ロガーが作動したら、V7 FTELOG 表は削除できます。

新しいバージョンへの Managed File Transfer for z/OS エージェントのマイグレーション

IBM MQ Managed File Transfer for z/OS エージェントを新しいバージョンにマイグレーションするには、以下の指示に従ってください。このトピックでは、IBM MQ 8.0 を初期バージョンとして使用し、IBM MQ 9.0 を新しいバージョンとして使用します。

重要: ここでの説明では、以下の点を想定しています。

- Managed File Transfer for z/OS (MFT) トポロジーの調整キュー・マネージャーおよびコマンド・キュー・マネージャーは、マイグレーション時には変更されません。
- エージェントのキュー・マネージャーはマイグレーションされていません。エージェントのキュー・マネージャーは、マイグレーションの前後で同じ IBM MQ for z/OS レベルのままです。
- マイグレーション中のエージェントは JCL を使用して管理されます。

ステップ 1 - 初期セットアップ

1. IBM MQ Managed File Transfer for z/OS 8.0 と同じ LPAR に IBM MQ Managed File Transfer for z/OS 9.0 をインストールします。
2. IBM MQ 9.0 で IBM MQ MFT 用の SBFGCMDSDS PDSE を見つけます。
SBFGCMDSDS PDSE について詳しくは、[SBFGCMDSDS ライブラリー](#)を参照してください。
3. このデータ・セット内で、BFGCOPY メンバーを編集して、エージェントを管理するために使用する PDSE の名前を指定します。
4. BFGCOPY メンバーを実行依頼します。これにより、既存の SBFGCMDSDS PDSE が手順 [25 ページの『3』](#) で指定された名前新しい PDSE にコピーされます。

ステップ 2 - 新しい PDSE の変更

1. BFGCOPY メンバーを使用して作成した新しい PDSE 内で、BFGCUSTM を編集し、以下の変数を更新します。

変数	値
++library++	この変数を新しい PDSE の名前で置き換えます。
armELEMENT	これを IBM MQ 8.0 エージェントで使用されるのと同じ値に設定します。
armELEMENTYPE	これを IBM MQ 8.0 エージェントで使用されるのと同じ値に設定します。
BFG_DATA	これを IBM MQ 8.0 エージェントで使用されるのと同じ値に設定します。
BFG_GROUP_NAME	これを IBM MQ 8.0 エージェントで使用されるのと同じ値に設定します。
BFG_JAVA_HOME	これは、エージェントを実行するために使用する Java ランタイム環境の UNIX System Services (USS) パスに設定する必要があります。これを IBM MQ 8.0 エージェントで使用されるのと同じ値に設定します。
BFG_JVM_PROPERTIES	これを IBM MQ 8.0 エージェントで使用されるのと同じ値に設定します。

変数	値
BFG_PROD	この変数は、IBM MQ 9.0 インストール済み環境の Managed File Transfer for z/OS の USS パスに設定する必要があります。
BFG_WTO	これを IBM MQ 8.0 エージェントで使用されるのと同じ値に設定します。
CLEAN_AGENT_PROPS	これはブランクに設定します。
connectionQMGr	これを IBM MQ 8.0 エージェントで使用されるのと同じ値に設定します。
coordinationQMGr	これを IBM MQ 8.0 エージェントで使用されるのと同じ値に設定します。
CREDENTIALS_PATH	これはブランクに設定します。
DB_PROPS_PATH	これを IBM MQ 8.0 エージェントで使用されるのと同じ値に設定します。
DLQ_HLQ	これを IBM MQ 8.0 エージェントで使用されるのと同じ値に設定します。
FTE_CONFIG	これはブランクに設定します。
MQ_HLQ	これを IBM MQ 8.0 エージェントで使用されるのと同じ値に設定します。
MQ_PATH	この変数は、エージェントのキュー・マネージャーに関連付けられた IBM MQ インストールの USS パスに設定する必要があります。
名前	これは、マイグレーションされているエージェントの名前に設定します。
QM	これを IBM MQ 8.0 エージェントで使用されるのと同じ値に設定します。これにより、エージェントが IBM MQ 9.0 を実行しているとき、エージェントは IBM MQ 8.0 を実行しているときと同じエージェント・キュー・マネージャーを使用することになります。
SERVICE_TYPE	エージェント
TMPDIR	これを IBM MQ 8.0 エージェントで使用されるのと同じ値に設定します。

2. 変更された BFGCUSTM メンバーを保存します。
3. BFGCUSTM を実行依頼します。メンバーは、IBM MQ 9.0 を使用しているときにエージェントを管理するために使用できる PDSE 内に新しいメンバーを作成します。



重要: メンバーには PDSE への排他的アクセスが必要なので、BFGCUSTM の実行中はその使用を停止する必要があります。

ステップ 3 - マイグレーションの実行

1. エージェントを管理するために使用する IBM MQ 8.0 PDSE を見つけます。
2. この PDSE 内で、BFGAGSP メンバーを実行依頼してエージェントを停止します。エージェントは、管理対象の転送がすべて完了すると停止します。
3. 次に、エージェントに対して作成したばかりの新しい IBM MQ 9.0 PDSE を見つけます。
4. この PDSE 内で、BFGAGST メンバーを実行依頼します。これにより、IBM MQ Managed File Transfer for z/OS 9.0 を使用してエージェントが開始します。

関連概念

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

このトピックの説明は、ご使用の IBM MQ for z/OS システムで、エンタープライズ内で Managed File Transfer を使用する代わりに方法として適用できます。

異なるオペレーティング・システムを使用する新しいマシンへの MFT のマイグレーション

新しいシステムまたはプラットフォームへの MFT 構成のマイグレーションを正常に行うために必要な主要ステップを取り上げます。このタスクで主に注目しているのは MFT 構成のマイグレーションですが、該当する場合にはキュー・マネージャーのマイグレーションについても取り上げています。

始める前に

マイグレーションするエージェントで進行中または保留中の転送がすべて完了し、以下のバックアップを取ったことを確認します。

- 調整キュー・マネージャー
- エージェント・キュー・マネージャー
- エージェント
- リソース・モニター
- 転送テンプレート
- スケジュールされた転送

重要: あるシステムの IBM MQ インストール環境の名前が、新しいシステムのインストール環境の名前と一致することはほとんどありません。ただし、古いシステムと新しいシステムのどちらにもインストール環境が 1 つしかない場合や、IBM MQ インストール・プロセスの中でインストール環境の名前を指定した場合は別です。

このタスクについて

以下のマイグレーション手順では、QMA がトポロジーの調整キュー・マネージャーでもあり、Agent1 というエージェントのエージェント・キュー・マネージャーでもあるというシナリオを想定します。

Agent1 にはモニター、転送テンプレート、スケジュールされた転送があります。また QMA は、ファイル転送用の受信側と送信側のチャンネルを使用する別のシステムで実行されている QMB というキュー・マネージャーにも接続しています。

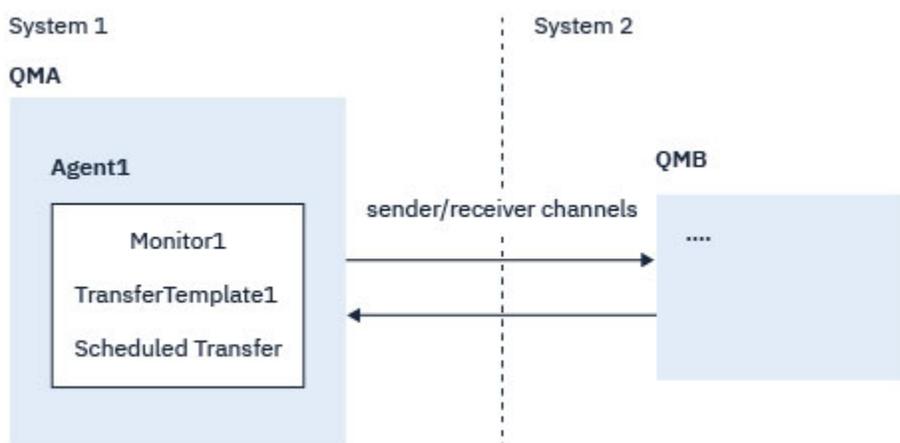


図 1. システム 1 における MFT 構成のマイグレーション



重要: 以下の手順では、MFT 構成のバックアップとリストアの方法のみを取り上げます。MFT を同一のオペレーティング・システムを使用する新しいマシンにマイグレーションする場合には、キュー・マネージャーのデータとログ・ファイルは、古いシステムのデータ・ファイルすべてを新しい

システムの適切なディレクトリーにコピーすることによって、バックアップおよびリストアすることができます。

ただし、新しいマシンのオペレーティング・システムが異なる場合、データ・ファイルはプラットフォーム固有の形式で作成されているため、マイグレーションできません。

手順

1. バックアップ手順

- a) **dmpmqcfcg** コマンドを使用してキュー・マネージャー構成を保存し、後でその定義から再作成します。

以下に例を示します。

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

- b) IBM MQ データ・ディレクトリーに保存されているエージェントの設定ファイルをバックアップします。 /MQ_DATA_PATH/mqft

mqft ディレクトリーには、通常、config、installation、logs という 3 つのサブディレクトリーがあります。これらのサブディレクトリーには、エージェント・インストール・データ、構成、データベース・ロガーのファイルがそれぞれ含まれています。

エージェントがプロトコル・ブリッジ・エージェントである場合には、エージェント構成ディレクトリーの ProtocolBridgeCredentials.xml ファイルもバックアップする必要があります。このファイルは、プロトコル・ブリッジ・エージェントがプロトコル・サーバーでの許可に使用するユーザー名および資格情報を定義します。

- c) MFT **fteListMonitors** コマンドに **-ox** オプションを設定して使用することによって、リソース・モニターの構成を XML ファイルにエクスポートします。

以下に例を示します。

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

- d) MFT **fteListTemplates** コマンドに **-x** オプションと **-o** オプションを設定して使用し、転送テンプレートを XML ファイルにエクスポートします。

例えば、次のコマンドを使用すると、現行ディレクトリーに TransferTemplate1.xml が作成されます。

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

- e) スケジュールされた転送定義を手動でバックアップします。

定義を XML ファイルにエクスポートすることはできませんが、MFT **fteListScheduledTransfers** コマンドを使用して定義を手動でバックアップすることによって、スケジュールされた転送をリストできます。

2. 再作成手順

- a) IBM MQ および MFT を新規システムにインストールした後、キュー・マネージャー QMA を再作成します。

- b) **runmqsc** コマンドを実行して QMA 構成を復元し、ステップ 28 ページの『1.a』で保存したキュー・マネージャー構成を解析します。

以下に例を示します。

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

- c) システム 2 で QMB に接続する送信側と受信側のチャンネルを再作成します。

- d) QMB キュー・マネージャー側で、QMA に接続する送信側チャンネルのホスト名やポート番号など、接続の詳細を更新します。

- e) バックアップしたエージェント構成ファイルすべてを新しいシステムにコピーして Agent1 を再作成し、エージェントを開始します。

- f) Monitor1のXMLファイルをインポートします。そのためには、MFT `fteCreateMonitor` コマンドに `-ix` オプションと `-f` オプションを設定して使用します。

以下に例を示します。

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

- g) メッセージ本体に `TransferTemplate1.xml` の内容が含まれるメッセージを、調整キュー・マネージャーの `SYSTEM.FTE` トピックに発行します。

スタンドアロン・アプリケーションを使用し、トピック・ストリングを指定します。

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

ここでの `<template_id>` は、`TransferTemplate1.xml` ファイル内で検出できる転送テンプレートIDです。

例として、xmlに以下の内容が含まれている場合について考慮します。

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

、トピック・ストリングは次のようになります。

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

- h) MFT `fteCreateTransfers` コマンドを使用して、スケジュールされた転送を手動で再作成します。

MFT シナリオの概要

このセクションでは、一般的な Managed File Transfer トポロジーを、システムをセットアップしてテスト・メッセージを転送するシナリオとともに紹介します。

- [一般的なトポロジー](#)
- [基本サーバーの構成](#)

MFT の一般的なトポロジー

このセクションでは、一般的な Managed File Transfer トポロジーをリストします。各図の両方向矢印は、キュー・マネージャーへの接続を表しています。

キュー・マネージャーの接続オプションについて詳しくは、[32 ページの『接続に関する考慮事項』](#)を参照してください。

1つのキュー・マネージャーが存在する基本トポロジー

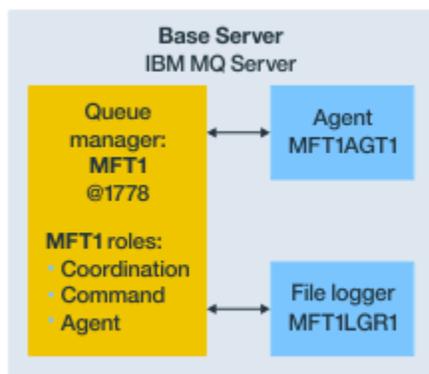


図 2. 1つのキュー・マネージャーが存在する基本トポロジー

基本トポロジーは、調整キュー・マネージャーを含んだ完全な構成を表します。構成名は調整キュー・マネージャーの名前と同じです。調整キュー・マネージャー名が MFT1 であれば、構成名も MFT1 です。

基本トポロジーは、最初に完成させる Managed File Transfer 構成です。基本構成が完成した後、ファイル交換するために、リモート・サーバー上のパートナー・エージェントが基本構成に追加されます。

基本トポロジーでは、基本トポロジー・サーバー外のファイルを交換しません。ただし、基本トポロジーでは、同じサーバー内の異なる場所にファイルを移動できるので、開発の目的で使用することもできます。

1つのパートナー・エージェントが存在する基本トポロジー

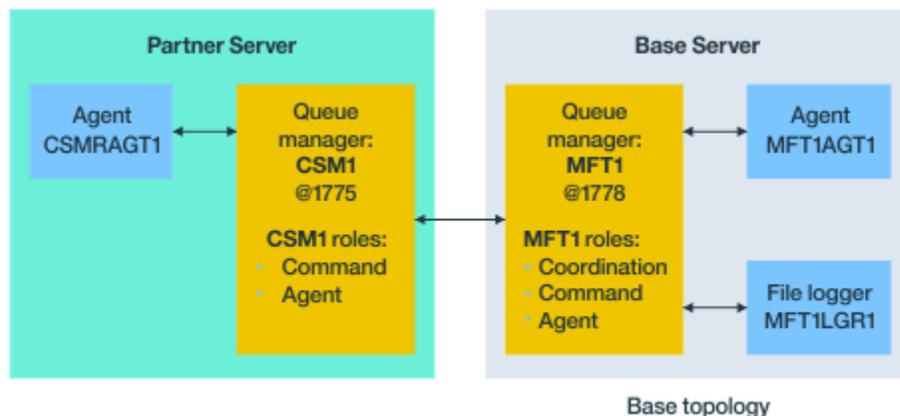


図 3. 1つのパートナー・エージェントが存在する基本トポロジー

このトポロジーでは、2つのエージェントの間でファイルを交換できます。最初に追加されたエージェントと同様の方法で、さらに別のパートナー・エージェントを追加できます。

Managed File Transfer キュー・マネージャーの3つの役割すべてに単一のキュー・マネージャーを使用することも、特定の役割に専用のキュー・マネージャーを使用することもできます。

例えば、調整キュー・マネージャー役割に専用のキュー・マネージャーを1つ使用し、コマンド役割とエージェント役割で2つ目のキュー・マネージャーを共有することもできます。

基本構成とは別個のサーバーにあるリモート・エージェント・キュー・マネージャーと基本構成の調整キュー・マネージャーの間の接続は、IBM MQ クライアント・チャンネル、つまり MQI チャンネルとして構成する必要があります。

調整キュー・マネージャーへの接続は、**fteSetupCoordination** コマンドで確立されます。パートナー・サーバーで調整キュー・マネージャー接続が IBM MQ クライアント・チャンネルとして構成されていない場合は、パートナー・エージェント・サーバーから **fteListAgents** などのコマンドを発行すると失敗します。

別個の調整キュー・マネージャーと1つのパートナー・エージェントが存在する基本トポロジー

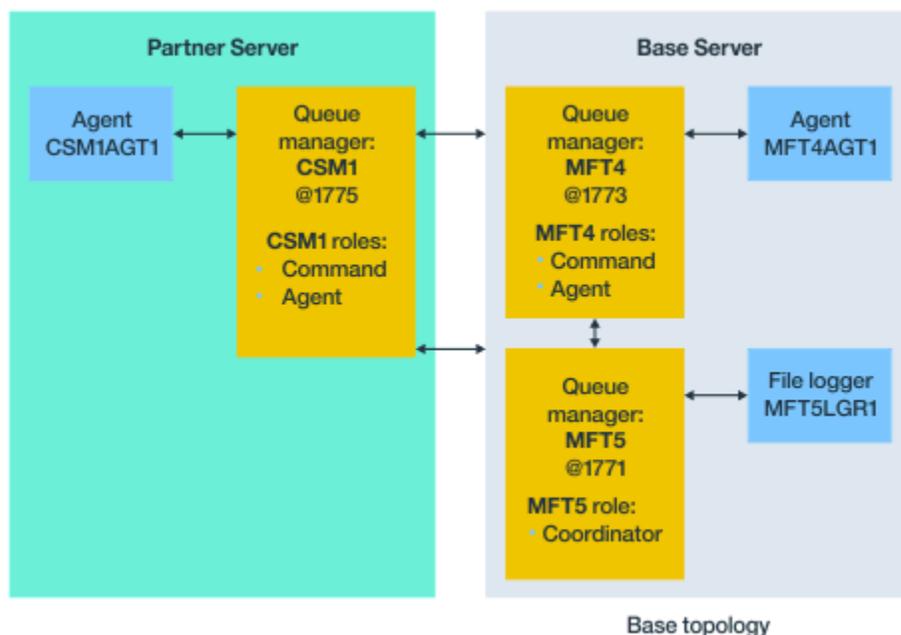


図 4. 別個の調整キュー・マネージャーと1つのパートナー・エージェントが存在する基本トポロジー

図 3 の基本トポロジーでは、基本サーバーのキュー・マネージャー MFT4 はコマンド役割とエージェント役割として共有され、キュー・マネージャー MFT5 は調整キュー・マネージャー役割専用です。

基本トポロジー内のキュー・マネージャー MFT4 および MFT5 を含め、このトポロジーに含まれるすべてのキュー・マネージャー間に接続が存在しなければなりません。

パートナー・サーバーのキュー・マネージャーでは、キュー・マネージャー CSM1 がエージェント・キュー・マネージャーとコマンド・キュー・マネージャーの役割を持っています。

このトポロジーでは、2つのエージェントの間でファイルを交換できます。図に示すように、各パートナー・エージェントをキュー・マネージャーに接続する必要があります。最初のパートナー・エージェントを追加したときと同様の方法で、さらに別のパートナー・エージェントを追加できます。

Managed File Transfer Agent パートナーが存在する基本トポロジー

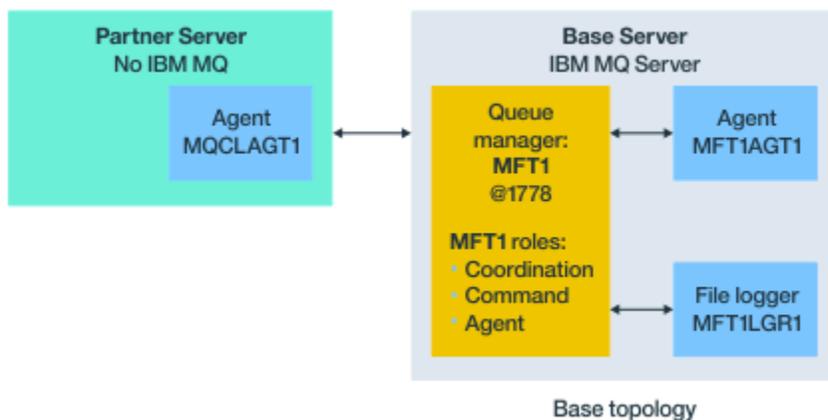


図 5. Managed File Transfer Agent パートナーが存在する基本トポロジー

このトポロジーでは、2つのエージェントの間でファイルを交換できます。

パートナー・エージェント (図では MQCLAGT1) のサーバーには、IBM MQ サーバーがインストールされていません。

パートナー・エージェントは IBM MQ がインストールされたサーバーと同じコマンドを使用して構成されますが、以下に示すいくつかの例外があります。

- このパートナー・エージェントの構成では、基本キュー・マネージャーまたはキュー・マネージャーへの IBM MQ クライアント接続を使用する必要があります。
- パートナー・エージェント・サーバーで構成コマンドによって作成される IBM MQ 調整キュー・マネージャー役割定義を実行する必要はありません。調整キュー・マネージャー定義は既に基本サーバーに存在します。

ただし、以下の操作が必要です。

- パートナー・サーバーでエージェントが作成されるときに生成されるエージェント・オブジェクト定義をコピーします。
- 定義ファイルを基本構成サーバーに転送します。
- 基本サーバーでエージェント・キュー・マネージャーの役割を持つキュー・マネージャーに定義を作成します。

この例では、MFT1 は 3 つの役割すべてを果たすので、MFT1 キュー・マネージャーにエージェント MQCLAGT1 のオブジェクトを作成します。

オブジェクト定義を基本サーバーにコピーする代わりに、エージェント MQCLAGT1 に対する **fteDefine** コマンドを、エージェント・キュー・マネージャーが置かれている基本サーバーで実行することもできます。 **fteDefine** コマンドによって生成される定義を使用して、エージェント・キュー・マネージャーに必要なエージェント定義を作成します。

例えば、この図の場合、パートナー・サーバーのエージェント・ディレクトリーにあるファイル MQCLAGT1_create.mqsc を基本構成サーバーにコピーして、必要なエージェント定義を MFT1 キュー・マネージャーに作成します。

パートナー・エージェント・サーバーで構成を完了すると、Managed File Transfer 構成ディレクトリーと必要なプロパティ・ファイルが作成されます。

V 9.0.1 IBM MQ 9.0.1 以降、パートナー・サーバーで、Fix Central から Managed File Transfer 再配布可能クライアントをインストールできるようになりました。

注: MQMFT 再配布可能クライアントは、既にパッケージ化されていて **genmqpkg** ユーティリティを使用する必要がないという点で、IBM MQ 再配布可能クライアントとは異なります。詳しくは、[再配布可能クライアント](#)を参照してください。

接続に関する考慮事項

上記の図では、エージェントとキュー・マネージャーを結ぶ各線は、キュー・マネージャーへの接続を表しています。

この接続として、以下が考えられます。

- ローカル接続
- バインディング接続、つまりメッセージ・チャンネル接続
- IBM MQ クライアント接続、つまり MQI 接続

構成で選択する接続のタイプは、指定するパラメーターによって決まります。

- キュー・マネージャー名パラメーターを指定して他の接続パラメーターを指定しなければ、バインディング接続を指定することになります。

使用されるキュー・マネージャーが Managed File Transfer 構成上ローカルであれば、基本構成サーバーで使用される接続もローカル接続になります。

- キュー・マネージャー名パラメーターを、対応するホスト、ポート、およびチャンネル名パラメーターとともに指定すると、IBM MQ クライアント接続を指定することになります。

エージェント・キュー・マネージャーと同じホストにエージェントが置かれるときは、バインディング・タイプ (結果的にローカル接続になる) を指定した方がより効率的です。

基本サーバーの構成

別個の構成キュー・マネージャーが存在する基本サーバーをセットアップする方法。

始める前に

以下の例は、次のことを前提としています。

- 32 ページの『[接続に関する考慮事項](#)』のセクションを確認して、構成におけるキュー・マネージャーへの接続のタイプへの影響を理解していること。
- IBM MQ インフラストラクチャーが機能していること。キュー・マネージャーのセットアップについては、[IBM MQ のキュー・マネージャーの構成](#)を参照してください。
- IBM MQ セキュリティー・タスクが完了していること。

ファイルに対するアクセス権限など、すべてのシステム・リソースに適切なセキュリティが構成されていること。

Managed File Transfer のセキュリティ構成については、[Managed File Transfer のセキュリティの概要](#)、および [Managed File Transfer アクションに対するユーザー権限](#)を参照してください。

- IBM MQ の構成後に、サンプル・プログラムを使用してメッセージを送受信するか、サンプル `amqscnxc` を使用して IBM MQ クライアント・タイプの接続をテストして、すべての IBM MQ 接続がテストされていること。

`amqscnxc` サンプルは、サンプル・コードにチャンネル接続を定義することで、キュー・マネージャーに接続されます。これは、MQI タイプの接続、つまり IBM MQ クライアント・タイプの接続が使用されるときに Managed File Transfer の接続に似ています。

- この説明は、基本構成に使用するサーバーに IBM MQ の 1 つのバージョンがインストールされていることを前提としています。基本サーバーに複数の IBM MQ インストール済み環境が存在する場合は、使用するバージョンの IBM MQ に対応する正しいファイル・パスを使用するよう注意してください。
- この説明で使用されるキュー・マネージャーは、接続認証を必要としません。

接続認証を必要とせずに最初の構成を完了する方が簡単な場合がありますが、企業で即時に接続認証を使用する必要がある場合は、[MQMFTCredentials.xml](#) 資格情報ファイルの構成方法について、[Managed File Transfer](#) および [IBM MQ 接続認証](#) を参照してください。

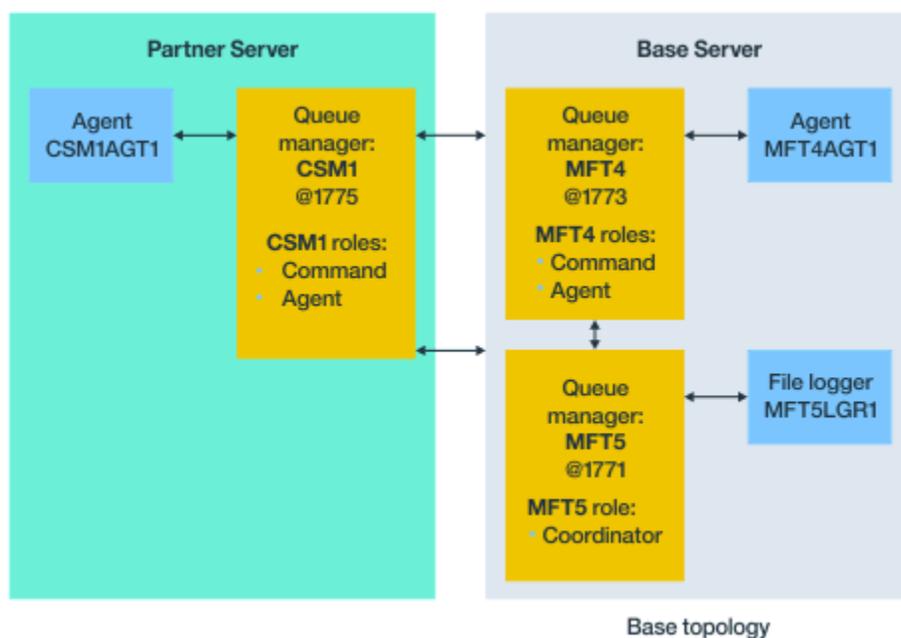


図 6. 別個の調整キュー・マネージャーと1つのパートナー・エージェントが存在する基本トポロジ

このタスクについて

この構成例でのキュー・マネージャー役割は、次のとおりです。

- 基本サーバー
 - キュー・マネージャー MFT5 は調整キュー・マネージャーです。
 - キュー・マネージャー MFT4 は、エージェント MFT4AGT1 のエージェント・キュー・マネージャーとして使用されるとともに、MFT5 構成における基本サーバー上のコマンド・キュー・マネージャーとしても機能します。
- パートナー・サーバー
 - キュー・マネージャー CSM1 は、エージェント CSM1AGT1 のエージェント・キュー・マネージャーと、MFT5 構成におけるパートナー・サーバー上のコマンド・キュー・マネージャーを兼ねています。
 - 基本サーバーのキュー・マネージャー MFT5 は、調整キュー・マネージャーです。

手順

1. 調整キュー・マネージャーの構成
2. コマンド・キュー・マネージャーの構成
3. エージェントのセットアップ
4. ロガーのセットアップ
5. パートナー・サーバーの構成

次のタスク

セットアップ例をテストできるように MQExplorer と MQMFT をセットアップします。

調整キュー・マネージャーの構成

ファイル転送を調整するための調整キュー・マネージャーを構成する方法。

始める前に

このシナリオ用にセットアップしたキュー・マネージャー間の完全な接続性を確保します。

このタスクについて

このタスクでは、調整キュー・マネージャー MFT5 をセットアップします。このセクションの説明は、IBM MQ インストール済み環境が 1 つ存在する環境で作業することを前提としています。

複数のインストール済み環境がある場合は、構成タスクを始める前に、[setmqenv](#) コマンドを使用し、IBM MQ パスとして、必要な IBM MQ のバージョンを設定する必要があります。

手順

1. Managed File Transfer 管理者としてログインします。
2. 次のコマンドを発行して調整キュー・マネージャーを指定し、構成ディレクトリー構造をセットアップします。

```
fteSetupCoordination -coordinationQMgr MFT5
```

調整キュー・マネージャーのディレクトリー

C:\data\mqft\config\MFT5

coordination.properties ファイル

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

このコマンドは、調整キュー・マネージャー C:\data\mqft\config\MFT5\MFT5.mqsc に対して実行する必要がある MQSC コマンド・ファイルも生成します。

3. C:\data\mqft\config\MFT5 ディレクトリーに移動します。
4. 次のコマンドを実行して、調整キュー・マネージャーとなるようにキュー・マネージャーを構成します。
ステップ 35 ページの『2』で発行したコマンドによって生成された MQSC コマンド・ファイルを指定する必要があります。

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

5. 任意のエディターで mft5.txt 結果ファイルを開きます。定義が正常に作成されていることを確認してください。

次のタスク

[コマンド・キュー・マネージャー](#) をセットアップします。

コマンド・キュー・マネージャーの構成

コマンド・キュー・マネージャーを構成する方法。

始める前に

調整キュー・マネージャーが構成済みであることを確認します。詳しくは、[34 ページの『調整キュー・マネージャーの構成』](#)を参照してください。

このタスクについて

このタスクでは、コマンド・キュー・マネージャーを指定します。

手順

以下のコマンドを発行します。

```
fteSetupCommands -connectionQMgr MFT4
```

次のメッセージが表示されます。 BFGCL0245I: ファイル
C:\data\mqft\config\MFT4\command.properties が正常に作成されました。

コマンド・キュー・マネージャーは、追加の IBM MQ 定義を必要としません。 **fteSetupCommands** を実行すると、command.properties ファイルが MFT5 構成ディレクトリーに作成されます。

次のタスク

[エージェントをセットアップ](#)します。

エージェントのセットアップ

ファイル転送エージェント MFT4AGT1 と、実行する必要がある MQSC スクリプトを準備する方法。

始める前に

コマンド・キュー・マネージャーをセットアップしておく必要があります。詳しくは、[35 ページの『コマンド・キュー・マネージャーの構成』](#)を参照してください。

このタスクについて

このタスクでは、Windows ファイル転送エージェント MFT4AGT1 を準備します。

手順

1. 以下のコマンドを発行します。

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

fteCreateAgent コマンドでエージェントを作成すると、エージェント MFT4AGT1 の agents ディレクトリーとサブディレクトリーが MFT5 ディレクトリーに追加されます。

data\MFT5\agents\MFT4AGT1 ディレクトリーには、以下のものがあります。

- agent.properties ファイル
 - MFT4AGT1_create.mqsc ファイル、エージェントが必要とする IBM MQ 定義が含まれています。
2. data\MFT5\agents\MFT4AGT1 ディレクトリーに移動し、以下のコマンドを発行して、必要なエージェント・キュー・マネージャー定義を作成します。

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

3. 任意のエディターで mft4.txt 結果ファイルを開き、定義が正常に作成されたことを確認します。
4. コマンド **fteStartAgent MFT4AGT1** を入力してエージェントを開始します。
5. コマンド **fteListAgents** を入力してエージェントを表示します。

次のような出力が表示されます。

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

注: Managed File Transfer 環境での接続認証を有効にしていない場合、BFGPR0127W メッセージは無視できます。

ftelistAgents コマンドを発行して、BFGCL0014W: 現在の選択基準に一致するエージェントが存在しません。というメッセージを受け取った場合、詳しくは、302 ページの『MFT エージェントが **ftelistAgents** コマンドによってリストされない場合に行う事柄』を参照してください。

次のタスク

ロガーをセットアップします。

ロガーのセットアップ

構成における転送アクティビティーに関する履歴情報と監査情報を保持するためには、ファイル・ロガーまたはデータベース・ロガーが必要です。この例では、ファイル・ロガーを作成します。

始める前に

次のものをセットアップしておく必要があります。

- 構成キュー・マネージャー
- コマンド・キュー・マネージャー
- エージェント

手順

1. 以下のコマンドを発行します。

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

fteCreateLogger コマンドを実行すると、`data\mqft\config\MFT5\loggers` ディレクトリーが `MFT5LGR1` サブディレクトリーとともに作成されます。

`MFT5LGR1` サブディレクトリーには `logger.properties` ファイルが格納されます。このディレクトリーには、ロガーが必要とする IBM MQ 定義が含まれる `MFT5LGR1_create.mqsc` というファイルもあります。

2. ディレクトリー `data\mqft\config\MFT5\loggers\MFT5LGR1` に移動します。
3. 関連する MQSC コマンド・ファイルを実行します。

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

これにより、ロガーが必要とする定義が作成されます。

a) オブジェクト定義の結果を調べて、必要なオブジェクトが正常に作成されたことを確認します。

4. コマンド **fteStartLogger** `MFT5LGR1` を発行してロガーを開始します。
5. `data\mqft\logs\MFT5\loggers\MFT5LGR1\logs` にあるファイル `output0.log` の内容を確認します。

ロガーに関する情報に続く最後のステートメントに、「BFGDB0023I: ロガーは開始アクティビティーを完了し、現在、実行中です」というメッセージがあるはずですが。

ロガーが初めて開始されたときは、`output0.log` にログ情報が書き込まれないことがあります。`output0.log` ファイルが空の場合は、**fteStopLogger** `MFT5LGR1` と入力してから **Enter** キーを押して、ロガーを再始動します。

fteStartLogger `MFT5LGR1` と入力してから **Enter** キーを押して、ロガーを再始動します。ファイル `output0.log` にデータが示されるようになります。

エージェント版の `output0.log` ファイルでも、エージェントが初めて開始されたときは同じ動作になります。

fteStopAgent コマンドと **fteStartAgent** コマンドを使用して、エージェントを停止してから開始します。その後、エージェントの `output0.log` ファイルにログ・データが書き込まれるようになります。

タスクの結果

この構成における調整キュー・マネージャーを設定した基本サーバーを構成しました。

次のタスク

次は、リモート・エージェントが含まれるパートナー・サーバーに関して同様の作業をします。

パートナー・サーバーの構成

基本サーバーに別個の調整キュー・マネージャーがある場合にパートナー・サーバーを構成する方法。

始める前に

構成キュー・マネージャーを含む基本サーバーをセットアップするためのすべてのタスクが完全に完了していることを確認します。

このタスクについて

IBM MQ とセキュリティー構成、および IBM MQ パスに関する前提と同じ前提が、パートナー・サーバーにも適用されます。

初めに **fteSetupCoordination** コマンドを使用して、MFT5 構成ディレクトリーをセットアップし、調整キュー・マネージャーを指定します。

手順

1. 次のコマンドを発行して、パートナー・サーバー構成ディレクトリーを作成します。

```
fteSetupCoordination -coordinationQMgr MFT5
-coordinationQMgrHost 177.16.20.15 -coordinationQMgrPort 1771
-coordinationQMgrChannel MQMFT.MFT5.SVRCONN
```

注:

- a. 調整キュー・マネージャーがパートナー・サーバーとは異なるサーバーにある場合、基本サーバーの調整キュー・マネージャーへの接続は、クライアント接続として定義する必要があります。
パートナー・サーバーで調整キュー・マネージャー接続を IBM MQ クライアント接続として定義しなければ、調整キュー・マネージャーに接続する Managed File Transfer コマンドはどれも、失敗することになります。
調整キュー・マネージャーに接続するコマンドの例として、**fteListAgents** があります。
 - b. IBM MQ 定義を作成する必要はありません。調整キュー・マネージャーが必要とする定義は、基本サーバーを構成したときに完了しているからです。
2. 次のコマンドを発行して、コマンド・キュー・マネージャーを指定します。

```
fteSetupCommands -connectionQMgr CSM1
```

コマンド・キュー・マネージャーは、追加の IBM MQ 定義を必要としません。

3. 次のコマンドを発行して、パートナー・エージェント・キュー・マネージャーを指定し、パートナー・エージェント・キュー・マネージャーを作成します。

```
fteCreateAgent -agentName CSM1AGT1 -agentQMgr CSM1
```

- CSM1AGT1 ディレクトリーに移動します。
- 次のコマンドを発行して、エージェントが必要とする IBM MQ 定義を作成します。

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

- 任意のエディターで `csm1.txt` ファイルを開いて、エージェントが必要とする定義がすべて正常に作成されたことを確認します。
- 次のコマンドを発行して、エージェントを開始します。

```
fteStartAgent CSM1AGT1
```

- fteListAgents** と入力してエージェントを表示します。
次のような出力が表示されます。

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

注: Managed File Transfer 環境での接続認証を有効にしていない場合、BFGPR0127W メッセージは無視できます。

fteListAgents コマンドを発行して、BFGCL0014W: 現在の選択基準に一致するエージェントが存在しません。というメッセージを受け取った場合、詳しくは、[302 ページの『MFT エージェントが fteListAgents コマンドによってリストされない場合に行う事柄』](#)を参照してください。

いずれかのエージェントの状況が UNREACHABLE である場合、詳しくは、[304 ページの『エージェントが UNKNOWN 状態にあると表示される場合の処理』](#)を参照してください。

IBM MQ Explorer と MFT のセットアップ

このタスクは、IBM MQ Explorer を Managed File Transfer 構成に接続するためのものです。

手順

- IBM MQ Explorer を開始します。
- 左側の Navigator パネルで、スクロールダウンしてフォルダー Managed File Transfer を展開します。
調整キュー・マネージャー MFT5 の項目があります。
- MFT5 を右クリックして、「**接続**」を選択します。
 - 表示されるドロップダウン・メニューで **Agents** を選択し、MFT4AGT1 と CSMAGT1 の両方のエージェントが Ready 状態であることを確認します。

次のタスク

[IBM MQ Explorer](#) を使用してセットアップ例をテストします。

IBM MQ Explorer を使用したファイル転送のテスト

このタスクでは、前のトピックの説明に従って IBM MQ Explorer をセットアップした後に、IBM MQ Explorer と Managed File Transfer を使用してファイル転送をテストする方法の例を示します。

始める前に

稼働中のシステムがあり、エージェントが READY であり、IBM MQ Explorer が動作していることを確認します。詳しくは、[39 ページの『IBM MQ Explorer と MFT のセットアップ』](#)を参照してください。

このタスクについて

転送のテストに使用するファイルと、そのコピー先となるディレクトリーを決定します。この例では、ファイル `test-file.txt` out of directory `C:\temp\mft` が使用されることを想定しています。

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

手順

- Windows で IBM MQ Explorer を開始します。
- 左側のナビゲーター・パネルで、Managed File Transfer フォルダを展開します。
調整キュー・マネージャ MFT5 の項目があります。
- MFT5 を右クリックして、「接続」を選択します。
- 接続されたら、MFT5 を右クリックして、「新規の転送」を選択します。
 - プルダウン・メニューを使用して、ソース・エージェントとして MFT4AGT1 を、宛先エージェントとして CSMAGT1 を選択します。
 - 「次へ」をクリックします。
 - 次のウィンドウで、「追加」をクリックします。
幅広のダイアログが表示されます。左側は Source 用で、右側は Destination 用です。
- 「Source」パネルで、以下を行います。
 - ファイルはテキストなので、「テキスト転送」を選択します。
 - 「参照」を選択してファイルを見つけます。
この場合、当該ファイルは `C:\temp\mft\test-file.txt` になります。
 **重要:** Destination パネルに入力する必要があるため、「OK」をクリックしないでください。
- 「Destination」パネルで、以下を行います。
 - 宛先でのファイルに付ける名前を入力します (例えば `test-file.txt`)。
相対パスの使用がサポートされています。絶対パスの先頭部分は、宛先エージェントを開始するユーザー ID のホーム・ディレクトリーです。
 - このオプションが必要な場合は、Overwrite files if present を選択します。
 - 「OK」をクリックします。
選択したファイルが「新規の転送」パネルに表示されます。
- MFT5 構成メニューが閉じていて +MFT5 が表示されている場合は、「+」符号をクリックしてメニューを展開します。
- MFT 構成を選択したままにしておきます。
次に、以下の手順を実行して、転送の状況を確認します。
- 調整キュー・マネージャ MFT5 の項目の中の「転送ログ」をクリックします。
- 「転送ログ」上部パネルのすぐ下にある Managed File Transfer - Current Transfer progress panel の状況を確認し、転送が完了するまで待ちます。
転送が成功を示していて、背景が緑色であれば、構成のテストを正常に完了したことになります。
転送が失敗して背景が赤色であれば、エラーが発生したことになります。
ほとんどの場合、上部「転送ログ」パネルの下のスクロール・バーを使用して、失敗の理由の要約を見ることができます。
 - 転送が失敗した理由を判別できない場合は、上部「転送ログ」パネルで転送の項目をダブルクリックします。
 - 表示されるポップアップ・パネルの左ペインで XML を選択します。

- c) 情報をスクロールして、エラーの原因を判別します。
- d) 必要な修正を行った後に、転送を再度テストします。

MFT の保護

インストール直後の変更のない状態では、Managed File Transfer のセキュリティー・レベルは、保護された環境におけるテストまたは評価のためには適している可能性があります。ただし、実稼働環境では、ファイル転送操作を開始できるユーザー、転送されているファイルの読み取りおよび書き込みができるユーザー、およびファイルの保全性を保護する方法の適切な管理について考慮する必要があります。

関連概念

[45 ページの『MFT のサンドボックス』](#)

ファイル・システムの中で、エージェントが転送処理時にアクセスできる領域を制限できます。エージェントがアクセスできる制限領域のことをサンドボックスといいます。制限の適用対象は、エージェントにすることも、転送を要求するユーザーにすることも可能です。

関連タスク

[51 ページの『MFT の SSL または TLS 暗号化の構成』](#)

IBM MQ Managed File Transfer で SSL または TLS を使用して、エージェントとエージェント・キュー・マネージャーの間の通信、接続先のコマンドとキュー・マネージャー、およびトポロジー内のさまざまなキュー・マネージャーとキュー・マネージャーの間の接続を保護することができます。

関連資料

[353 ページの『MFT 固有リソースのグループ権限の制限』](#)

関連する可能性のあるさまざまなオブジェクトのすべてについて、個々のユーザーに権限を付与する代わりに、Managed File Transfer のアクセス制御を管理する目的で、2つのセキュリティー・グループ FTEUSER と FTEAGENT を構成します。これらのグループの作成と埋め込みは、IBM MQ 管理者が行う責任があります。管理者は、ここで説明される構成の提案を拡大または変更する選択ができます。

[352 ページの『MFT 固有リソースの権限の管理』](#)

どのファイル転送要求でも、Managed File Transfer エージェント・プロセスでは、ローカル・ファイル・システムに対する一定レベルのアクセス権限が必要になります。さらに、エージェント・プロセスと関連付けられているユーザー ID と、ファイル転送操作を実行しているユーザーと関連付けられているユーザー ID の両方は、特定の IBM MQ オブジェクトを使用する権限を持つ必要があります。

[364 ページの『ファイル・システムにアクセスするための MFT 権限』](#)

どのファイル転送要求でも、Managed File Transfer エージェント・プロセスでは、ローカル・ファイル・システムに対する一定レベルのアクセス権限が必要になります。

[42 ページの『MFT と IBM MQ の接続認証』](#)

接続認証では、指定されたユーザー ID とパスワードを使用してアプリケーションを認証するようキュー・マネージャーを構成できます。関連付けられたキュー・マネージャーのセキュリティーが使用可能に設定されており、資格情報の詳細(ユーザー ID とパスワード)が必要な場合、キュー・マネージャーと正常に接続するには、その前に接続認証フィーチャーを使用可能にしておく必要があります。接続認証は互換モードでも、MQCSP 認証モードでも実行できます。

[365 ページの『commandPath MFT プロパティー』](#)

commandPath プロパティーを使用して、Managed File Transfer がコマンドを実行できる場所を指定します。指定された commandPath にある任意のコマンドは、エージェントにコマンドを送信できるリモート・クライアント・システムから事実上呼び出すことができるため、このプロパティーを設定するときは十分な注意を払ってください。

[364 ページの『MFT エージェント・ログ・メッセージおよび状況メッセージをパブリッシュする権限』](#)

Managed File Transfer エージェントは、調整キュー・マネージャーにパブリッシュされるさまざまなログ・メッセージ、進行メッセージ、および状況メッセージを発行します。これらのメッセージのパブリケーションは、IBM MQ セキュリティー・モデルの影響を受け、場合によっては、パブリケーションを可能にするために、さらに構成が必要になることがあります。

MFT と IBM MQ の接続認証

接続認証では、指定されたユーザー ID とパスワードを使用してアプリケーションを認証するようキュー・マネージャーを構成できます。関連付けられたキュー・マネージャーのセキュリティーが使用可能に設定されており、資格情報の詳細 (ユーザー ID とパスワード) が必要な場合、キュー・マネージャーと正常に接続するには、その前に接続認証フィーチャーを使用可能にしておく必要があります。接続認証は互換モードでも、MQCSP 認証モードでも実行できます。

資格情報の詳細を提供する方法

多くの Managed File Transfer コマンドで、資格情報の詳細を提供するための以下の方法がサポートされています。

コマンド行引数で詳細を提供する。

資格情報の詳細は、**-mquserid** パラメーターおよび **-mqpassword** をパラメーター使用して指定できます。**-mqpassword** が指定されないと、ユーザーはパスワードの入力を求められます。ここで入力された内容は表示されません。

詳細を資格情報ファイル **MQMFTCredentials.xml** から提供する。

資格情報の詳細は、平文または難読化テキストのいずれかとして **MQMFTCredentials.xml** ファイル内で事前に定義できます。

IBM MQ for Multiplatforms で **MQMFTCredentials.xml** ファイルをセットアップする方法については、[43 ページの『マルチプラットフォームでの MQMFTCredentials.xml の構成』](#)を参照してください。

IBM MQ for z/OS での **MQMFTCredentials.xml** ファイルのセットアップについては、[z/OS での MQMFTCredentials.xml の構成](#)を参照してください。

優先順位

資格情報の詳細は、次の優先順位で決まります。

1. コマンド行引数。
2. コマンドを実行している関連キュー・マネージャーとユーザーによって索引付けされた **MQMFTCredentials.xml**。
3. 関連キュー・マネージャーによって索引付けされた **MQMFTCredentials.xml**。
4. 以前のリリースの IBM MQ または IBM WebSphere MQ との互換性を許可する資格情報の詳細が提供されないデフォルトの後方互換性モード

注:

- **fteStartAgent** コマンドおよび **fteStartLogger** コマンドは、コマンド行引数 **-mquserid** と **-mqpassword** をサポートしておらず、資格情報の詳細は **MQMFTCredentials.xml** ファイルで指定する方法のみが可能です。

▶ z/OS

z/OS では、ユーザーのパスワードに小文字が含まれている場合でも、パスワードを大文字にする必要があります。例えば、ユーザーのパスワードが「password」であれば、「PASSWORD」と入力する必要があります。

関連資料

[MFT コマンドとその接続先のキュー・マネージャー](#)

Managed File Transfer トポロジは、さまざまなコンポーネントによって構成されます。

[MFT の資格情報ファイルのフォーマット](#)

MQMFTCredentials.xml ファイルには、ユーザー ID とパスワードに関する重要な情報が含まれています。**MQMFTCredentials.xml** ファイル内のエレメントは **MQMFTCredentials.xsd** スキーマに準拠する必要があります。資格情報ファイルのセキュリティーはユーザーの責任です。

マルチプラットフォームでの MQMFTCredentials.xml の構成

Managed File Transfer (MFT) がセキュリティーを有効にして構成されている場合、接続認証では、キュー・マネージャーに接続するすべての MFT コマンドでユーザー ID とパスワードの資格情報を提供する必要があります。同様に、MFT ロガーは、データベースへの接続時にユーザー ID とパスワードを指定する必要があります。この資格情報は、MFT 資格情報ファイルに保管できます。

このタスクについて

MQMFTCredentials.xml ファイル内のエレメントは MQMFTCredentials.xsd スキーマに準拠する必要があります。MQMFTCredentials.xml のフォーマットについては、[MFT 資格情報ファイル・フォーマット](#)を参照してください。

資格情報ファイルの例は、MQ_INSTALLATION_PATH/mqft/samples/credentials ディレクトリーにあります。

MFT 資格情報ファイルは、調整キュー・マネージャー用に 1 つ、コマンド・キュー・マネージャー用に 1 つ、各エージェントに 1 つ、各ロガーに 1 つ使用できます。あるいは、トポロジー内のすべてのもので使用される 1 つのファイルを使用することもできます。

MFT 資格情報ファイルのデフォルトの場所は以下のとおりです。

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

Windows **Windows**
%USERPROFILE% または %HOMEDRIVE%%HOMEPATH%

資格情報ファイルが別の場所に保管されている場合は、以下のプロパティーを使用して、コマンドが検索する場所を指定できます。

コマンドのタイプ	プロパティー・ファイル	プロパティー名
調整キュー・マネージャーに接続するコマンド	coordination.properties	coordinationQMGrAuthenticationCredentialsFile
コマンド・キュー・マネージャーに接続するコマンド	connection.properties	connectionQMGrAuthenticationCredentialsFile
エージェント・プロセスに接続するコマンド	agent.properties	agentQMGrAuthenticationCredentialsFile
ロガー・プロセスに接続するコマンド	logger.properties	loggerQMGrAuthenticationCredentialsFile

コマンドのタイプ	プロパティー・ファイル	プロパティー名
MFT エージェント	agent.properties	agentQMGrAuthenticationCredentialsFile
MFT loggers	logger.properties	loggerQMGrAuthenticationCredentialsFile

どのコマンドおよびプロセスがどのキュー・マネージャーに接続するかについては、[どの MFT コマンドおよびプロセスがどのキュー・マネージャーに接続するか](#)を参照してください。

資格情報ファイルにはユーザー ID とパスワードの情報が含まれているため、このファイルへの無許可アクセスを防止するには特別な権限が必要です。

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

Windows Windows

継承が有効になっていないことを確認してから、資格情報ファイルを使用するエージェントまたはローガーを実行しているユーザー ID を除き、すべてのユーザー ID を削除してください。

IBM MQ Explorer Managed File Transfer プラグインで MFT 調整キュー・マネージャーに接続するために使用される資格情報の詳細は、構成のタイプによって異なります。

グローバル (ローカル・ディスク上の構成)

グローバル構成は、調整プロパティーおよびコマンド・プロパティーで指定された資格情報ファイルを使用します。

ローカル (IBM MQ Explorer 内で定義する)

ローカル構成は、IBM MQ Explorer 内で関連付けられたキュー・マネージャーの接続詳細のプロパティーを使用します。

関連タスク

44 ページの『MQCSP 認証の有効化』

IBM MQ 9.0.3 for Continuous Delivery および IBM MQ 9.0.0 Fix Pack 2 for Long Term Support 以降、調整キュー・マネージャーまたはコマンド・キュー・マネージャーと接続する IBM MQ Explorer MFT プラグインの接続認証のために MQCSP 認証モードを有効にすることができます。また、調整キュー・マネージャーまたはコマンド・キュー・マネージャーに接続する Managed File Transfer エージェントの接続認証を行うために、MQCSP 認証モードを有効にすることもできます。

関連資料

MFT の資格情報ファイルのフォーマット

MQMFTCredentials.xml ファイルには、ユーザー ID とパスワードに関する重要な情報が含まれています。MQMFTCredentials.xml ファイル内のエレメントは MQMFTCredentials.xsd スキーマに準拠する必要があります。資格情報ファイルのセキュリティはユーザーの責任です。

関連情報

fteObfuscate: 機密データの暗号化

V 9.0.0.2

V 9.0.3

MQCSP 認証の有効化

IBM MQ 9.0.3 for Continuous Delivery および IBM MQ 9.0.0 Fix Pack 2 for Long Term Support 以降、調整キュー・マネージャーまたはコマンド・キュー・マネージャーと接続する IBM MQ Explorer MFT プラグインの接続認証のために MQCSP 認証モードを有効にすることができます。また、調整キュー・マネージャーまたはコマンド・キュー・マネージャーに接続する Managed File Transfer エージェントの接続認証を行うために、MQCSP 認証モードを有効にすることもできます。

このタスクについて

IBM MQ Explorer Managed File Transfer プラグインを使用する場合、または CLIENT トランスポートを使用してキュー・マネージャーに接続し、パスワードを指定する Managed File Transfer エージェントがある場合は、指定されたパスワードが 12 文字よりも長いと、エージェントはキュー・マネージャーを認証しません。これは、コードが MQCSP 認証を使用せず、パスワードの長さを 12 文字に制限する互換モードを使用して認証を行うためです。

IBM MQ 9.0.3 および IBM MQ 9.0.0 Fix Pack 2 以降では、デフォルトの互換モードを無効にして、MQCSP 認証モードを有効にできます。

手順

- IBM MQ Explorer の調整キュー・マネージャーまたはコマンド・キュー・マネージャーで互換モードを無効にして MQCSP 認証を有効にするには、以下の手順を実行します。

- a) 接続先キュー・マネージャーを選択します。
 - b) 右クリックして、ポップアップ・メニューから「**接続詳細**」>「**プロパティ**」を選択します。
 - c) 「**ユーザー ID**」タブをクリックします。
 - d) 「**ユーザー ID を有効にする**」が選択されていることを確認し、「**ユーザー ID の互換モード**」チェック・ボックスをクリアします。
- Managed File Transfer エージェントで互換モードを無効にして MQCSP 認証を有効にするには、パラメーター **useMQCSPAuthentication** を関連するユーザーの MQMFTCredentials.xml ファイルに追加して、その値を true に設定します。

このパラメーターは true に設定する必要があります。このパラメーターが指定されていない場合は、デフォルトで false に設定され、キュー・マネージャーに対するユーザーの認証に互換モードが使用されます。

次の例は、MQMFTCredentials.xml ファイルで **useMQCSPAuthentication** パラメーターを設定する方法を示しています。

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

関連資料

[42 ページの『MFT と IBM MQ の接続認証』](#)

接続認証では、指定されたユーザー ID とパスワードを使用してアプリケーションを認証するようキュー・マネージャーを構成できます。関連付けられたキュー・マネージャーのセキュリティが使用可能に設定されており、資格情報の詳細 (ユーザー ID とパスワード) が必要な場合、キュー・マネージャーと正常に接続するには、その前に接続認証フィーチャーを使用可能にしておく必要があります。接続認証は互換モードでも、MQCSP 認証モードでも実行できます。

[810 ページの『MFT 資格情報ファイル・フォーマット』](#)

MQMFTCredentials.xml ファイルには、ユーザー ID とパスワードに関する重要な情報が含まれています。MQMFTCredentials.xml ファイル内のエレメントは MQMFTCredentials.xsd スキーマに準拠する必要があります。資格情報ファイルのセキュリティはユーザーの責任です。

関連情報

[MQCSP パスワード保護](#)

MFT のサンドボックス

ファイル・システムの中で、エージェントが転送処理時にアクセスできる領域を制限できます。エージェントがアクセスできる制限領域のことをサンドボックスといいます。制限の適用対象は、エージェントにすることも、転送を要求するユーザーにすることも可能です。

エージェントがプロトコル・ブリッジ・エージェントまたは Connect:Direct ブリッジ・エージェントである場合は、サンドボックスはサポートされません。IBM MQ キューとの間で転送する必要のあるエージェントに、エージェント sandboxing を使用することはできません。

関連資料

[45 ページの『MFT エージェント・サンドボックスの処理』](#)

追加のセキュリティ・レベルを Managed File Transfer に加えるため、エージェントがアクセスできるファイル・システムの領域を制限することができます。

[47 ページの『MFT ユーザー・サンドボックスの処理』](#)

ファイルの転送先および転送元とすることが可能なファイル・システム内の領域を、転送を要求する MQMD ユーザー名に基づいて制限することができます。

MFT エージェント・サンドボックスの処理

追加のセキュリティ・レベルを Managed File Transfer に加えるため、エージェントがアクセスできるファイル・システムの領域を制限することができます。

エージェント・サンドボックス機能は、IBM MQ キューとの間で転送を行うエージェントに対して使用することはできません。サンドボックス機能によって IBM MQ キューへのアクセスの制限を実装するには、代わりにユーザー・サンドボックス機能を使用します。これはすべてのサンドボックス機能要件で推奨されるソリューションです。ユーザー・サンドボックス機能について詳しくは、[47 ページの『MFT ユーザー・サンドボックスの処理』](#)を参照してください。

エージェント・サンドボックスを有効にするには、制限を設定するエージェントの `agent.properties` ファイルに、次のプロパティを追加します。

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

ここで、

- `restricted_directory_name` は、許可あるいは拒否されるディレクトリー・パスです。
- `!` はオプションであり、`restricted_directory_name` の以下の値が拒否される (除外される) ことを指定します。`!` が指定されていない場合、`restricted_directory_name` は許可された (組み込まれた) パスです。
- `separator` は、プラットフォーム固有の分離文字です。

例えば、AGENT1 のアクセス権限を /tmp ディレクトリーのみで制限し、そのサブディレクトリー `private` のアクセスは許可しない場合、AGENT1: `sandboxRoot=/tmp:!/tmp/private` に所属する `agent.properties` ファイルに、次のようにプロパティを設定します。

`sandboxRoot` プロパティは、『[拡張エージェント・プロパティ](#)』で説明されています。

エージェント・サンドボックス機能、およびユーザー・サンドボックス機能は、いずれもプロトコル・ブリッジ・エージェント、または Connect:Direct ブリッジ・エージェントではサポートされていません。

UNIX、Linux、および Windows プラットフォームでのサンドボックスの機能

ULW UNIX、Linux、および Windows プラットフォームでは、サンドボックス機能により Managed File Transfer Agent の読み取り元および書き込み先のディレクトリーを制限します。サンドボックスがアクティブな場合、Managed File Transfer Agent は、許可すると指定されたディレクトリーと、その指定されたディレクトリーに含まれるサブディレクトリー (ただし、そのサブディレクトリーが `sandboxRoot` で拒否すると指定されていない場合) への読み取りと書き込みができます。Managed File Transfer のサンドボックスは、オペレーティング・システムのセキュリティより優先順位が上ではありません。Managed File Transfer Agent を開始したユーザーには、ディレクトリーからの読み取りとディレクトリーへの書き込みができるように、そのディレクトリーに対するオペレーティング・システムの適切なレベルのアクセス権限が必要です。リンクしているディレクトリーが指定された `sandboxRoot` ディレクトリー (およびサブディレクトリー) 以外にある場合、ディレクトリーへのシンボリック・リンクをたどることはできません。

z/OS でのサンドボックスの機能

z/OS z/OS の場合、サンドボックスは Managed File Transfer Agent の読み取り元および書き込み先であるデータ・セット名修飾子を制限します。Managed File Transfer Agent を開始したユーザーには、関係するどのデータ・セットに対しても正しいオペレーティング・システムの権限がなければなりません。`sandboxRoot` データ・セット名修飾子の値を二重引用符で囲むと、その値は通常の z/OS 規則に従い、完全修飾として処理されます。二重引用符を省略すると、`sandboxRoot` の前に現在のユーザー ID が接頭部として付けられます。例えば、`sandboxRoot` プロパティを `sandboxRoot=//test` に設定すると、エージェントは、次のデータ・セットに (標準 z/OS 表記で) アクセスすることができます。//
`username.test.**` ランタイムでは、完全に解決したデータ・セット名の初期レベルが `sandboxRoot` と一致しない場合、転送要求は拒否されます。

IBM i システムでのサンドボックスの機能

IBM i IBM i システムの統合ファイル・システムのファイルの場合は、サンドボックスは、Managed File Transfer Agent の読み取り元および書き込み先のディレクトリーを制限します。サンドボックスがアクティブな場合、Managed File Transfer Agent は、許可すると指定されたディレクトリーと、その指定さ

れたディレクトリーに含まれるサブディレクトリー (ただし、そのサブディレクトリーが `sandboxRoot` で拒否すると指定されていない場合) への読み取りと書き込みができます。Managed File Transfer のサンドボックスは、オペレーティング・システムのセキュリティーより優先順位が上ではありません。Managed File Transfer Agent を開始したユーザーには、ディレクトリーからの読み取りとディレクトリーへの書き込みができるように、そのディレクトリーに対するオペレーティング・システムの適切なレベルのアクセス権限が必要です。リンクしているディレクトリーが指定された `sandboxRoot` ディレクトリー (およびサブディレクトリー) 以外にある場合、ディレクトリーへのシンボリック・リンクをたどることはできません。

関連資料

50 ページの『ワイルドカード転送の追加検査』

Continuous Delivery の IBM MQ 9.0.1 から、および Long Term Support の IBM MQ 9.0.0 Fix Pack 1 からは、エージェントがファイルを転送できる場所を制限するためにユーザー・サンドボックスまたはエージェント・サンドボックスを使用してエージェントが構成されている場合、そのエージェントのワイルドカード転送に対して追加の検査を行うように指定できます。

45 ページの『MFT エージェント・サンドボックスの処理』

追加のセキュリティー・レベルを Managed File Transfer に加えるため、エージェントがアクセスできるファイル・システムの領域を制限することができます。

550 ページの『MFT agent.properties ファイル』

各 Managed File Transfer Agent には、独自のプロパティー・ファイル `agent.properties` があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。agent.properties ファイルには、エージェントの動作を変更するプロパティーを含めることもできます。

MFT ユーザー・サンドボックスの処理

ファイルの転送先および転送元とすることが可能なファイル・システム内の領域を、転送を要求する MQMD ユーザー名に基づいて制限することができます。

ユーザー・サンドボックスは、エージェントがプロトコル・ブリッジ・エージェントまたは Connect:Direct ブリッジ・エージェントである場合はサポートされません。

ユーザー・サンドボックスを有効にするには、制限を設定するエージェントの `agent.properties` ファイルに、次のプロパティーを追加します。

```
userSandboxes=true
```

このプロパティーが存在し `true` に設定されている場合、エージェントは `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name/UserSandboxes.xml` ファイル内の情報を使用して、転送を要求したユーザーがファイル・システムのどの部分にアクセス可能であるかを判別します。

`UserSandboxes.xml` XML は、`<sandbox>` エレメントを 0 個以上含んでいる 1 つの `<agent>` エレメントで構成されています。これらのエレメントは、どの規則がどのユーザーに適用されるかを記述します。`<sandbox>` エレメントの `user` 属性は、要求の MQMD ユーザーと突き合わせるために使用されるパターンです。

ファイル `UserSandboxes.xml` はエージェントによって定期的に再ロードされ、ファイルに有効な変更が加えられていれば、それはエージェントの動作に反映されます。デフォルトの再ロード間隔は 30 秒です。この間隔は、`agent.properties` ファイルの `xmlConfigReloadInterval` エージェント・プロパティーを指定することにより変更できます。

`userPattern="regex"` 属性または値を指定する場合、`user` 属性は Java 正規表現として解釈されます。詳しくは、659 ページの『MFT が使用する正規表現』を参照してください。

`userPattern="regex"` 属性も値も指定しない場合、`user` 属性は次のワイルドカード文字を持つパターンとして解釈されます。

- アスタリスク。0 個以上の文字を表します。
- 疑問符 (?)。ちょうど 1 文字を表します。

マッチングは、ファイル内で <sandbox> エlement がリストされている順序で実行されます。最初のマッチングのみが使用され、ファイル内にあるかもしれないそれ以降の他のマッチングはすべて無視されます。ファイルで指定された <sandbox> Element が、転送要求メッセージに関連付けられた MQMD ユーザーとマッチングしない場合、その転送はファイル・システムにアクセスできません。MQMD ユーザー名と user 属性の間にマッチングが検出された場合、そのマッチング項目を基にして、転送に適用される規則セットが <sandbox> Element 内で識別されます。この規則セットを使用して、転送の一環として読み取りまたは書き込みが可能であるファイルやデータ・セットが判別されます。

規則セットごとに、読み取り可能なファイルを識別する <read> Element および書き込み可能なファイルを識別する <write> Element を指定できます。規則セットから <read> または <write> Element を省略した場合、その規則セットに関連付けられたユーザーは、それぞれ読み取りまたは書き込みの実行を許可されないと想定されます。

注: UserSandboxes.xml ファイルの中で、<read> Element は <write> Element よりも前に、<include> Element は <exclude> Element よりも前に配置する必要があります。

<read> または <write> のそれぞれの Element には、ファイルがサンドボックス内にあるかどうか、転送可能であるかどうかを決定するために使用されるパターンが 1 つ以上含まれています。これらのパターンは、<include> および <exclude> Element を使用して指定します。<include> または <exclude> Element の name 属性は、突き合わせ対象となるパターンを指定します。オプションの type 属性は、名前値がファイルまたはキュー・パターンであるかを指定します。type 属性が指定されない場合、エージェントはパターンをファイルまたはディレクトリー・パス・パターンとして扱います。以下に例を示します。

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

エージェントは <include> および <exclude> name パターンを使用して、ファイル、データセット、またはキューの読み取りまたは書き込みが可能であるかどうかを判別します。操作は、正規のファイル・パス、データ・セット、またはキュー名が、包含パターンの少なくとも 1 つにマッチングし、除外パターンに 1 つもマッチングしない場合にのみ許可されます。<include> および <exclude> Element の name 属性を使用して指定するパターンには、エージェントを実行しているプラットフォームに適したパス分離文字および規則が使用されます。相対ファイル・パスを指定する場合、パスはエージェントの transferRoot プロパティを基準にして相対的に解決されます。

キューの制限を指定する場合、QUEUE@QUEUEMANAGER の構文がサポートされ、以下のルールが使用されます。

- アットマーク文字 (@) が項目から欠落している場合、パターンはいずれかのキュー・マネージャーでアクセスされるキュー名として扱われます。例えば、パターンが name である場合、name@** と同じように扱われます。
- アットマーク文字 (@) が項目の最初の文字である場合、パターンはキュー・マネージャー名として扱われ、キュー・マネージャーにあるすべてのキューにアクセスできます。例えば、パターンが @name である場合、**@name と同じように扱われます。

以下のワイルドカード文字は、<include> および <exclude> Element の name 属性の一部として指定した場合、特別な意味を持ちます。

*

単一のアスタリスクは、ディレクトリー名、またはデータ・セット名またはキュー名の修飾子の中の 0 個以上の文字と一致します。

?

疑問符 (?) は、ディレクトリー名、またはデータ・セット名かキュー名の修飾子の中の 1 文字にのみ一致します。

**

2つのアスタリスク文字は、ゼロ個以上のディレクトリー名、またはデータ・セット名またはキュー名のゼロ個以上の修飾子に一致します。また、パス分離文字で終わるパスには、パスの終わりに暗黙の "***" が追加されています。したがって、/home/user/ は /home/user/** と同じになります。

以下に例を示します。

- /**/test/** は、パス中に test ディレクトリーが含まれるすべてのファイルにマッチングします。
- /test/file? は、/test ディレクトリー内の、 スtring file で始まり任意の 1 文字が続くすべてのファイルにマッチングします。
- c:\test*.txt は、c:\test ディレクトリー内のすべてのファイルを .txt 拡張子で一致させます。
- c:\test***.txt は、'c:\test ディレクトリー内の任意のファイル、または .txt 拡張子を持つサブディレクトリーの 1 つに一致します。
-  // 'TEST.*.DATA' は、最初の修飾子が TEST で、任意の 2 番目の修飾子があり、さらに 3 番目の修飾子が DATA であるすべてのデータ・セットにマッチングします。
- *@QM1 は、単一修飾子を持つキュー・マネージャー QM1 上のすべてのキューと一致します。
- TEST.*.QUEUE@QM1 は、最初の修飾子が TEST で、任意の 2 番目の修飾子があり、さらに 3 番目の修飾子が QUEUE である、キュー・マネージャー QM1 にあるすべてのキューにマッチングします。
- **@QM1 は、キュー・マネージャー QM1 上の任意のキューと一致します。

シンボリック・リンク

UserSandboxes.xml ファイル内のファイル・パスで使用するシンボリック・リンクは、<include> および <exclude> エレメント内でハード・リンクを指定して、完全に解決する必要があります。例えば、/var が /SYSTEM/var にマップするシンボリック・リンクがある場合は、このパスを <tns:include name="/SYSTEM/var"/>として指定する必要があります。そうしないと、意図した転送はユーザー・サンドボックス・セキュリティ・エラーで失敗します。

例

この例は、以下の <sandbox> エレメントを AGENT_JUPITER の構成ディレクトリー内のファイル UserSandboxes.xml に追加することにより、MQMD ユーザー名 guest を持つユーザーが、エージェント AGENT_JUPITER が実行されているシステム上の /home/user/public ディレクトリーまたはそのサブディレクトリーから任意のファイルを転送できるようにする方法を示しています。

```
<?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>
```

例

次の例は、account に 1 つの数字が続く MQMD ユーザー名を持つユーザー (例: account4) に、以下のアクションの実行を許可する方法を示しています。

- /home/account ディレクトリーまたはそのいずれかのサブディレクトリー (エージェント AGENT_SATURN を実行しているシステム上の /home/account/private ディレクトリーを除く) から任意のファイルを転送します。

- エージェント AGENT_SATURN を実行しているシステム上の /home/account/output ディレクトリーまたはその任意のサブディレクトリーに、任意のファイルを転送します。
- 接頭部 ACCOUNT. で始まる、ローカル・キュー・マネージャーにあるキューからのメッセージを読み取ります。ただし、ACCOUNT.PRIVATE. で始まるものは除きます(つまり第 2 レベルに PRIVATE があるものです)。
- 任意のキュー・マネージャーにある接頭部 ACCOUNT.OUTPUT. で始まるキューにデータを転送します。

MQMD ユーザー名 account のユーザーがこれらのアクションを実行できるようにするには、AGENT_SATURN の構成ディレクトリーにあるファイル UserSandboxes.xml に以下の <sandbox> エレメントを追加します。

```
<?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>
```

関連資料

50 ページの『ワイルドカード転送の追加検査』

Continuous Delivery の IBM MQ 9.0.1 から、および Long Term Support の IBM MQ 9.0.0 Fix Pack 1 からは、エージェントがファイルを転送できる場所を制限するためにユーザー・サンドボックスまたはエージェント・サンドボックスを使用してエージェントが構成されている場合、そのエージェントのワイルドカード転送に対して追加の検査を行うように指定できます。

47 ページの『MFT ユーザー・サンドボックスの処理』

ファイルの転送先および転送元とすることが可能なファイル・システム内の領域を、転送を要求する MQMD ユーザー名に基づいて制限することができます。

550 ページの『MFT agent.properties ファイル』

各 Managed File Transfer Agent には、独自のプロパティー・ファイル agent.properties があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。agent.properties ファイルには、エージェントの動作を変更するプロパティーを含めることもできます。

V 9.0.1 ▶ V 9.0.0.1 ワイルドカード転送の追加検査

Continuous Delivery の IBM MQ 9.0.1 から、および Long Term Support の IBM MQ 9.0.0 Fix Pack 1 からは、エージェントがファイルを転送できる場所を制限するためにユーザー・サンドボックスまたはエージェント・サンドボックスを使用してエージェントが構成されている場合、そのエージェントのワイルドカード転送に対して追加の検査を行うように指定できます。

additionalWildcardSandboxChecking プロパティー

ワイルドカード転送の追加検査を有効にするには、検査を行うエージェントの agent.properties ファイルに、次のプロパティーを追加します。

```
additionalWildcardSandboxChecking=true
```

このプロパティが true に設定されている場合、ワイルドカードのファイル・マッチング用に定義されたサンドボックスの外側にある場所の読み取りを試行する転送要求がエージェントによって行われると、転送は失敗します。1つの転送要求内に複数の転送があり、サンドボックスの外側にある場所を読み取ろうとしたためにこれらの要求のいずれかが失敗した場合、転送全体が失敗します。検査が失敗すると、失敗の理由がエラー・メッセージに示されます。

additionalWildcardSandboxChecking プロパティがエージェントの agent.properties ファイルから省略されるか、false に設定されると、そのエージェントのワイルドカード転送で追加検査は行われません。

ワイルドカード検査のエラー・メッセージ

Continuous Delivery の IBM MQ 9.0.1 から、および Long Term Support の IBM MQ 9.0.0 Fix Pack 1 から、構成されたサンドボックス・ロケーションの外部のロケーションにワイルドカード転送要求が行われたときに報告されるメッセージが変更されました。

転送要求のワイルドカード・ファイル・パスが、制限されたサンドボックスの外にある場合、次のメッセージが表示されます。

BFGSS0077E: ファイル・パスを読み取ろうとしました: 「パス」 が拒否されました。
ファイル・パスは、制限された転送サンドボックスの外にあります。

複数転送要求内の転送に、制限されたサンドボックスの外にパスがあるワイルドカード転送要求が含まれている場合、次のメッセージが出されます。

BFGSS0078E: ファイル・パス path を読み取ろうとしましたが、別の転送として無視されました。
管理対象転送の項目が、制限された転送サンドボックスの外部で読み取ろうとしました。

制限された転送サンドボックスの外にファイルがある場合、次のメッセージが表示されます。

BFGSS0079E: ファイル file path を読み取ろうとしましたが、拒否されました。
ファイルは、制限された転送サンドボックスの外にあります。

次のメッセージは、別のワイルドカード転送要求が原因となりこの転送が無視された複数転送要求で表示されます。

BFGSS0080E: ファイル file path を読み取ろうとしましたが、別の転送として無視されました。
管理対象転送の項目が、制限された転送サンドボックスの外部で読み取ろうとしました。

ワイルドカードを含まない単一ファイル転送の場合、転送にサンドボックスの外にあるファイルが含まれているときに報告されるメッセージは前のリリースから変更されていません。

BFGI00056E で失敗: ファイル "FILE" を読み取ろうとしましたが、拒否されました。
ファイルは、制限された転送サンドボックスの外にあります。

関連資料

[47 ページの『MFT ユーザー・サンドボックスの処理』](#)

ファイルの転送先および転送元とすることが可能なファイル・システム内の領域を、転送を要求する MQMD ユーザー名に基づいて制限することができます。

[45 ページの『MFT エージェント・サンドボックスの処理』](#)

追加のセキュリティー・レベルを Managed File Transfer に加えるため、エージェントがアクセスできるファイル・システムの領域を制限することができます。

[550 ページの『MFT agent.properties ファイル』](#)

各 Managed File Transfer Agent には、独自のプロパティ・ファイル agent.properties があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。agent.properties ファイルには、エージェントの動作を変更するプロパティを含めることもできます。

MFT の SSL または TLS 暗号化の構成

IBM MQ Managed File Transfer で SSL または TLS を使用して、エージェントとエージェント・キュー・マネージャーの間の通信、接続先のコマンドとキュー・マネージャー、およびトポロジー内のさまざまなキュー・マネージャーとキュー・マネージャーの間の接続を保護することができます。

始める前に

SSL または TLS 暗号化を使用して、IBM MQ Managed File Transfer トポロジーを流れるメッセージを暗号化できます。これには以下が含まれます。

- エージェントとそのエージェント・キュー・マネージャーの間で受け渡されるメッセージ。
- 接続先のコマンドおよびキュー・マネージャーに関するメッセージ。
- トポロジー内のエージェント・キュー・マネージャー、コマンド・キュー・マネージャー、および調整キュー・マネージャーの間を流れる内部メッセージ。

このタスクについて

IBM MQ での SSL の使用に関する一般情報については、[SSL または TLS の取り扱い](#)を参照してください。IBM MQ の観点からすると、Managed File Transfer は、標準的な Java クライアント・アプリケーションです。

Managed File Transfer で SSL を使用するには、以下のステップを実行します。

手順

1. トラストストア・ファイルを作成し、オプションで鍵ストア・ファイルを作成します(これらのファイルは同じファイルにすることが可能です)。クライアント認証を必要としない場合(つまりチャンネル上で SSLAUTH=OPTIONAL)、鍵ストアを準備する必要はありません。トラストストアは、キュー・マネージャーの証明書を認証するためにのみ必要です。

IBM MQ で作業するには、トラストストアと鍵ストアの証明書を作成するために使用する鍵アルゴリズムが RSA でなければなりません。

2. SSL を使用するよう IBM MQ キュー・マネージャーをセットアップします。
例えば、IBM MQ Explorer を使用して SSL を使用するようキュー・マネージャーをセットアップする方法については、『[キュー・マネージャーでの SSL の構成](#)』を参照してください。
3. トラストストア・ファイルおよび鍵ストア・ファイル(存在する場合)を適切な場所に保存します。提案されている場所は `config_directory/coordination_qmgr/agents/agent_name` ディレクトリです。
4. 各 SSL 対応キュー・マネージャーの必要に応じて、SSL プロパティを Managed File Transfer の該当するプロパティ・ファイルに設定します。各プロパティ・セットは別個のキュー・マネージャー(エージェント、調整、およびコマンド)を参照します。ただし、1つのキュー・マネージャーがこれらの複数のロールを担う可能性はあります。

CipherSpec または **CipherSuite** プロパティのいずれかが必要です。ない場合にはクライアントは SSL を使用せずに接続を試行します。IBM MQ と Java では用語が異なるため、**CipherSpec** と **CipherSuite** の両方のプロパティが提供されています。Managed File Transfer は、どちらのプロパティも受け入れて必要な変換を行うため、両方のプロパティを設定する必要はありません。

CipherSpec と **CipherSuite** の両方のプロパティを指定した場合は、**CipherSpec** が優先されます。

PeerName プロパティはオプションです。このプロパティを、接続先キュー・マネージャーの識別名に設定できます。Managed File Transfer は、識別名が一致しない不正確な SSL サーバーへの接続をリジェクトします。

SslTrustStore および **SslKeyStore** プロパティを、トラストストア・ファイルおよび鍵ストア・ファイルを指すファイル名に設定します。これらのプロパティを既に実行中のエージェントに対してセットアップする場合、エージェントを停止してから再開し、SSL モードで再接続します。

プロパティ・ファイルにはプレーン・テキスト・パスワードが含まれるため、ファイル・システムの適切な許可を設定することを考慮してください。

SSL プロパティについて詳しくは、[605 ページの『MFT の SSL プロパティ』](#)を参照してください。

5. エージェントのキュー・マネージャーが SSL を使用する場合、そのエージェントを作成するときに必要な詳細を提供することはできません。そのエージェントを作成するには次のステップを実行します。

- a) **fteCreateAgent** コマンドを使用してエージェントを作成します。エージェントの存在を調整キュー・マネージャーにパブリッシュできないことに関する警告を受け取ります。
 - b) 直前のステップで作成された **agent.properties** ファイルを編集し、SSL 情報を追加します。エージェントが正常に開始すると、パブリッシュが再度試行されます。
6. エージェントまたは IBM MQ Explorer・インスタンスの実行中に、**agent.properties** ファイルまたは **coordination.properties** ファイルで SSL プロパティを変更した場合は、そのエージェントまたは IBM MQ エクスプローラーを再始動する必要があります。

関連資料

550 ページの『MFT agent.properties ファイル』

各 Managed File Transfer Agent には、独自のプロパティ・ファイル **agent.properties** があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。**agent.properties** ファイルには、エージェントの動作を変更するプロパティを含めることもできます。

クライアント・モードでチャンネル認証を使用してキュー・マネージャーに接続する操作

チャンネル・レベルでアクセスをより正確に制御するために、IBM WebSphere MQ 7.1 ではチャンネル認証レコードが導入されました。動作がこのように変更されたことにより、新しく作成された IBM WebSphere MQ 7.1 以降のキュー・マネージャーは、Managed File Transfer コンポーネントからのクライアント接続をデフォルトで拒否します。

チャンネル認証について詳しくは、[チャンネル認証レコード](#)を参照してください。

Managed File Transfer によって使用される SVRCONN のチャンネル認証構成が非特権 MCAUSER ID を指定している場合は、Managed File Transfer Agent とコマンドが正しく動作するように、キュー・マネージャー、キュー、およびトピックに特定の権限レコードを付与する必要があります。チャンネル認証レコードを作成、変更、または削除するには、MQSC コマンド **SET CHLAUTH** または PCF コマンド **Set Channel Authentication Record** を使用します。IBM WebSphere MQ 7.1 以降のキュー・マネージャーに接続するすべての Managed File Transfer エージェントに対して、すべてのエージェントに共通の 1 つの MCAUSER ID をセットアップすることもできますし、エージェントごとに異なる MCAUSER ID を個別にセットアップすることもできます。

各 MCAUSER ID に以下の権限を付与します。

- キュー・マネージャーに必要な権限レコード:

- connect
- setid
- inq

- キューに必要な権限レコード:

すべてのエージェント固有キュー (以下のリストでキュー名の最後が **agent_name** になっているもの) に関して、クライアント接続を使用して IBM WebSphere MQ 7.1 以降のキュー・マネージャーに接続するエージェントごとに、これらのキュー権限レコードを作成する必要があります。

- 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)

- トピックに必要な権限レコード:

- sub, pub (SYSTEM.FTE)

- ファイル転送に必要な権限レコード。

ソース・エージェントと宛先エージェントで MCAUSER ID が異なる場合には、ソースと宛先の両方のエージェント・キューに対して権限レコードを作成します。

例えば、ソース・エージェントの MCAUSER ID が **user1** で、宛先エージェントの MCAUSER ID が **user2** の場合、それぞれのエージェント・ユーザーに対して以下の権限を設定します。

エージェント・ユーザー	キュー	必要な権限
user1	SYSTEM.FTE.DATA.destination_agent_name	put
user1	SYSTEM.FTE.COMMAND.destination_agent_name	put
user2	SYSTEM.FTE.REPLY.source_agent_name	put
user2	SYSTEM.FTE.COMMAND.source_agent_name	put

Connect:Direct ブリッジ・エージェントと Connect:Direct ノードの間の SSL または TLS の構成

Connect:Direct ブリッジ・エージェントと Connect:Direct ノードが SSL プロトコルを使用して相互に接続するように構成します。そのためには、鍵ストアとトラストストアを作成し、Connect:Direct ブリッジ・エージェントのプロパティ・ファイルでプロパティを設定します。

このタスクについて

ここでは、認証局から鍵の署名を得るための手順を含めています。認証局を使用しない場合は、自己署名証明書を生成できます。自己署名証明書を生成する方法の詳細については、[UNIX および Windows システムでの SSL または TLS の取り扱い](#)を参照してください。

ここでは、Connect:Direct ブリッジ・エージェントの新しい鍵ストアとトラストストアを作成するための手順を含めています。Connect:Direct ブリッジ・エージェントに、IBM MQ キュー・マネージャーへのセキュア接続で使用できる鍵ストアとトラストストアが既にある場合は、Connect:Direct ノードへのセキュア接続で既存の鍵ストアとトラストストアを使用できます。詳細については [51 ページの『MFT の SSL または TLS 暗号化の構成』](#)を参照してください。

手順

Connect:Direct ノードの場合、以下のステップを実行します。

1. Connect:Direct ノードの鍵と署名付きの証明書を生成します。
そのためには、IBM MQ に用意されている IBM 鍵管理ツールを使用できます。詳細については、[SSL または TLS の取り扱い](#)を参照してください。
2. 鍵の署名を得るための要求を認証局に送信します。返ってくる証明書を受け取ります。
3. 認証局の公開鍵を組み込んだテキスト・ファイル (例えば /test/ssl/certs/CAcert) を作成します。
4. Connect:Direct ノードに Secure+ オプションをインストールします。
ノードが既に存在している場合は、インストーラーを再び実行し、既存のインストール環境の場所を指定し、Secure+ オプションだけのインストールを選択することによって、Secure+ オプションをインストールできます。
5. 新しいテキスト・ファイル (例えば /test/ssl/cd/keyCertFile/node_name.txt) を作成します。
6. 認証局から受け取った証明書と、/test/ssl/cd/privateKeys/node_name.key にある秘密鍵をそのテキスト・ファイルにコピーします。

/test/ssl/cd/keyCertFile/node_name.txt の内容は、以下の形式になっていなければなりません。

```
-----BEGIN CERTIFICATE-----
MIIcZnCCAgigAwIBAgIBGjANBgkqhkiG9w0BAQUFADBeMQswCQYDVQQGEWJHJqJES
```

```

MBAGA1UECBMJSJGFtCHNoaXJlMRaWdYdVQqHEwdIdXJzbGV5M0wwCgYdVQqKEwNJ
Qk0xDjAMBgNVBAStBU1RSVBUmQswCQYdVQqDEwJDQTAeFw0xMTAzMDEeXnJlWnDZa
Fw0yMTAyMjYxNjIwNDZaMFAxCzAJBgNVBAYTAkdCMRlWEAyDQqIEw1IYw1wc2hp
cmUxDDAKBgNVBA0TA0lCTTEOMAwGA1UECXMFTVtVGVUeXZzANBgNVBAMTBmJpbmJh
ZzCBnzANBgkqhkiG9w0BAQEFAAQBJQAwgYkCgYEAvgP1QIk1U9ypSKD1Xo0Do1yk
EyMFXB0UpZr1dVXjoSEC0vtWncJ199e+Vc4UpNybdyBu+Nkd1MNOF4QxeQcLAFj
WnhakqCiQ+JIAD5AurhnwChe0MV3kjA84GKH/r0SVqt1984mu/1DyS819XcfSSn
c00MsK1KbneVSCIV2XECaWEAAAn7MHkwCQYdVROTBAlWADAsBg1ghkgBhvCAQ0E
HxYdT3B1b1NTTCBHZW51cmF0ZwQgQ2VydG1maWNoGDUwHQYdVRO0BBYEFNXMIpSc
csBXUniW4A3Uzr3CRsv3MB8GA1UdIwQYMBaAFDXY8rmj41Vz5+FVAoQb++cns+B4
MA0GCSqGSIB3DQEBBQUAA4GBAFc7k1Xa4pGKYgwchxKpE3ZF6FNWY4vBXS216/ja
8h/v18+iv010C18t0ZOKSU95fyZLzOPkNCH7v+ItFSE3CIIEk9D1z2U6W091ICwn
17PL72Tdfal3kabwHYVf17IVcuL+VZsZ3HjLggP2qH09ZuJPspeT9+AxFVMLiaAb
8eHw
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,64A02DA15B6B6EF9

```

```

57kqxL0J/gRU0IQ6hVK2YN13B4E1jAi1gSme0I5ZpEIG8CHXISKB7/0cke2FTqsV
lvI99QyCxsDwoMnt5fj51v7aPmVeS60b0m+U1Gre8B/Ze18JVj204K2Uh72rDCXE
5e6eFxDuM207sQDy20euBVELJtM2k0kL1R0doQqS1U3XQNgJw/t3ZIx5hPXWEQT
rjRQ064BEhb+PzzxPF8uwzZ9IzUK9BJ/UUnqC6OdBR87IeA4pnJD1Jvb2ML7EN9Z
5Y+50hTKI80GvBvWX04fHyvIX5aslwhBoArXIS1AtNTtrptPvoaP1zyIAeZ60Cvo/
Sfo+A2UhmTEJe0JaZG2XZ3H495fAw/EHmjehzIACwukQ9nSIETgu4A1+CV64RJED
aYBCM8UjaAKbZDH5gn7+eBov0ssXAXWdyJBVhU0jXjvAj/e1h+kcSF1hax5D//AI
66nRMZzboSxNqkjcVd8wfdwP+bEjDzUaaarJTS71IFeLlw7eJ8MNAkMGicDkycL0
EPBU9X5QnHKLK0YHNN/1WgUk8qt3UytFXXfzTXGF3EbsWbBupkT5e5+1YcX80VZ6
sHFPN1H1uCNy/riUcBy9iviVeodX8Iom0chSy05DK18bwZnjYtUP+CtYHNFU5BaD
I+1uU0AeJ+wjQYKT1WaeIGZ3VxuNITJJul8y5qDTXXfX7vxM50oWxa6U5+AYuGUMg
/itPZmUmNzHjTk7ghT6i1IQ0aBowXXKJB1Mmq/6BQXN2IhkD9ys2qrvM1hdi5nAf
egmdiG50LolnBRqWbfr+DykpAhK4SaDi2F52Uxovw3Lhwi8dQP7lzQ==
-----END RSA PRIVATE KEY-----

```

7. Secure+ 管理ツールを開始します。

- Linux または UNIX のシステムでは、**spadmin.sh** コマンドを実行します。
- Windows のシステムでは、「スタート」 > 「プログラム」 > 「Sterling Commerce Connect:Direct」 > 「CD Secure+ 管理ツール」をクリックします。

CD Secure+ 管理ツールが開始します。

8. CD Secure+ 管理ツールで、**.Local** の行をダブルクリックして、SSL または TLS のメイン設定を編集します。

- 使用するプロトコルに応じて、「**SSL プロトコルを有効にする**」または「**TLS プロトコルを有効にする**」を選択します。
- 「**オーバーライドを無効にする**」を選択します。
- 少なくとも 1 つの暗号スイートを選択します。
- 双方向認証を設定する場合は、「**クライアント認証を有効にする**」の値を「はい」に変更します。
- 「**トラステッド・ルート証明書**」フィールドに、認証局の公開証明書ファイルのパス (/test/ssl/certs/CAcert) を入力します。
- 「**鍵証明書ファイル**」フィールドに、作成したファイルのパス (/test/ssl/cd/keyCertFile/node_name.txt) を入力します。

9. **.Client** の行をダブルクリックして、SSL または TLS のメイン設定を編集します。

- 使用するプロトコルに応じて、「**SSL プロトコルを有効にする**」または「**TLS プロトコルを有効にする**」を選択します。
- 「**オーバーライドを無効にする**」を選択します。

Connect:Direct ブリッジ・エージェントの場合は、以下の手順を実行します。

10. トラストストアを作成します。そのためには、ダミーの鍵を作成してから、そのダミーの鍵を削除します。

以下のコマンドを使用できます。

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

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

11. 認証局の公開証明書をトラストストアにインポートします。
以下のコマンドを使用できます。

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

12. Connect:Direct ブリッジ・エージェントのプロパティ・ファイルを編集します。
ファイルの任意の場所に以下の行を組み込みます。

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

この手順の例では、*protocol* は使用するプロトコル (SSL または TLS) で、*password* はトラストストアの作成時に指定したパスワードです。

13. 双方向認証を設定する場合は、Connect:Direct ブリッジ・エージェントの鍵と証明書を作成します。
 - a) 鍵ストアと鍵を作成します。
以下のコマンドを使用できます。

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

- b) 署名要求を生成します。
以下のコマンドを使用できます。

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

- c) 前の手順で受け取った証明書を鍵ストアにインポートします。証明書は、x.509 形式でなければなりません。
以下のコマンドを使用できます。

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

- d) Connect:Direct ブリッジ・エージェントのプロパティ・ファイルを編集します。
ファイルの任意の場所に以下の行を組み込みます。

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

この手順の例では、*password* は鍵ストアの作成時に指定したパスワードです。

関連タスク

[143 ページの『Connect:Direct ブリッジの構成』](#)

Managed File Transfer ネットワークと Connect:Direct ネットワークの間でファイルを転送するために、Connect:Direct ブリッジを構成します。Connect:Direct ブリッジのコンポーネントは、Connect:Direct ノードと、そのノードと通信するための専用の Managed File Transfer エージェントです。このエージェントのことを Connect:Direct ブリッジ・エージェントといいます。

MFT の構成

Managed File Transfer のフィーチャーは、インストール後に構成可能です。

関連概念

[256 ページの『ユーザー出口での MFT のカスタマイズ』](#)

Managed File Transfer のフィーチャーは、ユーザー出口ルーチンと呼ばれる独自のプログラムを使用してカスタマイズできます。

[246 ページの『MFT のリカバリーと再始動』](#)

エージェントまたはキュー・マネージャーが何らかの理由 (例えば、電源やネットワークの障害など) で使用できない場合、Managed File Transfer は、以下のシナリオで示すようにリカバリーを行います。

[272 ページの『MFT の使用のヒント』](#)

Managed File Transfer を最大限に活用するために役立ついくつかの提案を以下に示します。

[154 ページの『管理 MFT』](#)

Managed File Transfer を管理するには、Managed File Transfer コマンドを使用します。また、IBM MQ Explorer を使用して管理用タスクの一部を行うこともできます。

関連タスク

[170 ページの『MFT リソースのモニター』](#)

キューやディレクトリーなどの Managed File Transfer リソースをモニターできます。そのリソースで条件が満たされると、リソース・モニターがファイル転送などのタスクを開始します。 `fteCreateMonitor` コマンドを使用するか、IBM MQ Explorer に対応した Managed File Transfer プラグインの「モニター」ビューを使用して、リソース・モニターを作成できます。

[43 ページの『マルチプラットフォームでの MQMFTCredentials.xml の構成』](#)

Managed File Transfer (MFT) がセキュリティーを有効にして構成されている場合、接続認証では、キュー・マネージャーに接続するすべての MFT コマンドでユーザー ID とパスワードの資格情報を提供する必要があります。同様に、MFT ロガーは、データベースへの接続時にユーザー ID とパスワードを指定する必要があります。この資格情報は、MFT 資格情報ファイルに保管できます。

関連資料

[373 ページの『MFT コマンド』](#)

Managed File Transfer のすべてのコマンドがリストされており、その詳しい説明へのリンクも掲載されています。

[41 ページの『MFT の保護』](#)

インストール直後の変更のない状態では、Managed File Transfer のセキュリティー・レベルは、保護された環境におけるテストまたは評価のためには適している可能性があります。ただし、実稼働環境では、ファイル転送操作を開始できるユーザー、転送されているファイルの読み取りおよび書き込みができるユーザー、およびファイルの保全性を保護する方法の適切な管理について考慮する必要があります。

[550 ページの『MFT agent.properties ファイル』](#)

各 Managed File Transfer Agent には、独自のプロパティー・ファイル `agent.properties` があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。 `agent.properties` ファイルには、エージェントの動作を変更するプロパティーを含めることもできます。

[252 ページの『MFT での Apache Ant の使用』](#)

Managed File Transfer は、ファイル転送機能を Apache Ant ツールに組み込むために使用できるタスクを提供します。

[270 ページの『トラブルシューティング MFT』](#)

以下の参照情報を、Managed File Transfer のエラーの診断に役立ててください。

MFT 構成オプション (Multiplatforms)

Managed File Transfer は、セットアップに関する重要な情報が含まれた、操作に必要ないくつかのプロパティー・ファイルを提供します。これらのプロパティー・ファイルは、製品のインストール時に定義される構成ディレクトリーにあります。

それぞれがディレクトリーとプロパティー・ファイルのセットを含む構成オプションのセットを複数持つことができます。コマンド行に別の値を明示的に指定しない限り、すべての Managed File Transfer コマンドについて、これらのプロパティー・ファイルで定義されている値がデフォルトのパラメーターとして使用されます。

使用中のデフォルトの構成オプションのセットを変更するには、**fteChangeDefaultConfigurationOptions** コマンドを使用します。個々のコマンドに使用している構成オプションのセットを変更するには、Managed File Transfer コマンドで **-p** パラメーターを使用します。

構成オプションのセットの名前は、調整キュー・マネージャーの名前です。この名前は変更しないことをお勧めします。構成オプションのセットの名前を変更することは可能ですが、config ディレクトリーおよび logs ディレクトリーの名前を変更する必要があります。以下の例では、構成オプションのセットの名前を *coordination_qmgr_name* と表記します。

構成オプションのディレクトリー構造

製品を構成すると、構成ディレクトリー内にディレクトリーとプロパティー・ファイルが以下に示す構造で作成されます。以下のコマンドを使用すると、ディレクトリーとプロパティー・ファイルを変更することもできます。**fteSetupCoordination**、**fteSetupCommands**、**fteChangeDefaultConfiguration**、および **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 ディレクトリーは、構成オプションのディレクトリーです。構成ディレクトリー内に複数の構成オプション・ディレクトリーを置くことができます。*agent_name* ディレクトリーはエージェント・ディレクトリーです。このディレクトリーには *agent.properties* ファイルの他に *exits* ディレクトリーが含まれており、これはユーザー・出口ルーチン、および **fteCreateBridgeAgent** コマンドおよび **fteCreateCDAgent** コマンドで生成されるさまざまな XML ファイルのためのデフォルトの場所です。*agents* ディレクトリーには、構成オプションのセットのためのエージェント・ディレクトリーを複数置くことができます。

プロパティー・ファイル

installation.properties

installation.properties ファイルは、構成オプションのデフォルト・セットの名前を指定します。この項目は、Managed File Transfer を、使用する構成が含まれる一連の構造化されたディレクトリーおよびプロパティー・ファイルに向けます。一般に、構成オプションのセット名は、関連付けられている調整キュー・マネージャーの名前です。*installation.properties* ファイルについて詳しくは、[536 ページの『MFT installation.properties ファイル』](#)を参照してください。

coordination.properties

coordination.properties ファイルは、調整キュー・マネージャーへの接続の詳細を指定します。複数の Managed File Transfer インストール済み環境が同じ調整キュー・マネージャーを共有する可能性があるため、共有ドライブ上の共通 *coordination.properties* ファイルへのシンボリック・リンクを使用できます。*coordination.properties* ファイルについて詳しくは、[540 ページの『MFT coordination.properties ファイル』](#)を参照してください。

command.properties

command.properties ファイルは、コマンド発行時に接続先となるコマンド・キュー・マネージャーと、そのキュー・マネージャーに接続するために Managed File Transfer が必要とする情報を指定します。command.properties ファイルについて詳しくは、[546 ページの『MFT command.properties ファイル』](#)を参照してください。

agent.properties

各 Managed File Transfer Agent には、独自のプロパティ・ファイル agent.properties があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。agent.properties ファイルには、エージェントの動作を変更するプロパティを含めることもできます。agent.properties ファイルについて詳しくは、[550 ページの『MFT agent.properties ファイル』](#)を参照してください。

logger.properties

logger.properties ファイルは、ロガーの構成プロパティを指定します。logger.properties ファイルについて詳しくは、[118 ページの『MFT ロガーの構成プロパティ』](#)を参照してください。

プロパティ・ファイルとコード・ページ

Java の制限のために、すべての Managed File Transfer プロパティ・ファイルの内容は米国英語のままにする必要があります。米国英語以外のシステムでプロパティ・ファイルを編集する場合、Unicode エスケープ・シーケンスを使用する必要があります。

z/OS での MFT 構成オプション

z/OS での Managed File Transfer 構成オプションは、分散プラットフォームの場合のオプションと同じです。

[マルチプラットフォームの構成オプションの詳細については、57 ページの『MFT 構成オプション \(Multiplatforms\)』](#)を参照してください。

z/OS では、構成の場所が BFG_DATA 環境変数によって定義されます。BFG_DATA によって参照される UNIX System Services ディレクトリーの下に構成がまだ存在しない場合は、MFT コマンド PDSE ライブラリー・データ・セットの BFGCUSTM JCL スクリプトによって、構成を作成するために必要なジョブが生成されます。次いで、生成されたこれらのジョブを実行すると、構成が作成されます。アクセス可能な既存のディレクトリーを参照する BFG_DATA に、構成の作成は依存しています。

Multiplatforms 版と z/OS 版の両方に対応した同じ **fte** コマンドを使用して、構成を作成したり管理したりすることもできます。**fte** コマンドのリストについては、[373 ページの『MFT コマンド』](#)を参照してください。

V 9.0.1 Redistributable Managed File Transfer Agent の構成

オプションで提供される Redistributable Managed File Transfer Agent を、既存の IBM MQ インフラストラクチャーに接続するように構成し、IBM MQ をインストールしていないユーザーが Managed File Transfer 機能を利用してファイルを転送できるようにします。

始める前に

Redistributable Managed File Transfer Agent は Managed File Transfer の機能を提供しますが、次の例外があります。

- 調整キュー・マネージャー、コマンド・キュー・マネージャー、およびエージェント・キュー・マネージャーへのバインディング・モード接続はサポートされないため、クライアント・モード接続を使用する必要があります。コマンドを発行するときは、IBM MQ の一部としてインストールされた Managed File Transfer を使用する場合はオプションとなるパラメーター (キュー・マネージャーのホスト、ポート、名前、およびチャンネル名) を指定する必要があります。
- 以下のコマンドは、含まれていません。
 - fteCreateCDAgent.cmd
 - fteCreateLogger.cmd

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

使用可能コマンドの完全なリストについては、[インストールされる MFT コマンド・セットを参照してください](#)。

- Managed File Transfer Connect:Direct は、サポートされていません。
- IBM MQ Explorer は、含まれていません。

Windows Redistributable Managed File Transfer Agent を使用するには、次の Microsoft ライブラリーをシステムにインストールする必要があります。

- Microsoft Visual C++ 再頒布可能パッケージ 2008
- Microsoft Visual C++ 再頒布可能パッケージ 2012

これらのライブラリーは、Microsoft から入手できます。[The latest supported Visual C++ downloads](#) を参照してください。

このタスクについて

ユーザーが IBM MQ をインストールしなくてもローカル環境と既存の IBM MQ インフラストラクチャーの間でファイル転送を行えるようにしたい場合があります。次の手順に従って、Redistributable Managed File Transfer Agent をダウンロードして解凍します。

手順

1. Redistributable Managed File Transfer Agent パッケージを [IBM Fix Central](#) からダウンロードします。

a) 使用するオペレーティング・システム用のパッケージを選択します。

- **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) パッケージの解凍先ディレクトリーを決定します。以下に例を示します。

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

2. ダウンロードしたパッケージのコンテンツを解凍します。

- **Windows** Windows エクスプローラー・ツールを使用して解凍します。
- **Linux** 次のように解凍して untar します。

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

それから、次のようにします。

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

以下のディレクトリーが作成されます。

- **Windows** **Linux** bin:必要なすべての MFT コマンドが含まれます
- **Windows** bin64: Windows 64 ビットの OS のサポートに必要なライブラリーが含まれます
- **Windows** **Linux** java:IBMJRE および IBM MQ ライブラリーが含まれています
- **Windows** **Linux** licenses: ライセンス・ファイルが含まれています
- **Windows** **Linux** mqft: Ant サポートおよびコア MFT 機能サポートに必要な ant および lib ディレクトリーが含まれています
- **Windows** **Linux** swtag: ライセンス・マネージャーがマシン上のこのインストール環境を識別するために必要な swidtag ファイルが含まれています

次のタスク

これで、MFT エージェントを構成する準備が整いました。次の手順については、次のトピックを参照してください。[61 ページの『初期構成の作成』](#)

関連資料

[321 ページの『再配布可能 MFT エージェントの構成時に起こる可能性のあるエラー』](#)
 Redistributable Managed File Transfer Agent の構成時のエラー・メッセージ

関連情報

[ビデオ: How to install and configure the Redistributable Managed File Transfer Agent \(YouTube\)](#)

初期構成の作成

Managed File Transfer Agent を、既存の IBM MQ 構成に接続するように構成できます。

始める前に

V 9.0.1 Redistributable Managed File Transfer Agent ・パッケージのコンテンツをダウンロードして解凍します。詳細については [59 ページの『Redistributable Managed File Transfer Agent の構成』](#) を参照してください。

このタスクについて

IBM MQ サーバーで実行されているキュー・マネージャーとの接続をセットアップするためのコマンドを使用できる環境を作成します。

手順

1. Redistributable Managed File Transfer Agent の環境を作成します。

fteCreateEnvironment コマンドを実行すると、MFT Agent の構成情報を使用して MFT データ・ディレクトリーが作成されます。ダウンロードした MFT Agent 再配布可能コンポーネントを解凍したときに作成された bin ディレクトリーにいることを確認します。以下のコマンドを実行します。

- **Windows**

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

- **Linux**

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

このコマンドは、次の 1 つのオプション・パラメーターを取ります。

-d

MFT 構成を作成、保管、および保守するデータ・パスの場所。ユーザーがデータの場所を指定せずに **fteCreateEnvironment** を実行した場合は、Redistributable Managed File Transfer Agent が解凍された場所にディレクトリー `mftdata` が作成されます。

注: 再配布可能エージェントが Windows サービスとして実行される場合、このサービスを機能させるには、システム環境で **BFG_DATA** 環境変数を設定する必要があります。

次のようにデータ・パスの場所を指定して環境変数 `BFG_DATA` を設定することもできます。

```
BFG_DATA=Datapath location
```

エージェントの作成、開始、停止、およびそのほかのコマンドを実行する前に、`BFG_DATA` 変数に正しいデータ・パスの場所が設定されていることを確認してください。

2. IBM MQ 接続をセットアップします。

a) **fteSetupCoordination** コマンドを使用して調整キュー・マネージャーをセットアップします。

fteSetupCoordination コマンドは、調整キュー・マネージャーに必要なセットアップ、および詳細な構成に必要なディレクトリーを作成します。Redistributable Managed File Transfer Agent はクライアント・モードで機能します。バインディング・モードはサポートされないため、エラーを回避するためにこのコマンドには追加のパラメーターを指定する必要があります。

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

fteSetupCoordination コマンドの使用に関する詳細と手順については、[514 ページの『fteSetupCoordination』](#)を参照してください。調整キュー・マネージャーの構成方法については、[98 ページの『MFT 用の調整キュー・マネージャーの構成』](#)を参照してください。

b) コマンド・キュー・マネージャーを作成し、セットアップします。

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

fteSetupCommands コマンドの使用に関する詳細と手順については、[512 ページの『fteSetupCommands: MFT command.properties ファイルの作成』](#)を参照してください。

3. エンドポイントに MFT エージェント定義を作成します。

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

fteCreateAgent コマンドを使用してエージェントおよびエージェント・キュー・マネージャーを構成する方法について詳しくは、[388 ページの『fteCreateAgent \(MFT エージェントの作成\)』](#)を参照してください。

エージェントごとに、手順 [62 ページの『2』](#) および [62 ページの『3』](#) で、エージェント・キュー・マネージャーのキューとトピックの定義を作成することになります。

4. エージェントを開始すれば、ファイルを転送するための準備は完了です。

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

以下のコマンドを実行して、エージェントの状況を確認できます。

```
ftelistAgents
```

ftelistAgents コマンドの使用方法について詳しくは、[473 ページの『ftelistAgents: 調整キュー・マネージャーの MFT エージェントのリスト』](#)を参照してください。

関連概念

[57 ページの『MFT の構成』](#)

Managed File Transfer のフィーチャーは、インストール後に構成可能です。

57 ページの『MFT 構成オプション (Multiplatforms)』

Managed File Transfer は、セットアップに関する重要な情報が含まれた、操作に必要ないくつかのプロパティ・ファイルを提供します。これらのプロパティ・ファイルは、製品のインストール時に定義される構成ディレクトリーにあります。

関連資料

437 ページの『fteCreateTransfer: 新規ファイル転送の開始』

fteCreateTransfer コマンドは、コマンド行から新規のファイル転送を作成して開始します。このコマンドは、ファイル転送の即時開始、将来の日時におけるファイル転送のスケジュール、ファイル転送の 1 回以上の繰り返し、および特定の条件に基づくファイル転送の起動を実行できます。

z/OS MFT エージェントまたはロガーのコマンド・データ・セットの作成

特定の調整のため、特定の Managed File Transfer Agent または Managed File Transfer Logger 用の Managed File Transfer コマンド・テンプレート・データ・セットから、コマンドの PDSE データ・セットを作成することができます。

このタスクについて

以下のステップを完了します。

手順

1. MFT コマンド・テンプレート PDSE ライブラリー・データ・セット SBFGCMD5 のコピーを作成します。

z/OS SBFGCMD5 は新しいライブラリー (*prefix.agent.JCL_* など) にコピーする必要があります。SBFGCMD5(BFGCOPY) メンバーに以下の置換を行った更新済みバージョンを使用することができます。

- `++supplied-library++` を SBFGCMD5 PDSE の完全修飾名に置き換えます。
- **z/OS** `++service-library++` を、新規 MFT コマンド PDSE ライブラリー・データ・セットの完全修飾名に置き換えます。 `++service-library++` は、作成されるエージェントまたはロガー・サービス用の出力データ・セットです。

2. 新規 MFT コマンド PDSE ライブラリー・データ・セットについて、メンバー BFGCUSTOM を編集します。これは、エージェントまたはロガー用にコマンドをカスタマイズするための JCL スクリプトです。各変数は `++variable name++` という形式で指定されます。これを必要な値に置き換える必要があります。各種の JCL 変数の説明については、77 ページの『z/OS JCL 変数』を参照してください。BFGSTDIN DD ステートメントでは、変数を Variables、Properties、および Environment という 3 つのカテゴリーで定義します。ステートメントの形式は以下のとおりです。

```
[Variables]
variable1=value1
variable2=value2
....
variableN=valueN
[Properties]
property1=property value1
property2=property value2
...
propertyN=property valueN
[Environment]
custom_variable1=value1
custom_variable2=value2
....
custom_variableN=valueN
```

Variables は、各コマンドに必要なセットアップと環境変数のセットを定義します。

Properties は、MFT 構成プロパティの指定変更を定義します。ご使用の環境に合わせてエージェントまたはロガーをカスタマイズするために、必要に応じて、エージェントおよびロガーのプロパティを追加できます。すべてのプロパティのリストについては、90 ページの『構成プロパティ・ファイ

ル』を参照してください。この機能は、MFT 構成プロパティ・ファイル (UNIX System Services ファイルとして保守される) にアクセスしなくても済むようにするために用意されています。

Environment では、追加で必要なカスタム環境変数を任意に定義します。

- 新規 MFT コマンド PDSE ライブラリー・データ・セットに対して、ジョブ BFGCUSTM を実行依頼します。このジョブは、エージェントまたはロガーに適切な JCL コマンドのセットを、PDSE の新規メンバーとして生成します。コマンドの完全なリストについては、82 ページの『z/OS エージェントおよびロガー・コマンドの JCL スクリプト』を参照してください。

ジョブ BFGCUSTM は、DISP=OLD の DD ステートメントを含む JCL が入っているライブラリーを更新します。実行依頼の後にエディターを終了して、ジョブが実行できるようにしなければなりません。

出力ジョブ・ログを調べて、JCL スクリプトが正常に実行されたことを確認してください。何らかの障害がある場合には、それらを修正して BFGCUSTM ジョブを再び実行依頼してください。

さらに BFGCUSTM JCL スクリプトは、UNIX System Services MFT 構成プロパティ・ファイルがそれぞれ整合するように、必要に応じてそれらのファイルを更新します。CoordinationQMGr プロパティによって定義される構成が存在しない場合には、警告メッセージが出力されます。その場合、生成された BFGCFGR および BFGCMCR のジョブを実行して、構成プロパティ・ファイルを作成する必要があります。エージェントの編集の場合は BFGAGCR、ロガーの編集の場合は BFGLGCRS を実行する必要があります。指定された構成が既に存在する場合には、BFTCUSTM JCL スクリプトで定義されているすべてのプロパティにより、構成が更新されます。

関連概念

59 ページの『z/OS での MFT 構成オプション』

z/OS での Managed File Transfer 構成オプションは、分散プラットフォームの場合のオプションと同じです。

関連タスク

76 ページの『MFT の既存のエージェントまたはロガーのコマンド・データ・セットの更新 (z/OS)』

Managed File Transfer コマンド・テンプレート・データ・セットから作成された Managed File Transfer コマンド PDSE ライブラリー・データ・セットを更新できます。

z/OS Managed File Transfer for z/OS の構成

Managed File Transfer for z/OS では、コンポーネントを正常に動作させるためのカスタマイズが必要です。

このタスクについて

以下の作業が必要です。

- PDSE メンバーを編集して構成データを指定します
- 調整キュー・マネージャーを定義します
- コマンド・キュー・マネージャーを定義します
- 1 つ以上のエージェントを構成します
- オプション: データを Db2 に保管するようロガー・タスクを構成します

実行する必要があるタスクの順序については、以下のトピックで詳しく説明されています。

関連概念

64 ページの『MFT 構成の確認』

開始する前に、システムの構成を確認する必要があります。

関連情報

のインストール Managed File Transfer for z/OS

z/OS MFT 構成の確認

開始する前に、システムの構成を確認する必要があります。

Managed File Transfer (MFT) を使用するには、定義した MFT 構成ごとに以下の役割で動作するキュー・マネージャーが1つ以上必要です。

- 調整キュー・マネージャー。このキュー・マネージャーは、コーディネーター上のトピックにパブリッシュされた、構成内の各エージェントの状況に関する情報を維持します。
- 1つ以上のコマンド・キュー・マネージャーまたは接続キュー・マネージャー。このキュー・マネージャーは、MFT コマンドを使用するための IBM MQ ネットワークへの入り口点として機能します。
- 1つ以上のエージェント・キュー・マネージャー。このキュー・マネージャーは、MFT エージェントと IBM MQ ネットワークの間の通信を提供します。

上記の役割は、それぞれに別々のキュー・マネージャーで実行することも、すべての役割をまとめて1つのキュー・マネージャーで実行する (最も単純な構成) こともできます。

既存の MFT 環境に z/OS キュー・マネージャーを追加する場合は、z/OS キュー・マネージャーと構成内の他のキュー・マネージャーの間の接続を定義する必要があります。これは、手動で定義された伝送キューを使用するか、クラスター化を使用することによって行えます。

各 MFT エージェントは、単一のキュー・マネージャーと通信します。複数のエージェントが同じキュー・マネージャーと通信する場合は、エージェント・キュー・マネージャーに次の複数のキューをエージェントごとに定義します。

- 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

SYSTEM.FTE.COMMAND.*などのプロファイルを使用して汎用セキュリティー・プロファイルを定義することも、エージェントごとに特定のプロファイルを定義することもできます。

関連概念

65 ページの『[ご使用になる前に](#)』

Managed File Transfer (MFT) 構成は、UNIX System Services (USS) のファイルと PDSE データ・セットを使用します。

z/OS ご使用になる前に

Managed File Transfer (MFT) 構成は、UNIX System Services (USS) のファイルと PDSE データ・セットを使用します。

構成と操作のほとんどは PDSE の JCL を使って行われるため、USS 環境での作業に精通しておく必要があります。

ISPF から OMVS にアクセスすることも、ワークステーションでコマンドを使って Telnet タイプのセッションを使用することもできます (例えば Telnet Putty や SSH)。

ISPF から OMVS を使用する場合、標準 ISPF エディターや、ブラウズ・コマンド **oedit** および **obrowse** を使用できます。

以下の USS コマンドに精通しておく必要があります。

コマンド	関数
ls -ltr path	パス内のファイルに関する情報をリストします。
ls -ltrd directory	指定されたディレクトリー内のファイルではなく、ディレクトリーに関する情報をリストします。
find path -name xxx	パスのディレクトリーで、xxxx という名前のファイルを検索します。xxx には大/小文字の区別があり、*zzz のように指定できます。
chmod xxx path	ファイル・アクセス許可を変更します。
df -k path	ファイル・システムに残っているフリー・スペース量を報告します。-k はフリー・スペースを KB 単位で報告します。
du -kt path	パスの下のディレクトリーのサイズを報告します。サイズは KB 単位で報告されます。
oedit filename	ファイルを OMVS で編集します。
obrowse filename	ファイル名を参照します

以下の表の項目を確認し、お客様の会社に適した項目を表に記入してください。これらの値は、メンバー BFGCUSTM を編集する際に必要になります。

名前	データの例	コメント
BFG_JAVA_HOME	/java/java71_bit64_GA/J7.1_64/	
BFG_GROUP_NAME	MQM	
LIBRARY	SCEN.FTE.JCL	MFT PDSE の名前。各エージェントまたはロガー・タスクのコピーが必要です。
TMPDIR	/tmp	一時ファイルの読み取り/書き込みアクセスが可能な USS パス。
SERVICE_TYPE	AGENT または LOGGER	
名前	AGENT1	
BFG_PROD	/var/ibm/wmqmft	
BFG_DATA		必要に応じて記入してください
BFG_JVM_PROPERTIES		必要に応じて記入してください
QMGR	MQPV	
MQ_PATH	/mqm/V8R0M0	
MQ_LANG	E	
Db2_HLQ	SYS2.Db2.V10	
FTE_CONFIG		マイグレーションで使用します
CREDENTIAL_PATH		マイグレーションで使用します
DB_PROPS_PATH		マイグレーションで使用します

表 4. メンバー BFGCUSTM に必要なパラメーター (続き)		
名前	データの例	コメント
BFG_WTO	YES	syslog の MFT メッセージを取得します。
ADMIN_JOB1		ジョブ・カード。ジョブはすべて同じ JCL カードで生成されます。
OUTPUT_CLASS	*	
JOBCARD1		これは、長時間実行されているタスク、エージェント、およびロガー用のジョブ・カードです。
PATH	bin:/usr/bin:/usr/sbin	
armELEMTYPE	ARM が使用されている場合、ARM ポリシーで指定されている ARM ELEMTYPE を使用します。例えば、エージェントの場合は armELEMTYPE=SYSBFGAG、ロガーの場合は armELEMTYPE=SYSBFGLG のようになります。ARM が使用されていない場合、このパラメーターを空白に設定します (例、armELEMTYPE=)。	
armELEMENT	ARM が使用されている場合、このエージェントまたはロガーの ARM ポリシーで指定されている ARM ELEMENT 値を使用します。ARM が使用されていない場合、このパラメーターを空白に設定します (例、armELEMENT=)。	
coordinationQMgr	MQPV	必須構成

さらに、以下の変数を確認し、必要に応じて値を指定する必要があります。

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

これらのプロパティーは、AGENT または LOGGER に共通しています。

注: クライアント接続にはホスト、ポート、およびチャンネルが必要ですが、ローカル・マシン上のバインディング接続では空白のままにしておく必要があります。

関連概念

[68 ページの『確認する項目』](#)

十分なディスク・スペースがあり、データを保管するためのディレクトリーがあり、必要ファイルが存在することを確認します。

[72 ページの『メンバー BFGCUSTM の編集』](#)

ジョブを実行する前に、メンバー BFGCUSTM を編集し、お客様の会社で使用するパラメーターの値を入力する必要があります。

z/OS 確認する項目

十分なディスク・スペースがあり、データを保管するためのディレクトリーがあり、必要ファイルが存在することを確認します。

十分なディスク・スペースの確認

構成固有のファイルを保管しようとしているファイル・システムに十分な使用可能ディスク・スペースが存在することを確認します。

エージェント・トレースが有効な場合、デフォルトで 100 MB のディスク・スペースを使用する場合があります。

構成ファイル自体は小さく、サイズは数 KB ほどです。

2つのエージェントと1つのローガーを使用する計画の場合、少なくとも 300 MB が必要です。コマンド **df -k path** を使用できます。path は、インストール済み環境固有のファイルの場所です。これにより、使用可能スペースと合計スペースが KB で表示されます。

300 MB は 307,200 KB なので、少なくとも 310,000 KB を使用可能にする必要があります。

Managed File Transfer データを保管するためのディレクトリーの作成と確認

Managed File Transfer (MFT) データを保管するためのディレクトリーが必要です。

ファイル・システム **df -k /var** に十分なスペースがあることを確認してください。このファイル・システムには少なくとも 310,000 KB の使用可能スペースが必要です。

このファイル・システムが作成されていない場合、**mkdir** コマンド (例えば **mkdir /var/mft**) を使用します。

コマンド **ls -ltrd /var/mft** を使用すると、このディレクトリーに対してユーザーが持つ権限が表示されます。

所有者またはグループが正しくない場合、コマンド **chown owner:group /var/mft** を使用します。

グループの許可が正しくない場合、以下のコマンドを使用して、所有者とグループに読み取り、書き込み、および実行の権限を付与します。コマンド **chmod 775 /var/mft** は、すべてのユーザーに読み取りと実行の権限を付与します。

ファイルが存在することとそのファイルに対するアクセス権限があることの確認

カスタマイズの際に使用するファイルに対して **ls -ltr** コマンドを使用します。以下に例を示します。

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

この結果は次のようになります。

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

drwxr-xr-x は、以下のものを意味します。

d

これはディレクトリーです。

rwX

所有者 SYSTASK に、ディレクトリーに対する読み取り、書き込み、および実行のアクセス権限があります。

r-x

グループ *TSOUSER* のユーザーは、ディレクトリー内のファイルを読み取りおよび実行できます。

r-x

汎用アクセスです。つまり、どのユーザーもディレクトリー内のファイルを読み取りまたは実行できます。

以下で指定されるファイルを確認してください。

Path	構成を行うユーザーに必要なアクセス権限
BFG_JAVA_HOME	読み取りと実行
/tmp	読み取りと書き込み
BFG_PROD	読み取り
BFG_DATA	書き込み
MQ_PATH	読み取り

関連概念

65 ページの『ご使用になる前に』

Managed File Transfer (MFT) 構成は、UNIX System Services (USS) のファイルと PDSE データ・セットを使用します。

69 ページの『MFT の一般的な構成 (z/OS)』

異なる Managed File Transfer 構成の概要

z/OS MFT の一般的な構成 (z/OS)

異なる Managed File Transfer 構成の概要

Managed File Transfer は、キュー・マネージャーに接続したエージェントを使用してデータを転送します。

MFT は、以下のような複数のキュー・マネージャーを使用できます。

- データを転送する 1 つ以上のキュー・マネージャー。
- 要求を実行するコマンド・キュー・マネージャー。例えば、転送を開始する要求がこのキュー・マネージャーに送信されると、関連したコマンドが MFT エージェントにルーティングされます。
- 作業を管理する調整キュー・マネージャー。

Managed File Transfer (MFT) には、以下の 3 つの共通の構成があります。

1. ローカル接続を使用して 1 つのキュー・マネージャーで 1 つ以上のエージェントを使用する構成。これは、データ・セットの内容を IBM MQ キューに入れるために使用できます。
2. クライアント・バインディングを使用して 1 つのキュー・マネージャーで分散マシン上の MFT クライアントを使用する構成。
3. チャンネルで 2 つのキュー・マネージャーを接続して、各マシンで 1 つ以上のエージェントを使用する構成。それらのエージェントでは、クライアント・バインディングもローカル・バインディングも可能です。

以下の点に注意してください。

1. MFT は、Java で記述されており、MFT を構成したり操作したりするためのシェル・スクリプトと JCL が組み込まれています。
2. Db2 の状況とアクティビティをログに記録して、Db2 表に保管できます。
3. MFT を構成する担当者は、UNIX システム・サービス (USS) に精通している必要があります。以下に例を示します。
 - /u/userID/myfile.txt2 などの名前のファイルを収容するディレクトリー構造

- 以下のような USS コマンド

cd (ディレクトリーの変更)

ls (リスト)

chmod (ファイル・アクセス許可の変更)

chown (ファイルの所有権の変更や、ファイルやディレクトリーにアクセスできるグループの変更)

4. MFT を構成して実行するには、USS で以下の製品が必要になります。

- Java (/java/java71_bit64_GA/J7.1_64/ など)
- IBM MQ V800 (/mqm/V8R0M03 など)
- 状況や履歴のために Db2 を使用する場合は Db2 JDBC ライブラリー (/db2/db2v10/jdbc/lib など)

調整キュー・マネージャーが必要です。ただし、同じキュー・マネージャーを使って、エージェントの実行、コマンドの処理、および調整を実行できます。複数のキュー・マネージャーを使用している場合、コーディネーターとして機能するものを1つ選択する必要があります。

IBM MQ 接続の確認

既存の MFT コーディネーター・キュー・マネージャーがある場合、構成を行うキュー・マネージャーと調整キュー・マネージャーおよびコマンド・キュー・マネージャーの間の接続が必要です。

z/OS での MQMFTCredentials.xml の構成

Managed File Transfer (MFT) がセキュリティーを有効にして構成されている場合、接続認証では、ユーザー ID とパスワードの資格情報を提供するために、すべての MFT エージェント、およびキュー・マネージャーに接続するコマンドが必要になります。

同様に、MFT ロガーは、データベースへの接続時にユーザー ID とパスワードを指定する必要がある場合があります。

この資格情報は、MFT 資格情報ファイルに保管できます。資格情報ファイルはオプションですが、環境をカスタマイズする前に必要な1つ以上のファイルを定義する方が簡単です。

これに加えて、資格情報ファイルがある場合は、受け取る警告メッセージの数が少なくなります。警告メッセージは、MFT でキュー・マネージャーのセキュリティーがオフであると見なされたために認証の詳細が指定されないということを通知します。

資格情報ファイルの例は、MQ_INSTALLATION_PATH/mqft/samples/credentials ディレクトリーにあります。

MQMFTCredentials.xml ファイルの例を以下に示します。

```
<?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>
```

ユーザー ID ADMIN を持つジョブは、キュー・マネージャー MQPH に接続する必要があるときに、ユーザー ID JOHNDOEH を渡し、パスワード cXXXX を使用します。

ジョブが他のユーザー ID で実行され、MQPH に接続する場合、そのジョブはユーザー ID NONEH とパスワード yXXXX を渡します。

MQMFTCredentials.xml ファイルのデフォルトの場所は、z/OS Unix System Services (USS) 上のユーザーのホーム・ディレクトリーです。USS 上の別の場所、または区分データ・セット内のメンバーのいずれかにファイルを保管することもできます。

資格情報ファイルが別の場所に保管されている場合は、以下のプロパティーを使用して、コマンドが検索する場所を指定できます。

表 6.: 各種コマンドの MQMFTCredentials.xml ファイルの場所を定義するプロパティー。

コマンドのタイプ	プロパティー・ファイル	プロパティー名
調整キュー・マネージャーに接続するコマンド	coordination.properties	coordinationQMGrAuthenticationCredentialsFile
コマンド・キュー・マネージャーに接続するコマンド	connection.properties	connectionQMGrAuthenticationCredentialsFile
エージェント・プロセスに接続するコマンド	agent.properties	agentQMGrAuthenticationCredentialsFile
ロガー・プロセスに接続するコマンド	logger.properties	loggerQMGrAuthenticationCredentialsFile

表 7.: エージェントおよびロガー・プロセスの MQMFTCredentials.xml ファイルの場所を定義するプロパティー。

コマンドのタイプ	プロパティー・ファイル	プロパティー名
MFT エージェント	agent.properties	agentQMGrAuthenticationCredentialsFile
MFT loggers	logger.properties	loggerQMGrAuthenticationCredentialsFile

どのコマンドおよびプロセスがどのキュー・マネージャーに接続するかについては、[どの MFT コマンドおよびプロセスがどのキュー・マネージャーに接続するかを参照してください。](#)

区分データ・セット内に資格情報ファイルを作成するには、以下のステップを実行します。

- VB の形式と論理レコード長 (Lrecl) 200 で PDSE を作成します。
- データ・セット内にメンバーを作成し、データ・セットとメンバーのメモを取り、以下のコードをメンバーに追加します。

```
<?xml version="1.0" encoding="IBM-1047"?>
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MQMFTCredentials"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://wmqfte.ibm.com/MFTCredentials MQMFTCredentials.xsd">
  <!--credentials information goes here-->
</tns:mqmftCredentials>
```

資格情報ファイルは、RACF®などのセキュリティー製品を使用して保護できますが、Managed File Transfer コマンドを実行するユーザー ID、およびエージェントとロガーのプロセスを管理するユーザー ID には、このファイルに対する読み取り権限が必要です。

このファイルの情報は、メンバー BFGCROBS の JCL を使って覆い隠すことができます。これはファイルを取り、IBM MQ ユーザー ID とパスワードを暗号化します。例えば、メンバー BFGCROBS は次の行を取ります。

```
<tns:qmgr name="MQPI" user="JOHNDOE2" mqUserId="JOHNDOE1" mqPassword="yXXXX" />
```

そして、次のものを作成します。

```
<tns:qmgr mqPasswordCipher="e977c61e9b9c363c" mqUserIdCipher="c394c5887867157c"
name="MQPI" user="JOHNDOE2"/>
```

このユーザー ID から IBM MQ ユーザー ID へのマッピングを保持する場合、ファイルにコメントを追加できます。例えば

```
<!-- name="MQPI" user="ADMIN" mqUserId="JOHNDOE1 -->
```

これらのコメントは、覆い隠すプロセスでは変更されません。

注: 内容は覆い隠されますが、高い強度で暗号化されるわけではありません。そのファイルにアクセスできるユーザー ID を制限する必要があります。

関連タスク

[Multiplatforms での MQMFTCredentials.xml の構成](#)

Managed File Transfer (MFT) がセキュリティーを有効にして構成されている場合、接続認証では、キュー・マネージャーに接続するすべての MFT コマンドでユーザー ID とパスワードの資格情報を提供する必要があります。同様に、MFT ロガーは、データベースへの接続時にユーザー ID とパスワードを指定する必要があります。この資格情報は、MFT 資格情報ファイルに保管できます。

JCL ライブラリーを作成するための SBFGCMDS のコピー

エージェントおよびロガーごとに JCL ライブラリーを作成する必要があります。JCL には、エージェントまたはロガーの作成および実行に使用する構成およびジョブが含まれています。

BFGCOPY メンバーを編集して実行することによって、エージェントおよびロガーごとに IBM 提供の SBFGCMDS ライブラリーのコピーを作成します。

このライブラリーは、エージェントまたはロガーの構成を定義する場合に使用され、カスタマイズ後には、必要な Managed File Transfer 構成の作成、およびエージェントまたはロガーの作成に使用できるジョブが置かれます。

メンバー BFGCUSTM をこのプロセスの一部として作成します。

注: USS コマンドに精通している場合、他のプラットフォームで使用するのと同じコマンドを使用して z/OS を構成できます。

関連概念

69 ページの『[MFT の一般的な構成 \(z/OS\)](#)』

異なる Managed File Transfer 構成の概要

72 ページの『[メンバー BFGCUSTM の編集](#)』

ジョブを実行する前に、メンバー BFGCUSTM を編集し、お客様の会社で使用するパラメーターの値を入力する必要があります。

メンバー BFGCUSTM の編集

ジョブを実行する前に、メンバー BFGCUSTM を編集し、お客様の会社で使用するパラメーターの値を入力する必要があります。

特定の値を必要とするパラメーターのリストについては、[メンバー BFGCUSTM に必要なパラメーターを参照](#)してください。

さらに、以下の変数を確認し、必要に応じて値を指定する必要があります。

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

これらのプロパティーは、AGENT または LOGGER に共通しています。

注: クライアント接続にはホスト、ポート、およびチャンネルが必要ですが、ローカル・マシン上のバインディング接続ではブランクのままにしておく必要があります。

これが Managed File Transfer 環境で最初のキュー・マネージャーであり、かつ、この同じキュー・マネージャーを調整、コマンドおよびエージェント実行に使用する場合は、この値をローカル・キュー・マネージャーの名前に設定します。

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

MQPV は、ローカル・キュー・マネージャーの名前です。

ジョブを実行依頼します。これにより、PDSE が更新され、指定されたパスの下にディレクトリー構造が作成されます。

このジョブは排他的に使用する必要があるため、ジョブの実行中は PSDE の使用を停止する必要があります。

ヒント: ジョブ BFGCUSTM を実行依頼するときは常に、そのジョブによってすべての JCL ファイルが置換されます。変更するメンバーごとに名前変更する必要があります。

関連概念

65 ページの『ご使用になる前に』

Managed File Transfer (MFT) 構成は、UNIX System Services (USS) のファイルと PDSE データ・セットを使用します。

75 ページの『エージェントの作成』

エージェント固有の PDSE (例えば `user.MFT.AGENT1`) を作成するには、PDSE をコピーする必要があります。以前のエージェント構成またはロガー構成があれば、その構成から PDSE をコピーします。初めて構成を作成する場合は、MFT に用意されている PDSE をコピーします。

z/OS 調整キュー・マネージャーの定義

Managed File Transfer では、調整キュー・マネージャーとして機能するキュー・マネージャーを作成する必要があります。

選択した構成に応じて、このキュー・マネージャーは、ローカル MVS システムか、別のマシンに配置されます。前者の場合、その接続はバインディング接続になり、後者の場合は、クライアント接続になります。

構成ステップを正常に実行した後、PDSE に構成済みメンバーが存在するようになります。

メンバー BFGCFR は調整キュー・マネージャーを定義します。このジョブは以下のことを行います。

1. Managed File Transfer (MFT) ディレクトリーにディレクトリー構造を作成し、構成ファイルを作成します。
2. CSQUTIL を実行して IBM MQ リソースを定義します。

調整キュー・マネージャーがリモート・マシン上にある場合、このジョブ・ステップは失敗します。

メンバー BCFCFCR は USS にファイルを作成し、MQ 定義を作成します。このジョブは、以下のことを行います。

1. MFT トピックを作成します
2. MFT キューを作成します
3. `NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)` が `NAMES(SYSTEM.BROKER.DEFAULT.STREAM, SYSTEM.BROKER.ADMIN.STREAM, SYSTEM.FTE)` となるように変更します。
4. `ALTER QMGR PSMODE(ENABLED)` を実行します。

変更を行う前に `DISPLAY NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)` コマンドが発行されます。NAMLIST がデフォルトではない場合、名前リストを変更して `SYSTEM.FTE` を名前リストに追加する必要があります。

メンバー BCFCFCR を独自の接頭部を使って名前変更します (例えば CCPCFCR)。このファイルを再びカスタマイズするとそれが置換されるためです。

資格情報ファイルの名前を挿入して、名前変更されたこのメンバーを編集します。以下に例を示します。

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

ジョブを保存し実行依頼します。ジョブを再実行依頼する必要がある場合、`-f`オプションを追加する必要があります。

このジョブを実行すると、それによって作成される IBM MQ リソースがリストされます。これらのリソースを保護する必要があります。

```
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)
```

関連タスク

74 ページの『[コマンド・キュー・マネージャーの定義](#)』

同じキュー・マネージャーを調整キュー・マネージャーとコマンド・キュー・マネージャーとして使用することも、新しくコマンド・キュー・マネージャーを作成することもできます。

z/OS

コマンド・キュー・マネージャーの定義

同じキュー・マネージャーを調整キュー・マネージャーとコマンド・キュー・マネージャーとして使用することも、新しくコマンド・キュー・マネージャーを作成することもできます。

このタスクについて

コマンド・キュー・マネージャーは必須ですが、同じキュー・マネージャーを調整キュー・マネージャーとコマンド・キュー・マネージャーに使用することができます。そうしない場合は、新しいコマンド・キュー・マネージャーを作成する必要があります。これは調整キュー・マネージャーと同じマシンに配置することができますが、そうする必要があるわけではありません。

手順

1. メンバー BFGCMCR を独自の接頭部 (例えば CCPCMCR) を使って名前変更します。
BFGCMCR は必ず名前変更してください。このファイルを再カスタマイズすると、置き換えられるためです。
2. 資格情報ファイルの名前を挿入して、名前変更されたメンバーを編集します。
以下に例を示します。

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

3. ジョブを保存し実行依頼します。
ジョブを再実行依頼する必要がある場合、`-f`オプションを追加する必要があります。
このキュー・マネージャーは、**ftePingAgent** などのコマンドで使用されます。
4. このメンバーを確認し、実行依頼し、その出力を確認します。

次のタスク

エージェントの作成方法について詳しくは、75 ページの『[エージェントの作成](#)』を参照してください。

関連概念

73 ページの『調整キュー・マネージャーの定義』

Managed File Transfer では、調整キュー・マネージャーとして機能するキュー・マネージャーを作成する必要があります。

関連タスク

43 ページの『マルチプラットフォームでの MQMFTCredentials.xml の構成』

Managed File Transfer (MFT) がセキュリティーを有効にして構成されている場合、接続認証では、キュー・マネージャーに接続するすべての MFT コマンドでユーザー ID とパスワードの資格情報を提供する必要があります。同様に、MFT ロガーは、データベースへの接続時にユーザー ID とパスワードを指定する必要があります。この資格情報は、MFT 資格情報ファイルに保管できます。

関連資料

810 ページの『MFT 資格情報ファイル・フォーマット』

MQMFTCredentials.xml ファイルには、ユーザー ID とパスワードに関する重要な情報が含まれています。MQMFTCredentials.xml ファイル内のエレメントは MQMFTCredentials.xsd スキーマに準拠する必要があります。資格情報ファイルのセキュリティーはユーザーの責任です。

z/OS エージェントの作成

エージェント固有の PDSE (例えば *user.MFT.AGENT1*) を作成するには、PDSE をコピーする必要があります。以前のエージェント構成またはロガー構成があれば、その構成から PDSE をコピーします。初めて構成を作成する場合は、MFT に用意されている PDSE をコピーします。

メンバー BFGCUSTM を確認し、異なる資格情報ファイルを使用する必要がある場合は、作成します。

内容のほとんどは、72 ページの『メンバー BFGCUSTM の編集』で詳述されているカスタマイズと同じです。

以下の変更を行う必要があります。

- //SYSEXEC DD DSN=SCEN.FTE.JCL.AGENT1
- LIBRARY がエージェント PDSE と一致するようにする
- SERVICE_TYPE=AGENT
- NAME が (PDSE と一致する) エージェント JOBCARD の名前になるようにする
- BFG_JVM_PROPERTIES="-Xmx1024M" を変更する

このジョブを実行依頼します。その際、このジョブはデータ・セットへの排他的アクセスが必要であることを覚えておいてください。

エージェントのジョブの名前の形式はすべて *BFGAG** になります。

メンバー *BFGAGCR* の名前を変更します。このジョブは Managed File Transfer ディレクトリーのファイルを更新し、CSQUTIL を使用してローカル・キュー・マネージャーのエージェント固有キューを作成します。資格情報ファイルの名前 (-credentialsFile //'SCEN.FTE.JCL.VB(CREDOLD) など) を指定します。名前を指定しないと、エージェントを開始するためのジョブが資格情報ファイルを使用しなくなります。

出力を確認し、プロセスが正常に実行されたことを確認します。

ヒント: *agent.properties* ファイルのパス名を、ジョブの出力からエージェントの PDSE のメンバーにコピーします。

例えば、*/u/userid/fte/wmqmft/mqft/config/MQPA/agents/AGENT1/agent.properties* をメンバー AGENT にコピーします。

こうしておくと、プロパティ・ファイルを表示して */u/userid/fte/wmqmft/mqft/logs/MQPA/agents/AGENT1/logs* という行を追加する必要がある場合に便利です。

これはトレース・ファイルの保管場所です。

関連概念

73 ページの『調整キュー・マネージャーの定義』

Managed File Transfer では、調整キュー・マネージャーとして機能するキュー・マネージャーを作成する必要があります。

[76 ページの『エージェントの使用』](#)

エージェントが正しく動作していることを確認するためのさまざまなコマンドの使用法。

関連タスク

[74 ページの『コマンド・キュー・マネージャーの定義』](#)

同じキュー・マネージャーを調整キュー・マネージャーとコマンド・キュー・マネージャーとして使用することも、新しくコマンド・キュー・マネージャーを作成することもできます。

エージェントの使用

エージェントが正しく動作していることを確認するためのさまざまなコマンドの使用法。

エージェントの開始

メンバー BFGAGST の名前を変更し、メンバーを確認し、ジョブを実行依頼します。

これが動作している場合、「BFGAG0059I: エージェントは正常に開始されました」というメッセージが表示されます。

アクティブ・エージェントの表示

メンバー BFGAGLI の名前を変更し、メンバーを確認し、調整キュー・マネージャーを使用するジョブを実行依頼します。

接続の問題があれば、それを解決する必要があります。

動作確認のためのエージェントに対する ping

メンバー BFGAGPI の名前を変更し、メンバーを確認し、コマンド・キュー・マネージャーを使用するジョブを実行依頼します。

接続の問題があれば、それを解決する必要があります。

テスト転送の実行

詳しくは、[83 ページの『検証転送の実行』](#)を参照してください。

エージェントの停止

メンバー BFGAGSP の名前を変更し、メンバーを確認し、ジョブを実行依頼します。

メンバー BFGAGST を使用してエージェントを再始動します。

関連概念

[75 ページの『エージェントの作成』](#)

エージェント固有の PDSE (例えば `user.MFT.AGENT1`) を作成するには、PDSE をコピーする必要があります。以前のエージェント構成またはロガー構成があれば、その構成から PDSE をコピーします。初めて構成を作成する場合は、MFT に用意されている PDSE をコピーします。

MFT の既存のエージェントまたはロガーのコマンド・データ・セットの更新 (z/OS)

Managed File Transfer コマンド・テンプレート・データ・セットから作成された Managed File Transfer コマンド PDSE ライブラリー・データ・セットを更新できます。

手順

1. BFGCUSTM JCL スクリプト・メンバーを編集して、BFGSTDIN DD ステートメント内の変数とプロパティを更新します。

以前に定義したプロパティを除去する場合には、エントリーを除去するのではなく、その値を空白に設定します。BFGCUSTM JCL スクリプトの実行時に、指定したプロパティは、エージェントおよびロガーの実際の UNIX System Services プロパティ・ファイルへの更新として適用されます。あるプロパティを空白値に設定することは、そのプロパティを削除するという指示になります。

2. ジョブ BFGCUSTM を実行依頼します。このジョブは、エージェントまたはロガーに適切な JCL コマンドのセットを再び生成します。コマンドの完全なリストについては、[82 ページの『z/OS エージェントおよびロガー・コマンドの JCL スクリプト』](#)を参照してください。出力ジョブ・ログを調べて、JCL スクリプトが正常に実行されたことを確認してください。何らかの障害がある場合には、それらを修正して BFGCUSTM ジョブを再び実行依頼してください。

タスクの結果

生成された JCL スクリプトに変更を加えて、独自のロジックを追加することができます。ただし、BFGCUSTM を再び実行するときには、カスタム・ロジックを上書きすることになる場合があるので、注意してください。

関連概念

59 ページの『z/OS での MFT 構成オプション』

z/OS での Managed File Transfer 構成オプションは、分散プラットフォームの場合のオプションと同じです。

関連タスク

63 ページの『MFT エージェントまたはロガーのコマンド・データ・セットの作成』

特定の調整のため、特定の Managed File Transfer Agent または Managed File Transfer Logger 用の Managed File Transfer コマンド・テンプレート・データ・セットから、コマンドの PDSE データ・セットを作成することができます。

z/OS

z/OS JCL 変数

置換値、JCL 変数、および構成プロパティを BFGCUSTM スクリプト内で使用できます。

以下の表に、MFT コマンド PDSE ライブラリー・データ・セット内の BFGCUSTM JCL スクリプトの置換値をリストします。BFGCUSTM ジョブを実行依頼する前に、これらの置換値を適切な値に置き換える必要があります。

置換変数	値
++library++	MFT コマンド PDSE ライブラリーを含むデータ・セットの名前。
++bfg_java_home++	Java インストールの場所。
++bfg_prod++	MFT 製品のインストール先の UNIX System Services ルート・ディレクトリーの場所。

以下の表では、MFT コマンド PDSE ライブラリー・データ・セット内の BFGCUSTM JCL スクリプトの BFGSTDIN DD ステートメントのための環境変数 ([Variables] セクション内) を説明します。BFGCUSTM ジョブを実行依頼する前に、置換値 (2 つの正符号 ++ で囲まれた値) で指定されたすべての変数を適切な値に置き換える必要があります。

表 9. 環境変数	
環境変数	値
LIBRARY	MFT コマンド PDSE ライブラリーを含むデータ・セットの名前。
TMPDIR	一時ファイル用の UNIX System Services ディレクトリー。
BFG_PROD	MFT 製品のインストール先の UNIX System Services ルート・ディレクトリーの場所。
BFG_DATA	Managed File Transfer for z/OS のデータ・ディレクトリーの場所。これは <i>DATA_DIR</i> のパスです。
BFG_JAVA_HOME	Java インストールの場所。
BFG_JVM_PROPERTIES	オプション。BFG_JVM_PROPERTIES 環境変数の値を設定します。これらのプロパティは、Java 仮想マシンに引き渡されます。

表 9. 環境変数 (続き)

環境変数	値
BFG_GROUP_NAME	<p>通常、mqm ファイル・グループは、MFT 構成のデータ・ファイルおよびコマンドに関連付けられます。したがって、mqm グループのメンバーであるすべてのユーザーが MFT 構成へのアクセスと変更を行うことができます。詳しくは、350 ページの『IBM MQ の MFT でのファイル・システムの権限』を参照してください。</p> <p>z/OS システムの場合、ファイル・グループは USS ファイル・システム・エンティティであり、mqm ファイル・グループは必ずしも定義されません。BFG_GROUP_NAME 環境変数を使用すると、MFT 構成データ・ファイル用に z/OS USS ファイル・システム・グループを関連付けることができます。例えば、USS シェルのプロンプトに次のように入力します。</p> <pre data-bbox="860 745 1471 819">export BFG_GROUP_NAME=FTEGB</pre> <p>このようにすると、現在の USS セッションでこれ以後に作成するすべての構成ファイルにグループ <i>FTEGB</i> を関連付ける操作が定義されます。</p> <p>BFG_GROUP_NAME は、ブランク値に設定することも、削除することもできます。</p> <p>注: MFT 構成が複数のユーザー ID によって使用される計画の場合、初めて BFGCUSTM を実行する際に、必要なすべてのユーザー ID がアクセスできるグループを BFG_GROUP_NAME に設定することが重要です。BFGCUSTM を再度実行する場合には、BFG_GROUP_NAME を変更してはなりません (変更する場合は、BFG_DATA によって参照されるディレクトリ内のすべてのファイルおよびディレクトリの USS グループ・ファイル・アクセス許可も、新しい BFG_GROUP_NAME 設定を反映するように変更する必要があります)。</p> <p>V9.0.0.1 BFG_GROUP_NAME 環境変数が非ブランクの値に設定されている z/OS システムで fteMigrateAgent コマンドを実行すると、このコマンドによって、ユーザーが BFG_GROUP_NAME 変数で指定されたグループのメンバーであるかどうかを確認されます。ユーザーが指定されたグループに属していない場合、コマンドはエラー・メッセージ BFGCL0502E: 「要求された操作を実行する権限がありません。」を報告することがあります。実行に失敗しました。このコマンドを正常に実行するためにユーザーが満たす必要がある基準について詳しくは、483 ページの『fteMigrateAgent: FTE 7.0 エージェントの MQ 7.5 以降へのマイグレーション』を参照してください。</p>

表 9. 環境変数 (続き)

環境変数	値
BFG_WTO	BFG_WTO に YES、ON、または TRUE を設定すると、z/OS ログインが使用可能になります。これにより、エージェントのイベント・ログに書き込まれるメッセージが z/OS オペレーター・ログ機能にも書き込まれるかどうかを制御できるため、JCL からエージェントを実行した場合に自動化製品へのアクセスがより簡単になります。宛先コードは「プログラマー情報」(11) で、記述子コードは「通知」(12) です。
SERVICE_TYPE	MFT コマンド・ライブラリーがエージェント用か、ロガー用かを指定します。有効な値は AGENT または LOGGER です。
名前	SERVICE_TYPE 値に対するエージェントまたはロガーの名前。
QMGR	SERVICE_TYPE 値に対するエージェントまたはロガーに関連付けられたローカル・キュー・マネージャーの名前。
OUTPUT_CLASS	SYSOUT データ・セットの出力クラス。デフォルトは * であり、この場合はジョブ・ステートメントからの MSGCLASS パラメーターと同じ出力クラスを要求します。
MQ_PATH	LIBPATH 環境変数を作成するために BFGPROF で使用されます。
MQ_HLQ	IBM MQ データ・セットの高位修飾子。
MQ_LANG	必要な言語。
DB2_HLQ	オプション。Db2 データ・セット用の高位修飾子。
JOBCARD1	JCL コマンド・ジョブのヘッダー行 1。
JOBCARD2	JCL コマンド・ジョブのヘッダー行 2。
JOBCARD3	JCL コマンド・ジョブのヘッダー行 3。
ADMIN_JOB1	admin ジョブのヘッダー行 1。
ADMIN_JOB2	admin ジョブのヘッダー行 2。
ADMIN_JOB3	admin ジョブのヘッダー行 3。
FTE_CONFIG	マイグレーションの対象となる既存の WMQFTE 構成。マイグレーションが必要ない場合には、ブランク値に設定します。
CREDENTIAL_PATH	マイグレーションの対象となる資格情報ファイルのパス (例: /u/user1/agent3)。Managed File Transfer のマイグレーション対象となる資格情報ファイルは、IBM WebSphere MQ File Transfer Edition 7.0.4.4 の構成情報および構成ファイルとは別のファイルに置かれている必要があります。マイグレーション・コマンド BFGAGMG および BFGGLMG JCL スクリプトでのみ必要です。マイグレーションが必要ない場合には、ブランク値に設定します。

表 9. 環境変数 (続き)	
環境変数	値
DB_PROPS_PATH	マイグレーションの対象となるデータベース・ローガー・プロパティ・ファイルを指定します。このオプションは、プロパティ・ファイルがデフォルトの名前およびパスである <code>config_directory/coordination_qmgr/databaselogger.properties</code> を使用しない場合のみ必須です。マイグレーションが必要ない場合には、空白値に設定します。

注: IBM MQ jar ファイルは MFT に付属しており、ディレクトリー `MQMFT product root/java/lib` にあり、常に使用され、構成することはできません。

以下の表では、MFT コマンド PDSE ライブラリー・データ・セット内の BFGCUSTM JCL スクリプト用の BFGSTDIN DD ステートメントのための必須 MFT 構成プロパティを記述します。BFGCUSTM ジョブを実行依頼する前に、置換値 (2 つの正符号 ++ で囲まれた値) で指定されたプロパティを、適切な非空白値に置き換える必要があります。これらのプロパティは、MFT 構成プロパティの指定変更を定義します。ご使用の環境に合わせてエージェントまたはローガーをカスタマイズするために、エージェントおよびローガーのプロパティを追加することができます。すべてのプロパティのリストについては、[90 ページの『構成プロパティ・ファイル』](#)を参照してください。

表 10. BFGSTDIN DD ステートメント用の必須構成プロパティ	
プロパティ	値
coordinationQMGr	エージェントまたはローガーが関連付けられる構成のための調整キュー・マネージャーの名前。
coordinationQMGrHost	オプション。調整キュー・マネージャーが実行されるシステムのホスト名。このプロパティの値を空白のままにした場合、バインディング・モードでの接続が想定されます。
coordinationQMGrPort	オプション。調整キュー・マネージャーが listen するポート番号。このパラメーターは、 <code>coordinationQMGrHost</code> プロパティにも非空白の値を指定している場合にもみ使用されます。
coordinationQMGrChannel	オプション。調整キュー・マネージャーへの接続に使用するチャンネル。このパラメーターは、 <code>coordinationQMGrHost</code> プロパティにも非空白の値を指定している場合にもみ使用されます。
connectionQMGr	エージェントまたはローガーが関連付けられる構成のためのコマンド・キュー・マネージャーの名前。
connectionQMGrHost	オプション。コマンド・キュー・マネージャーが実行されるシステムのホスト名。このプロパティの値を空白のままにした場合、バインディング・モードでの接続が想定されます。
connectionQMGrPort	オプション。コマンド・キュー・マネージャーが listen するポート番号。このパラメーターは、 <code>connectionQMGrHost</code> プロパティにも非空白の値を指定している場合にもみ使用されます。

表 10. BFGSTDIN DD ステートメント用の必須構成プロパティ (続き)

プロパティ	値
connectionQMGrChannel	オプション。コマンド・キュー・マネージャーへの接続に使用するチャンネル。このパラメーターは、connectionQMGrHost プロパティにも非ブランクの値を指定している場合にのみ使用されます。

z/OS z/OS エージェントおよびロガー・コマンドの JCL スクリプト

MFT コマンド PDSE ライブラリー・データ・セットにある使用可能な JCL コマンドのセット。

表 11. MFT コマンド PDSE ライブラリー・データ・セットにある使用可能な JCL コマンド

Member	fte コマンド行コマンドの説明
BFGCOPY	このライブラリーのコピーを作成するジョブ
BFGCUSTM	エージェントまたはロガー用にこのライブラリーをカスタマイズするジョブ
BFGCFCR	<u>fteSetupCoordination</u>
BFGCMCR	<u>fteSetupCommands</u>
BFGAGCR	<u>fteCreateAgent</u> 。SERVICE_TYPE 変数を AGENT に設定した場合にのみ作成されます。
BFGLGCRS	<u>fteCreateLogger</u> 。SERVICE_TYPE 変数を LOGGER に設定した場合にのみ作成されます。
BFGAGST	<u>fteStartAgent</u> 。SERVICE_TYPE 変数を AGENT に設定した場合にのみ作成されます。
BFGAGSTP	fteStartAgent プロシージャ。SERVICE_TYPE 変数を AGENT に設定した場合にのみ作成されます。
BFGAGPI	<u>ftePingAgent</u> 。SERVICE_TYPE 変数を AGENT に設定した場合にのみ作成されます。
BFGAGSP	<u>fteStopAgent</u> 。SERVICE_TYPE 変数を AGENT に設定した場合にのみ作成されます。
BFGLGST	<u>fteStartLogger</u> 。SERVICE_TYPE 変数を LOGGER に設定した場合にのみ作成されます。
BFGLGSTP	fteStartLogger プロシージャ。SERVICE_TYPE 変数を LOGGER に設定した場合にのみ作成されます。
BFGLGSP	<u>fteStopLogger</u> 。SERVICE_TYPE 変数を LOGGER に設定した場合にのみ作成されます。
BFGAGSH	<u>fteShowAgentDetails</u> 。SERVICE_TYPE 変数を AGENT に設定した場合にのみ作成されます。
BFGLGSH	<u>fteShowLoggerDetails</u> 。SERVICE_TYPE 変数を LOGGER に設定した場合にのみ作成されます。
BFGCFDF	<u>fteChangeDefaultConfigurationOptions</u>

表 11. MFT コマンド PDSE ライブラリー・データ・セットにある使用可能な JCL コマンド (続き)

Member	fte コマンド行コマンドの説明
BFGAGCL	fteCleanAgent 。SERVICE_TYPE 変数を AGENT に設定した場合にのみ作成されます。
BFGAGDE	fteDeleteAgent 。SERVICE_TYPE 変数を AGENT に設定した場合にのみ作成されます。
BFGLGDE	fteDeleteLogger 。SERVICE_TYPE 変数を LOGGER に設定した場合にのみ作成されます。
BFGPRSH	fteDisplayVersion
BFGAGLI	fteListAgents 。SERVICE_TYPE 変数を AGENT に設定した場合にのみ作成されます。
BFGMNL	fteListMonitors
BFGSTLI	fteListScheduledTransfers
BFGTMLI	fteListTemplates
BFGAGMG	fteMigrateAgent SERVICE_TYPE 変数を AGENT に設定した場合にのみ作成されます。
BFGLGMG	fteMigrateLogger SERVICE_TYPE 変数を LOGGER に設定した場合にのみ作成されます。
BFGCROBS	fteObfuscate サンプル
BFGRAS	fteRAS
BFGAGTC	fteSetAgentTraceLevel 。SERVICE_TYPE 変数を AGENT に設定した場合にのみ作成されます。
BFGLGTC	508 ページの『 fteSetLoggerTraceLevel 』 SERVICE_TYPE 変数を LOGGER に設定した場合にのみ作成されます。
BFGPRANS	fteAnt サンプル
BFGTRCAS	fteCancelTransfer サンプル
BFGMNCRS	fteCreateMonitor サンプル
BFGTMCRS	fteCreateTemplate サンプル
BFGTRCRS	fteCreateTransfer サンプル
BFGMNDES	fteDeleteMonitor サンプル
BFGSTDES	fteDeleteScheduledTransfer サンプル
BFGTMDES	fteDeleteTemplates サンプル

z/OS

注: MQSC スクリプトを作成するコマンドや削除スクリプトを参照するコマンドの場合、JCL はスクリプトを実行するよう求めてきますが、スクリプトはジョブによって既に実行されています。

z/OS **検証転送の実行**

製品が正常に動作していることを確認するための転送の実行方法。

メンバー BFGTRCRS の名前を変更し、編集します。

1. %BFGCMD CMD=fteCreateTransfer -h の前に /* を追加します。
2. メンバー内の他のコメントを除去します。
3. -sa および -da で現行エージェント名を指定します。
4. JCL を保存します。
5. JCL を実行依頼します。

この JCL は、コマンド・キュー・マネージャーに接続します。

z/OS ロギング・タスクの構成

ロギング・タスクは、調整キュー・マネージャーと同じイメージで実行する必要があります。Db2 にログを記録できます。

ロギング・タスクの作成

PDSE をコピーして、ロガー固有の PDSE を作成します。例えば、user.MFT.LOGGER などです。

異なる資格情報ファイルを使用する必要がある場合、作成します。詳しくは、70 ページの『z/OS での MQMFTCredentials.xml の構成』を参照してください。

メンバー `BFGCUSTM` を確認します。内容のほとんどは、前のカスタマイズと同じです
ただし、以下のことを行う必要があります。

- //SYSEXEC DD DSN=SCEN.FTE.JCL.... を変更する
- エージェント PDSE と一致するように LIBRARY を変更する
- QMGR を調整キュー・マネージャーの名前に変更する
- SERVICE_TYPE=LOGGER を作成する
- NAME を (PDSE と一致する) ロガーの名前になるよう変更する
- JOBCARD を確認し、このジョブ名をエージェントのジョブ名とは異なるものに変更する
- BFG_JVM_PROPERTIES="-Xmx1024M" を確認する

Db2 ロガーを使用している場合、ファイルを作成すると便利です。そうすることにより、Db2 の問題の特定に役立つ Db2 トレースをキャプチャーすることができます。

ファイルの名前は JVM プロパティーで指定されています。JDBC トレース・プロパティー・ファイルの内容は次のようになります。

```
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
```

2つの JVM プロパティーを設定します。

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

/u/.../sql.properties は Db2 トレース・プロパティー・ファイルの名前で、DBCA は Db2 サブシステムの名前です。

このジョブを実行依頼します。その際、このジョブはデータ・セットへの排他的アクセスが必要であることに注意してください。エージェントのジョブの名前はすべて `BFGGLG*` のようになります。

ファイルへのロギング

Db2 へのロギングについて詳しくは、[85 ページの『ロギング・タスクの作成 \(Db2 にロギングする場合\)』](#)を参照してください。

メンバー BFGLGCRS の名前を変更します。このジョブは Managed File Transfer (MFT) ディレクトリーのファイルを更新し、CSQUTIL を使用してローカル・キュー・マネージャーのエージェント固有キューを作成します。

元のファイルには、コマンドの構文をリストするコマンド %BFGCMD CMD=fteCreateLogger -h が含まれています。

ロガー・タスクを作成するには、%BFGCMD CMD=fteCreateLogger -h の前に /* を置いてこのステートメントをコメント化します。その際、列 1 は必ずブランクにします。

2 番目のコマンドからコメントを除去し、ステートメントを構成します。以下に例を示します。

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

出力を確認して正常に処理されていることを確認します。

ヒント: logger.properties ファイルのパス名を、ジョブの出力からエージェントの PDSE のメンバーにコピーします。

例えば、メンバー APATH にコピーします。

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

これは、プロパティー・ファイルを表示する必要がある場合に便利です。

ディレクトリーをこのファイルに追加します。

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

ファイルにログを記録する場合、ログ・ファイルはこのディレクトリーに保管されます (例: LOGGER0-20140522123654897.log)。

トレース・ファイルは、例えば次のようなログ・サブディレクトリーに保管されます。

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

これでロギング・タスクを開始できるようになります。

ロギング・タスクの作成 (Db2 にロギングする場合)

メンバー BFGLGCRS の名前を変更します。

このジョブは MFT ディレクトリーのファイルを更新し、CSQUTIL を使用してローカル・キュー・マネージャーのエージェント固有キューを作成します。

以下を把握しておく必要があります。

表 12. Db2 変数

Db2 名	例
-dbName databaseName	これは、Db2 サブシステムのメッセージ DSNL004I のロケーション値から取得できます
-dbDriver filePath	例えば /db2/db2v10/jdbc/classes/db2jcc.jar
-dbLib filePath	例えば /db2/db2v10/jdbc/lib/libdb2jcct2zos_64.so

ファイルを編集します。元のファイルには、コマンドの構文をリストするコマンド %BFGCMD CMD=fteCreateLogger -h が含まれています。

2 番目のコマンドからコメントを除去し、ステートメントを構成します。例えば

```
%BFGCMD CMD=fteCreateLogger +
-p MQPH +
-loggerQMgr MQPH +
-loggerType DATABASE +
-dbType DB2 +
-databaseName DSNDBCP +
-dbDriver /db2/db2v10/jdbc/classes/db2jcc.jar +
-dbLib /db2/db2v10/jdbc/lib/ +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>' +
LOGGER
```

ロガー・タスクを作成するには、%BFGCMD CMD=fteCreateLogger -h の前に /* を置いてこのステートメントをコメント化します。その際、列 1 は必ずブランクにします。

ジョブを実行依頼し、出力を確認して正常に処理されていることを確認します。

ヒント: logger.properties ファイルのパス名を、ジョブの出力からエージェントの PDSE のメンバーにコピーします。

例えば、メンバー APATH にコピーします。

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

これは、プロパティ・ファイルを表示する必要がある場合に便利です。

トレース・ファイルは、例えば次のようなログ・サブディレクトリーに保管されます。

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

Db2 表の作成

Db2 表を作成する必要があります。定義は USS ファイル mqft/sql/ftelog_tables_zos.sql で指定されています。

PDSE にメンバー Db2 を作成します。このメンバーを編集し、コマンド行で COPY コマンドを使用します。USS 定義ファイルからコピーします。

サイト固有の要件は大きく異なる場合があるため、このファイルでは表の基本構造と、表が配置される表スペースのみを指定しています。

可能な限り最大の表の行を保持できるページ・サイズを持つバッファ・プールを使用して表が作成されるように、表スペースが SQL スクリプトによって指定されます。LOB の場所などの属性は指定されないことに注意してください。

データベース管理者は、このファイルのコピーを変更して、パフォーマンスに関連したこれらの属性を定義することもできます。

このファイルでは、デフォルト・スキーマ名として FTELOG、デフォルト表スペース名として FTELOGTS、データベース名として FTELOGDB を使用すると想定します。必要に応じて、ファイルの先頭にあるコメントで説明されているプロセスに従ってこれらの名前を変更し、既存のデータベースおよびローカル命名規則に一致させることができます。

重要: ファイルにはコメントがあり、**DSNTINAD** などのバッチ・プログラムはコメントを受け入れないため、**SPUFI** などのオンライン機能を使用してコマンドを実行してください。

ロガー・タスクの開始

メンバー BFGLGST の名前を変更し、確認して、実行依頼します。「BFGDB0023I: ロガーは開始アクティビティを完了し、現在、実行中です」というメッセージが表示されるはずで

ロガーの操作

ロガーの状況を表示するには、メンバー BFGLGSH の名前を変更し、確認して、実行依頼します。

ロガーを停止するには、メンバー BFGLGSP の名前を変更し、確認して、実行依頼します。

MFT の環境変数 (z/OS)

USS 環境から直接、または独自の JCL スクリプトからコマンドを実行している場合には、カスタマイズや構成の後に、Managed File Transfer で提供される構成スクリプトや管理スクリプトを実行するには、その前にいくつかの環境変数を設定する必要があります。これらの変数は、スクリプトの呼び出し元のユーザーごとおよび環境ごとに設定しなければなりません。

他の製品との競合を避けるために、ホーム・ディレクトリーに `.wmqfterc` スクリプトを作成することもできます。このようにすると、`.wmqfterc` スクリプトが Managed File Transfer の各スクリプトによって呼び出され、このスクリプトを使用して Managed File Transfer のカスタム環境設定を指定することができます。

さらに、オプションの環境変数 `BFG_WTO` もあり、この変数は JCL からエージェントを実行する際にオペレーター・ログにメッセージを送信するために設定できます。

環境変数	値
BFG_JAVA_HOME	Java インストールの場所。サポートされている Java のレベルの詳細については、 IBM MQ System Requirements を参照してください。
BFG_DATA	Managed File Transfer for z/OS のデータ・ディレクトリーの場所。これは <code>DATA_DIR</code> へのパスです。

表 13. 必須の z/OS 環境変数 (続き)

環境変数	値
STEPLIB	<p>次の IBM MQ データ・セットを含む必要があります。</p> <ul style="list-style-type: none"> • SCSQAUTH • SCSQANLE • SCSQLOAD <p>また、z/OS システム上でデータベース・ロガー・コンポーネントを実行する場合は、STEPLIB に、以下の Db2 データ・セットがこの表示順で組み込まれている必要があります。</p> <ul style="list-style-type: none"> • SDSNEXIT • SDSNLOD2 • SDSNLOAD
LIBPATH	<p>z/OS UNIX システム・サービス・スペース内の IBM MQJava ライブラリーの場所を含める必要があります (IBM MQ 8.0 の場合、デフォルトは /mqm/V8R0M0/java/lib です)。</p>

以下は、Managed File Transfer の環境変数を正しく構成している .profile の例です。

```
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
```

また、必要に応じて、以下の環境変数を設定することもできます。

表 14. オプションの z/OS の環境変数

環境変数	値
BFG_WTO	<p>以下のいずれかの値を指定すると、BFG_WTO が使用可能になります。</p> <ul style="list-style-type: none"> • YES • ON • TRUE <p>以下のいずれかの値を指定すると、BFG_WTO が使用不可になります。これらの値は、大文字と小文字を区別しない。</p> <ul style="list-style-type: none"> • NULL • NO • OFF • FALSE <p>z/OS のロギングを使用可能にします。この環境変数は、デフォルトでは使用不可になります。</p> <p>エージェントのイベント・ログに書き込まれるメッセージは、z/OS のオペレーター・ログ機能にも書き込まれます。これにより、JCL からエージェントを実行した際に、自動化製品のアクセスがより簡単に行えます。宛先コードは「プログラマー情報」(11)で、記述子コードは「通知」(12)です。</p>

表 14. オプションの z/OS の環境変数 (続き)

環境変数	値
BFG_GROUP_NAME	<p>通常、mqm ファイル・グループは、Managed File Transfer 構成のデータ・ファイルおよびコマンドに関連付けられます。したがって、mqm グループのメンバーであるすべてのユーザーが Managed File Transfer 構成へのアクセスと変更を行うことができます。詳しくは、350 ページの『IBM MQ の MFT でのファイル・システムの権限』を参照してください。</p> <p>z/OS システムの場合、ファイル・グループは USS ファイル・システム・エンティティであり、mqm ファイル・グループは必ずしも定義されません。BFG_GROUP_NAME 環境変数を使用すると、Managed File Transfer 構成データ・ファイル用に代わりの既存の z/OS USS ファイル・システム・グループを定義できます。例えば、USS シェルのプロンプトに次のように入力します。</p> <pre data-bbox="857 793 1471 869">export BFG_GROUP_NAME=FTEGB</pre> <p>このようにすると、現在の USS セッションでこれ以後に作成するすべての構成ファイルにグループ FTEGB を関連付ける操作が定義されます。</p> <p>BFG_GROUP_NAME は、ブランク値に設定することも、削除することもできます。</p> <p>V9.0.0.1 BFG_GROUP_NAME 環境変数が非ブランクの値に設定されている z/OS システムで fteMigrateAgent コマンドを実行すると、このコマンドによって、ユーザーが BFG_GROUP_NAME 変数で指定されたグループのメンバーであるかどうかを確認されます。ユーザーが指定されたグループに属していない場合、コマンドはエラー・メッセージ BFGCL0502E: 「要求された操作を実行する権限がありません。」を報告することがあります。実行に失敗しました。このコマンドを正常に実行するためにユーザーが満たす必要がある基準について詳しくは、483 ページの『fteMigrateAgent: FTE 7.0 エージェントの MQ 7.5 以降へのマイグレーション』を参照してください。</p>

z/OS 構成プロパティ・ファイル

Managed File Transfer で使用されるプロパティの要約。

- [540 ページの『MFT coordination.properties ファイル』](#)
- [546 ページの『MFT command.properties ファイル』](#)
- [550 ページの『MFT agent.properties ファイル』](#)
- [ロガー構成プロパティ・ファイル](#)

Managed File Transfer は、ARM を使用可能なアプリケーションです。

始める前に

ARM の使用可能化と、ご使用のシステムに合わせて ARM ポリシーを定義する方法については、[z/OS 自動再始動管理プログラム \(ARM\) の使用](#)を参照してください。

MFT DB ロガー機能を使用して、再始動と Db2 データベースへの再接続を自動的に実行する場合は、ARM が唯一の有効な再始動マネージャーになります。

このタスクについて

ARM を使用すると、エージェント/ロガーのプロパティ `armELEMTYPE` および `armELEMENT` を設定することにより、エージェントおよびロガーの再始動を構成できます。プロパティ `armELEMTYPE` では ARM エLEMENT のタイプを定義し、プロパティ `armELEMENT` には ARM を登録するELEMENT の名前を指定します。

- エージェントの ELEMTYPE には `SYSBFGAG` を設定し、`armELEMENT` にはエージェント名に対応する名前を設定できます。
- ロガーの ELEMTYPE には `SYSBFGLG` を設定し、`armELEMENT` にはロガー名に対応する名前を設定できます。

注: ARM で再始動するように構成したエージェントとロガーは、バッチ・ジョブと開始済みタスクからのみ正常に実行できます。そのようなエージェントまたはロガーを USS コマンド行から直接開始しようとすると、操作は失敗し、ARM エラーの理由コードが出されます。

例

次に示す再始動ポリシーの例では、エージェント `BFGFT7CAG1` をキュー・マネージャー `FT7C` に依存するものとして定義しています。

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

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

JZOS Java ランチャーで Managed File Transfer for z/OS を使用する

このトピックの説明は、ご使用の IBM MQ for z/OS システムで、エンタープライズ内で Managed File Transfer を使用する代わりに方法として適用できます。

概要

Managed File Transfer for z/OS (MFT) には、z/OS の標準的なインストール手順を使用します。MFT コマンドを実行する別の方法として、JCL と JZOS Java ランチャーを使用することもできます。

詳しくは、『[JZOS Batch Launcher and Toolkit](#)』を参照してください。

JCL を正しく処理できない場合は、[301 ページの『JZOS でよくある MFT 問題』](#)を参照してください。

IBM MQ 8.0 以降の JCL の例



重要: IBM WebSphere MQ File Transfer Edition 7.0 では、パラメーターは、`BFG_` ではなく `FTE_` で始まります。

```
//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
/*
```

ここで、

- (1) は、組み込み JCL ステートメントの場所です
- (2) 1) の場所にある指定の JCL メンバーを組み込みます
- (2A) これは //STDENV を拡張します (以下を参照)
- (3) これが、実行されるコマンドです (接頭部 fte は付けない)
- (4) これらの行は必須であり、JZOS の情報をセットアップします
- (5) コマンドへのパラメーター
- BFGJCL8 メンバー (独自の名前を選択できます) は JZOS を呼び出します。このメンバーには、STEPLIB および MFT の実行に必要なその他の JCL があります。

組み込む必要があるその他の JCL

IBM MQ for z/OS ライブラリーのための JCL と、Db2 ロガーを使用する必要がある場合は Db2 ライブラリーのための JCL を組み込む必要があります。

以下に例を示します。

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

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

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

ここで、

- (1) は、JZOS プログラムの名前です。SYS1.SIEALNKE でシステム上のバージョンを探します。追加の診断を行う場合は、PARM='+T' を追加します。
- (2) これが、JZOS プログラムが含まれているデータ・セットです。
- (3) これは、シェル・スクリプトのメンバー名です。これは、MFTに必要なパラメーターを定義します。93 ページの『[MFT を定義するためのシェル・スクリプト](#)』を参照してください。

任意のデータ・セットおよびメンバーを指定できます。JCL ジョブはこれを拡張するため、ファイルの最後にする必要があります。91 ページの『[IBM MQ 8.0 以降の JCL の例](#)』の 2A を参照してください。

MFT を定義するためのシェル・スクリプト

92 ページの『[組み込む必要があるその他の JCL](#)』の例では、メンバー BFGZENV8 が使用されています。これは、JZOS プロファイルに基づきます。

MFT V8 と IBM WebSphere MQ File Transfer Edition 7.0 では同じ構成ファイルを使用できますが、少し変更が必要です。前述のとおり、MFT V8 より前のバージョンでは、パラメーターが FTE で始まります。93 ページの『[ファイルの例](#)』を参照してください。

以下を把握しておく必要があります。

- Java がインストールされている場所
- IBM MQ for z/OS Java ライブラリーの場所
- MFT ファイルの場所
- ユーザー ID が IBM MQ for z/OS 管理者として見なされるためには、特定のグループに属している必要があります。そのグループの名前が必要です。
- メッセージで英語を使用していない場合は、指定する言語を把握しておく必要があります。

ファイルの例

```
# 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=" "
```

ここで、

```
export _BPXK_AUTOCVT="ON"
```

Unicode 変換に必要です。

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

Java ディレクトリーの場所です。Java のパスの名前を指定します。このディレクトリーには、bin ディレクトリーおよびその他のディレクトリーが含まれています。

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

Java 実行可能ステートメントのパス・ステートメントをセットアップします。

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

Java 実行可能ステートメントのライブラリー・パスをセットアップします。

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

LIBPATH ステートメントに Java ライブラリーを追加します。

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

ライブラリー・パスに IBM MQ for z/OS ライブラリーを追加します。USS の IBM MQ for z/OS ライブラリーの名前を指定します。

export LIBPATH

LIBPATH を JZOS で使用できるようにします。

export BFG_JAVA_HOME="\${JAVA_HOME}"

BFG_JAVA_HOME に、上記で指定した JAVA_HOME の値を設定します。

export BFG_WTO="YES"

BFG_WTO を YES に設定すると、WTO を使用するジョブ・ログにメッセージが表示されます。

export BFG_GROUP_NAME=MQADM

指定したグループのメンバーであるユーザー ID は、IBM MQ for z/OS 管理者と見なされます。

export BFG_PROD="/HMF8800/"

MFT コードが置かれているパスです。

export BFG_DATA="/u/johndoe/fteconfig"

MFT 構成情報が保管されている場所です。

export BFG_LANG="-Duser.language=de "

言語をドイツ語として定義するステートメントをコメント化したものです。

export BFG_LANG=" "

言語にデフォルト (英語) を指定します。

/lib/messages/BFGNVMessages_*.properties の MFT 製品のコンテンツに、使用可能な言語がリストされています。デフォルトにするには、値をブランクにします。これは、英語が使用されることを意味します。

V7 では、以下を指定します。

```
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=" "
```

/u/johndoe/fteconfig は、BFG_DATA のものとは異なることに注意してください。

関連情報

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

計画 [Managed File Transfer](#)

IBM i IBM i での MFT の構成

Managed File Transfer のインストール後に、この製品の使用を開始するには、調整キュー・マネージャーおよびエージェントの構成の一部を実行する必要があります。

このタスクについて

インストールの完了後、調整キュー・マネージャーおよびエージェントを使用してファイルを転送できるようにするには、その前に、Managed File Transfer で提供されている、新規調整キュー・マネージャーおよび新規エージェント用の構成スクリプトを実行する必要があります。その後、作成したエージェントを開始する必要があります。

手順

1. 新規のすべての調整キュー・マネージャーの場合: `coordination_qmgr_name.mqsc` ファイル内の MQSC コマンドを調整キュー・マネージャーに対して実行します。調整キュー・マネージャーがインストール済み環境と同じコンピューター上にない場合、キュー・マネージャーがあるコンピューターに MQSC スクリプト・ファイルをコピーしてから、スクリプトを実行します。
 - a) IBM i のコマンド行で、CALL QSHELL コマンドを使用して qshell を開始します。
 - b) 次のディレクトリーに移動します: /QIBM/UserData/mqm/mqft/config/
`coordination_qmgr_name`

- c) `coordination_qmgr_name` をキュー・マネージャーの名前に置き換えて、以下のコマンドを発行します。

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

この代わりに、調整キュー・マネージャーを手動で構成できます。詳しくは、[98 ページの『MFT 用の調整キュー・マネージャーの構成』](#)を参照してください。

2. すべての新規エージェントの場合: エージェント・キュー・マネージャーに MQSC コマンドを `agent_name_create.mqsc` ファイルで実行します。

エージェントのキュー・マネージャーがそのエージェントと同じコンピューター上にない場合、キュー・マネージャーがあるコンピューターに MQSC スクリプト・ファイルをコピーしてから、スクリプトを実行します。

- a) IBM i のコマンド行で、CALL QSHELL コマンドを使用して qshell を開始します。
- b) 次のディレクトリーに移動します: `/QIBM/UserData/mqm/mqft/config/agent_qmgr_name/agents`
- c) `agent_qmgr_name` をエージェント・キュー・マネージャーの名前に置き換え、`agent_name` をエージェントの名前に置き換えて、以下のコマンドを発行します。

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

この代わりに、エージェント・キュー・マネージャーを手動で構成できます。詳しくは、[99 ページの『MFT エージェント・キュー・マネージャーの構成』](#)を参照してください。

3. インストールの一部として QMFT サブシステムをまだ開始していない場合は、IBM i コマンド行から、コマンド `STRSBS SBSDB(QMQMMFT/QMFT)` または `STRSBS QMQMMFT/QMFT` を使用して QMFT サブシステムを開始します。
4. **fteStartAgent** コマンドを使用して、新規エージェントを開始します。
 - a) IBM i のコマンド行で、CALL QSHELL コマンドを使用して qshell を開始します。
 - b) 次のディレクトリーに移動します: `/QIBM/ProdData/mqm/bin`
 - c) AGENT をエージェントの名前に置き換えて、以下のコマンドを発行します。

```
./fteStartAgent AGENT
```

次のタスク

エージェントがアクセスできるファイル・システムの領域を限定するために、サンドボックスをセットアップすることをお勧めします。この機能についての説明は [45 ページの『MFT エージェント・サンドボックスの処理』](#)にあります。

関連概念

[95 ページの『初めて使用する際の MFT の構成』](#)

Managed File Transfer のエージェントおよびキュー・マネージャーの構成タスクの中には、初回の使用時に一度実行すれば済むものもあります。

初めて使用する際の MFT の構成

Managed File Transfer のエージェントおよびキュー・マネージャーの構成タスクの中には、初回の使用時に一度実行すれば済むものもあります。

IBM MQ への接続

Managed File Transfer に関連した通信を含む、IBM MQ キュー・マネージャーとのネットワーク通信にはすべて、IBM MQ チャネルが関係します。IBM MQ チャネルはネットワーク・リンクの一端を表します。チャネルは、メッセージ・チャネルまたは MQI チャネルのいずれかに分類されます。

Managed File Transfer およびチャネル

Managed File Transfer は MQI チャネルを使用して、クライアント・モードでエージェントをそのエージェント・キュー・マネージャーに接続したり、コマンド・アプリケーション (例えば **fteCreateTransfer**) をそのコマンド・キュー・マネージャーや調整キュー・マネージャーに接続したりします。デフォルト構成では、これらの接続は SYSTEM.DEF.SVRCONN という SVRCONN チャネルを使用して確立されます。これは、デフォルトですべてのキュー・マネージャーに存在します。これらのデフォルトがあるので、基本 Managed File Transfer インストールでは、どの MQI チャネルも変更不要です。

メッセージ・チャネル・エンドポイントは 6 種類ありますが、このトピックでは送信側と受信側の対についてのみ扱います。他のチャネルの組み合わせについては、[分散キューイング・コンポーネント](#)を参照してください。

必要なメッセージ・パス

IBM MQ メッセージはメッセージ・チャネル経由でのみ移動が可能のため、Managed File Transfer が必要とするすべてのメッセージ・パスでチャネルを使用できるようにする必要があります。これらのパスは、直接的である必要はありません。メッセージは、必要に応じて、中間キュー・マネージャーを経由することができます。このトピックでは、直接的な Point-to-Point 通信についてのみ扱います。これらのオプションの詳細については、[リモート・キュー・マネージャーへのアクセス方法](#)を参照してください。

Managed File Transfer で使用される通信パスは、次のとおりです。

エージェントからエージェント

ファイルの転送が行われる 2 つのエージェントは、関連付けられているキュー・マネージャー間の双方向通信を必要とします。このパスは大量のデータを渡すため、必要に応じて、パスをできるだけ短く、高速で、低コストにすることを検討してください。

エージェントから調整

転送に参加するエージェントからのログ・メッセージは、調整キュー・マネージャーに到達できなければなりません。

コマンドからエージェント

コマンド・アプリケーションまたは IBM MQ Explorer (コマンド・キュー・マネージャーを使用する) が接続するどのキュー・マネージャーも、これらのコマンド・アプリケーションを使用して制御するエージェントのキュー・マネージャーにメッセージを送信できなければなりません。コマンドによってフィードバック・メッセージを表示できるようにするには、双方向接続を使用します。

詳しくは、企業で使用しているプラットフォーム用の『[IBM MQ インストール済み環境の検証](#)』を参照してください。

関連概念

[102 ページの『MFT と連動するマルチインスタンス・キュー・マネージャーの構成』](#)

IBM WebSphere MQ 7.0.1 以降では、マルチ・インスタンス・キュー・マネージャーの作成をサポートしています。マルチ・インスタンス・キュー・マネージャーはスタンバイ・サーバーで自動的に再始動します。Managed File Transfer は、マルチ・インスタンス・エージェント・キュー・マネージャー、マルチ・インスタンス調整キュー・マネージャー、およびマルチ・インスタンス・コマンド・キュー・マネージャーへの接続をサポートします。

関連タスク

[97 ページの『MFT ネットワーク・キュー・マネージャーの構成』](#)

ご使用の Managed File Transfer ネットワークに複数の IBM MQ キュー・マネージャーが含まれている場合、これらの IBM MQ キュー・マネージャーは、互いにリモート側で通信できる必要があります。

[98 ページの『MFT 用の調整キュー・マネージャーの構成』](#)

fteSetupCoordination コマンドを実行したら、`MQ_DATA_PATH/mqft/config/coordination_qmgr_name` ディレクトリー内の `coordination_qmgr_name.mqsc` スクリプトを実行して、調整キュー・マネージャーに関する必要な構成を行います。ただし、この構成を手動で行う場合は、調整キュー・マネージャーに対して以下の手順を実行します。

MFT ネットワーク・キュー・マネージャーの構成

ご使用の Managed File Transfer ネットワークに複数の IBM MQ キュー・マネージャーが含まれている場合、これらの IBM MQ キュー・マネージャーは、互いにリモート側で通信できる必要があります。

このタスクについて

キュー・マネージャーの相互通信を可能にするための構成には、次の 2 つの方法があります。

- IBM MQ キュー・マネージャー・クラスターをセットアップする方法。

IBM MQ キュー・マネージャー・クラスターの情報や構成方法については、『[キュー・マネージャー・クラスターの構成](#)』を参照してください。

- キュー・マネージャー間のチャンネルをセットアップする方法。この方法については、以下で説明します。

キュー・マネージャー間のチャンネルのセットアップ

キュー・マネージャー間に以下のメッセージ・チャンネルをセットアップします。

- エージェント・キュー・マネージャーから調整キュー・マネージャー
- コマンド・キュー・マネージャーからエージェント・キュー・マネージャー
- エージェント・キュー・マネージャーからコマンド・キュー・マネージャー (コマンドによってフィードバック・メッセージを表示できるようにするため)
- コマンド・キュー・マネージャーから調整キュー・マネージャー
- エージェント・キュー・マネージャーから Managed File Transfer のネットワーク内に含まれる他のエージェント・キュー・マネージャー

この通信のセットアップ方法についてさらに情報が必要な場合は、まず『[MQSC を使用したリモート IBM MQ オブジェクトの管理](#)』を参照してください。

提案されているサンプルとなるステップのいくつかは、以下のとおりです。

手順

1. IBM MQ のキュー・マネージャー上に、調整キュー・マネージャーと同じ名前で伝送キューを作成する。以下の MQSC コマンドを使用できます。

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

2. IBM MQ キュー・マネージャーで、Managed File Transfer の調整キュー・マネージャーへの送信側チャンネルを作成する。前のステップで作成された伝送キューの名前は、このチャンネルの必須パラメーターです。IBM WebSphere MQ 7.5 または Managed File Transfer エージェントで Managed File Transfer との通信が必要な場合は、送信側チャンネルの CONVERT パラメーターが no に設定されていることを確認してください。(以前のバージョンの IBM WebSphere MQ File Transfer Edition では、メッセージは常に UTF-8 形式でパブリッシュされていました。これは、データ変換によってメッセージが破損することを意味します。IBM MQ 8.0 以降の Managed File Transfer 上のエージェントの場合、メッセージはブランク形式でパブリッシュされるため、これは必要ありません。)

以下の MQSC コマンドを使用できます。

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

注: CONVERT(NO) は、必要なときにのみ設定してください。

3. Managed File Transfer の調整キュー・マネージャーで、IBM MQ キュー・マネージャーへの受信側チャンネルを作成する。この受信側チャンネルに、IBM MQ のキュー・マネージャー上の送信側チャンネルと同じ名前を付けます。

以下の MQSC コマンドを使用できます。

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

次のタスク

次いで、調整キュー・マネージャーの構成ステップとして、『[調整キュー・マネージャーの構成](#)』に従います。

MFT 用の調整キュー・マネージャーの構成

`fteSetupCoordination` コマンドを実行したら、`MQ_DATA_PATH/mqft/config/coordination_qmgr_name` ディレクトリー内の `coordination_qmgr_name.mqsc` スクリプトを実行して、調整キュー・マネージャーに関する必要な構成を行います。ただし、この構成を手動で行う場合は、調整キュー・マネージャーに対して以下の手順を実行します。

このタスクについて

手順

1. SYSTEM.FTE という名前のローカル・キューを作成します。
2. SYSTEM.FTE キューを SYSTEM.QPUBSUB.QUEUE.NAMELIST 名前リストに追加します。
3. SYSTEM.FTE という名前のトピックを、トピック・ストリング SYSTEM.FTE で作成します。
4. SYSTEM.FTE トピックの非永続メッセージ配信 (NPMSGDLV) 属性と永続メッセージ配信 (PMSGDLV) 属性が ALLAVAIL に設定されていることを確認します。
5. 調整キュー・マネージャーのパブリッシュ/サブスクライブ・モード (PSMODE) 属性が ENABLED に設定されていることを確認します。

次のタスク

調整キュー・マネージャーとして構成されているキュー・マネージャーで `strmqm -c` コマンドを実行すると、このコマンドは [ステップ 2](#) で行われた変更を削除します (SYSTEM.FTE キューを SYSTEM.QPUBSUB.QUEUE.NAMELIST 名前リストに追加します)。これは、`strmqm -c` がデフォルトの IBM MQ オブジェクトを再作成し、Managed File Transfer の変更を元に戻すためです。したがって、`strmqm -c` を使用してキュー・マネージャーを開始した場合は、以下のいずれかの手順を実行します。

- キュー・マネージャーで再度 `coordination_qmgr_name.mqsc` スクリプトを実行します。
- [ステップ 2](#) を繰り返します。

関連概念

[95 ページの『IBM MQ への接続』](#)

Managed File Transfer に関連した通信を含む、IBM MQ キュー・マネージャーとのネットワーク通信にはすべて、IBM MQ チャンネルが関係します。IBM MQ チャンネルはネットワーク・リンクの一端を表します。チャンネルは、メッセージ・チャンネルまたは MQI チャンネルのいずれかに分類されます。

[102 ページの『MFT と連動するマルチインスタンス・キュー・マネージャーの構成』](#)

IBM WebSphere MQ 7.0.1 以降では、マルチ・インスタンス・キュー・マネージャーの作成をサポートしています。マルチ・インスタンス・キュー・マネージャーはスタンバイ・サーバーで自動的に再始動します。Managed File Transfer は、マルチ・インスタンス・エージェント・キュー・マネージャー、マルチ・インスタンス調整キュー・マネージャー、およびマルチ・インスタンス・コマンド・キュー・マネージャーへの接続をサポートします。

関連タスク

[97 ページの『MFT ネットワーク・キュー・マネージャーの構成』](#)

ご使用の Managed File Transfer ネットワークに複数の IBM MQ キュー・マネージャーが含まれている場合、これらの IBM MQ キュー・マネージャーは、互いにリモート側で通信できる必要があります。

関連資料

514 ページの『[fteSetupCoordination](#)』

fteSetupCoordination コマンドは、Managed File Transfer のプロパティ・ファイルと調整キュー・マネージャー・ディレクトリーを作成します。

MFT エージェント・キュー・マネージャーの構成

インストール後に、`MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` ディレクトリー内の `agent_name_create.mqsc` スクリプトを実行して、エージェント・キュー・マネージャーに関する必要な構成を行います。ただし、この構成を手動で行う場合は、エージェント・キュー・マネージャーに対して以下の手順を実行します。

手順

1. エージェント操作キューを作成します。

これらのキューの名前は以下のとおりです。

- `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`

キューのパラメーターについて詳しくは、[620 ページの『MFT エージェント・キュー設定』](#)を参照してください。

2. エージェント権限キューを作成します。

これらのキューの名前は以下のとおりです。

- `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`

キューのパラメーターについて詳しくは、[620 ページの『MFT エージェント・キュー設定』](#)を参照してください。

次のタスク

プロトコル・ブリッジ・エージェントの作成と構成について詳しくは、[393 ページの『fteCreateBridgeAgent \(MFT プロトコル・ブリッジ・エージェントの作成および構成\)』](#) および [229 ページの『FTPS サーバー用のプロトコル・ブリッジの構成』](#)を参照してください。

IBM MQ File Transfer 構造の作成

Managed File Transfer 構造は、同じマシン上のキュー・マネージャーに接続されている単一のエージェントに基づいて構成できます。

このタスクについて

MFT の構成は、エージェントがあるマシン上の IBM MQ DataPath の下のファイル構造に保管されます。

以下のサンプル構成は、SAMPLECOORD (セキュリティー無効) という名前の MFT IBM MQ 8.0 キュー・マネージャーと、SAMPLEAGENT という名前の単一の MFT エージェントを対象としています。

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

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

この例は、キュー・マネージャーのセキュリティーが無効になっていると想定しています。以下のコマンドは **runmqsc** で実行され、キュー・マネージャーの再始動後にセキュリティーを無効にします。

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

IBM MQ 8.0 以降の MFT でセキュリティーを有効にした構成の場合、**CONNAUTH** では、キュー・マネージャーに接続するすべての MFT コマンドでユーザー ID とパスワードの資格情報を提供する必要があります。各コマンドに追加のパラメーター **-mquserid** および **-mqpassword** を適用するか、MQMFTCredentials.xml ファイルを定義することができます。以下のサンプル資格情報ファイルでは、キュー・マネージャー SAMPLECOORD に接続する際に、ユーザー ID として **fteuser**、そのパスワードとして **MyPassword** を使用するように定義されています。

```
<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>
```

詳しくは、[42 ページの『MFT と IBM MQ の接続認証』](#)を参照してください。

注:

- MFT の構成ディレクトリーを見つけるには、**fteDisplayVersion -v** コマンドを使用します。
- z/OS ユーザーの場合、MQMFTCredential.xml ファイルは、変数レコード・フォーマット (RECFM=V)、または不定形式レコード・フォーマット (RECFM=U) の区分データ・セットのメンバーとして配置できます。
- セキュリティーを有効にした構成の場合は、以下のステップにパラメーター **-credentialsFile full credential file path** を追加して、資格情報を関連するキュー・マネージャーに関連付けます。
- 次のコマンドを使用して、MQMFTCredential.xml の平文パスワードを難読化することができます。

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

手順

1. 調整キュー・マネージャーを作成します。

調整キュー・マネージャーとは、単一のキュー・マネージャーであり、すべての転送ログおよび状況情報をそのエージェントから受け取るために使用されます。以下のコマンドを実行します。

```
fteSetupCoordination -coordinationQMgr coordination_qmgr_name
```

これにより、基本的な最上位構成が作成され、`coordination_qmgr_name.mqsc` を呼び出す IBM MQ スクリプト・ファイルが作成されます。

その後、次の IBM MQ コマンドを実行して構成をキュー・マネージャーにロードする必要があります。

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

注: キュー・マネージャーへの TCP クライアント接続では、以下を使用できます。

```
fteSetupCoordination -coordinationQMgr coordination_qmgr_name  
-coordinationQMgrHost coordination_qmgr_host -coordinationQMgrPort coordination_qmgr_port  
-coordinationQMgrChannel coordination_qmgr_channel
```

作成された `coordination_qmgr_name.mqsc` では、調整キュー・マネージャーが実行されているマシンで **runmqsc** コマンドを実行する必要があります。

2. コマンド・キュー・マネージャーを作成します。

コマンド・キュー・マネージャーとは、単一のキュー・マネージャーであり、これは、IBM MQ インフラストラクチャーが MFT 要求を関連エージェントに経路指定できるように事前構成されています。以下のコマンドを実行します。

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

これにより、調整ディレクトリーに `command.properties` ファイルが作成されます。-p はオプションであり、コマンドがデフォルト調整のためにセットアップされる場合には必要ないことに注意してください。

注: キュー・マネージャーへの TCP クライアント接続では、以下を使用できます。

```
fteSetupCommands -p coordination_qmgr_name -commandQMgr connection_qmgr_name  
-commandQMgrHost connection_qmgr_host -commandQMgrPort connection_qmgr_port  
-commandQMgrChannel connection_qmgr_channel
```

3. エージェントを作成します。

エージェントとは、ファイルを送受信できるアプリケーションのことです。以下のコマンドを実行します。

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

これにより、調整の下にエージェント構成が作成され、エージェントの構成ディレクトリーに `agent_name.mqsc` を呼び出すための IBM MQ スクリプト・ファイルが作成されます。

次の IBM MQ コマンドを実行して、IBM MQ スクリプト・ファイルをキュー・マネージャーにロードします。

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

注: キュー・マネージャーへの TCP クライアント接続では、以下を使用できます。

```
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. エージェントを開始します。

以下のコマンドを実行します。

```
fteStartAgent -p coordination_qmgr_name agentName
```

エージェントはバックグラウンドで開始され、コマンド・プロンプトが返されます。エージェントが実行されていることを確認するには、次のコマンドを実行します。

```
fteListAgents -p coordination_qmgr_name
```

これにより、エージェントの状況が表示されます。エージェントが正常に実行されている場合、READY 状態にあると報告されます。

タスクの結果

MFT の基本インフラストラクチャーの使用準備が整い、**fteCreateTransfer** コマンドを使用して転送を要求できるようになります。あるいは、IBM MQ Explorer を使用できる場合は、MFT プラグインを使用して転送を作成およびモニターします。

『ステップ 3: エージェントを作成します』を繰り返して、さらにエージェントを構成に追加できます。TCP クライアント接続を使用する場合、これらのエージェントを複数の異なるマシンに配置することができます。異なるマシンに配置する場合には **fteSetupCoordination** コマンドと **fteSetupCommands** コマンドを各マシンで繰り返す必要がありますが、mqsc スクリプトを実行する必要はありません。

より複雑な構成の場合は、調整および各エージェントにそれぞれ別個のキュー・マネージャーを設定することができます。その場合には、各種キュー・マネージャーを接続する必要があります。

関連資料

[514 ページの『fteSetupCoordination』](#)

fteSetupCoordination コマンドは、Managed File Transfer のプロパティ・ファイルと調整キュー・マネージャー・ディレクトリーを作成します。

[512 ページの『fteSetupCommands: MFT command.properties ファイルの作成』](#)

fteSetupCommands コマンドは、Managed File Transfer command.properties ファイルを作成します。このプロパティ・ファイルは、コマンドの発行時に IBM MQ ネットワークに接続するキュー・マネージャーの詳細を指定します。

[388 ページの『fteCreateAgent \(MFT エージェントの作成\)』](#)

fteCreateAgent コマンドは、Managed File Transfer Agent とその関連構成を作成します。

[494 ページの『fteObfuscate: 機密データの暗号化』](#)

fteObfuscate コマンドは、資格情報ファイル内の機密データを暗号化します。これにより、他のユーザーが資格情報ファイルにアクセスした際に、内容が読み取られるのを防ぐことができます。

[810 ページの『MFT 資格情報ファイル・フォーマット』](#)

MQMFTCredentials.xml ファイルには、ユーザー ID とパスワードに関する重要な情報が含まれています。MQMFTCredentials.xml ファイル内のエレメントは MQMFTCredentials.xsd スキーマに準拠する必要があります。資格情報ファイルのセキュリティはユーザーの責任です。

[550 ページの『MFT agent.properties ファイル』](#)

各 Managed File Transfer Agent には、独自のプロパティ・ファイル agent.properties があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。agent.properties ファイルには、エージェントの動作を変更するプロパティを含めることもできます。

[302 ページの『MFT エージェントが fteListAgents コマンドによってリストされない場合に行う事柄』](#)

エージェントが **fteListAgents** コマンドによってリストされないか、IBM MQ Explorer に表示されない場合、または IBM MQ Explorer の「転送ログ」にファイル転送が表示されない場合は、いくつかの問題判別ステップを実行して原因を調査することができます。

MFT と連動するマルチインスタンス・キュー・マネージャーの構成

IBM WebSphere MQ 7.0.1 以降では、マルチ・インスタンス・キュー・マネージャーの作成をサポートしています。マルチ・インスタンス・キュー・マネージャーはスタンバイ・サーバーで自動的に再始動します。

Managed File Transfer は、マルチ・インスタンス・エージェント・キュー・マネージャー、マルチ・インスタンス調整キュー・マネージャー、およびマルチ・インスタンス・コマンド・キュー・マネージャーへの接続をサポートします。

マルチ・インスタンス・キュー・マネージャーの構成

重要: IBM MQ マルチ・インスタンス・キュー・マネージャーの構成については、[マルチ・インスタンス・キュー・マネージャー](#)を参照してください。Managed File Transfer と連携するように複数インスタンス・キュー・マネージャーを構成する前に、この情報をお読みください。

エージェント・キュー・マネージャーとしてのマルチ・インスタンス・キュー・マネージャーの使用

エージェントがマルチ・インスタンス・キュー・マネージャーのアクティブ・インスタンスとスタンバイ・インスタンスの両方に接続できるようにするには、agentQMGrStandby プロパティをエージェントの agent.properties ファイルに追加します。agentQMGrStandby プロパティは、スタンバイ・キュー・マネージャー・インスタンスについてクライアント接続に使用されるホスト名とポート番号を定義します。プロパティの値は MQ CONNAME フォーマット、つまり *host_name(port_number)* で指定する必要があります。

agentQMGr プロパティは、マルチ・インスタンス・キュー・マネージャーの名前を指定します。agentQMGrHost プロパティは、アクティブ・キュー・マネージャー・インスタンスのホスト名を示し、agentQMGrPort プロパティは、アクティブ・キュー・マネージャー・インスタンスのポート番号を示します。エージェントは、マルチ・インスタンス・キュー・マネージャーのアクティブ・インスタンスとスタンバイ・インスタンスの両方にクライアント・モードで接続する必要があります。

詳しくは、『550 ページの『MFT agent.properties ファイル』』を参照してください。

以下の例は、QM_JUPITER というマルチ・インスタンス・キュー・マネージャーに接続する、AGENT1 の agent.properties ファイルの内容を示しています。QM_JUPITER のアクティブ・インスタンスは、システム host1 上にあり、クライアント接続にポート番号 1414 を使用します。QM_JUPITER のスタンバイ・インスタンスは、システム host2 上にあり、クライアント接続にポート番号 1414 を使用します。

```
agentName=AGENT1
agentDesc=
agentQMGr=QM_JUPITER
agentQMGrPort=1414
agentQMGrHost=host1
agentQMGrChannel=SYSTEM.DEF.SVRCONN
agentQMGrStandby=host2(1414)
```

調整キュー・マネージャーとしてのマルチ・インスタンス・キュー・マネージャーの使用

マルチ・インスタンス調整キュー・マネージャーのアクティブ・インスタンスとスタンバイ・インスタンスの両方の接続を可能にするには、coordinationQMGrStandby プロパティを Managed File Transfer トポロジーのすべての coordination.properties ファイルに追加します。

詳しくは、『540 ページの『MFT coordination.properties ファイル』』を参照してください。

以下の例は、QM_SATURN というマルチ・インスタンス調整キュー・マネージャーへの接続の詳細を示す、coordination.properties ファイルの内容を示しています。QM_SATURN のアクティブ・インスタンスは、システム coordination_host1 上にあり、クライアント接続にポート番号 1420 を使用します。QM_SATURN のスタンバイ・インスタンスは、システム coordination_host2 上にあり、クライアント接続にポート番号 1420 を使用します。

```
coordinationQMGr=QM_SATURN
coordinationQMGrHost=coordination_host1
coordinationQMGrPort=1420
coordinationQMGrChannel=SYSTEM.DEF.SVRCONN
coordinationQMGrStandby=coordination_host2(1420)
```

Managed File Transfer スタンドアロン・ロガーは、そのキュー・マネージャーと常にバインディング・モードで接続している必要があります。スタンドアロン・ロガーをマルチ・インスタンス調整キュー・マネージャーと共に使用している場合、スタンドアロン・ロガーをバインディング・モードで別のキュー・マネージャーに接続してください。これを行うステップについては、『127 ページの『MFT スタンドアロン・ロガーの代替構成』』で説明されています。スタンドアロン・ロガーのキュー・マネージャーと調整キュー・マネージャーの間のチャンネルを、マルチ・インスタンス調整キュー・マネージャーの両方のインスタンスのホスト名およびポート番号で定義する必要があります。その方法について詳しくは、『複数インスタンス・キュー・マネージャー』を参照してください。

IBM MQ Explorer 用の Managed File Transfer プラグインは、調整キュー・マネージャーにクライアント・モードで接続します。マルチ・インスタンス調整キュー・マネージャーのアクティブ・インスタンスが失敗すると、調整キュー・マネージャーのスタンバイ・インスタンスがアクティブになり、プラグインが再接続します。

Managed File Transfer コマンド **fteList*** および **fteShowAgentDetails** は、調整キュー・マネージャーに直接接続します。マルチ・インスタンス調整キュー・マネージャーのアクティブ・インスタンスが使用不可になると、これらのコマンドは調整キュー・マネージャーのスタンバイ・インスタンスに接続しようとしします。

コマンド・キュー・マネージャーとしてのマルチ・インスタンス・キュー・マネージャーの使用

マルチ・インスタンス・コマンド・キュー・マネージャーのアクティブ・インスタンスとスタンバイ・インスタンスの両方の接続を使用可能にするには、`connectionQMGrStandby` プロパティを Managed File Transfer トポロジーのすべての `command.properties` ファイルに追加します。

詳しくは、『546 ページの『MFT `command.properties` ファイル』』を参照してください。

以下の例は、QM_MARS というマルチ・インスタンス・コマンド・キュー・マネージャーへの接続の詳細を示す、`command.properties` ファイルの内容を示しています。QM_MARS のアクティブ・インスタンスは、システム `command_host1` 上にあり、クライアント接続にポート番号 1424 を使用します。QM_MARS のスタンバイ・インスタンスは、システム `command_host2` 上にあり、クライアント接続にポート番号 1424 を使用します。

```
connectionQMGr=QM_SATURN
connectionQMGrHost=command_host1
connectionQMGrPort=1424
connectionQMGrChannel=SYSTEM.DEF.SVRCONN
connectionQMGrStandby=command_host2(1424)
```

関連概念

95 ページの『IBM MQ への接続』

Managed File Transfer に関連した通信を含む、IBM MQ キュー・マネージャーとのネットワーク通信にはすべて、IBM MQ チャンネルが関係します。IBM MQ チャンネルはネットワーク・リンクの一端を表します。チャンネルは、メッセージ・チャンネルまたは MQI チャンネルのいずれかに分類されます。

関連タスク

97 ページの『MFT ネットワーク・キュー・マネージャーの構成』

ご使用の Managed File Transfer ネットワークに複数の IBM MQ キュー・マネージャーが含まれている場合、これらの IBM MQ キュー・マネージャーは、互いにリモート側で通信できる必要があります。

98 ページの『MFT 用の調整キュー・マネージャーの構成』

fteSetupCoordination コマンドを実行したら、`MQ_DATA_PATH/mqft/config/coordination_qmgr_name` ディレクトリー内の `coordination_qmgr_name.mqsc` スクリプトを実行して、調整キュー・マネージャーに関する必要な構成を行います。ただし、この構成を手動で行う場合は、調整キュー・マネージャーに対して以下の手順を実行します。

MFT ログ・メッセージの保持

Managed File Transfer は、ファイル転送進行情報およびログ情報を調整キュー・マネージャーに送信します。調整キュー・マネージャーは、この情報を SYSTEM.FTE トピックに一致するサブスクリプションにパブリッシュします。サブスクリプションがない場合、この情報は保持されません。

転送進行情報またはログ情報が業務上重要な場合、以下のステップのいずれかを実行して、情報が確実に保持されるようにします。

- Managed File Transfer のデータベース・ロガーを使用して、Oracle または Db2 データベースの SYSTEM.FTE/Log トピックにパブリッシュされるメッセージをコピーします。
- サブスクリプションを SYSTEM.FTE トピックに定義する。これにより、IBM MQ キューにパブリケーションが保管されます。ファイル転送を行う前にこのサブスクリプションを定義して、すべての進行メッセージとログ・メッセージがキューに確実に保持されるようにします。
- メッセージ・キュー・インターフェース (MQI) または IBM MQ JMS を使用して永続サブスクリプションを作成し、そのサブスクリプションに送信されるパブリケーションを処理するアプリケーションを作成する。このアプリケーションは、すべての進行メッセージとログ・メッセージを確実に受け取るように、ファイルが転送される前に稼働している必要があります。

これらの各手法は、この後のセクションで詳しく説明します。

ログ情報の保存のために IBM MQ Explorer ・プラグインに依存することはしないでください。

ログ・メッセージ保存のための Managed File Transfer データベース・ロガーの使用

データベース・ロガーは、Managed File Transfer のオプションのコンポーネントです。このコンポーネントを使用して、分析および監査のために、ログ情報をデータベースにコピーできます。データベース・ロガーは、調整キュー・マネージャーおよびデータベースをホストするシステムにインストールする、スタンドアロンの Java アプリケーションです。データベース・ロガーについては、[106 ページの『MFT ロガーの構成』](#)を参照してください。

IBM MQ エクスプローラー・プラグインを使用した進行メッセージおよびログ・メッセージの保持

IBM MQ Explorer ・プラグインのインスタンスが初めて開始される時に、インスタンスにより永続サブスクリプションが調整キュー・マネージャーに作成されます。この永続サブスクリプションを使用して、「**転送ログ**」および「**現在の転送進行状況**」ビューに表示される情報が収集されます。

永続サブスクリプションの名前には、そのサブスクリプションが IBM MQ Explorer MFT プラグインで作成されたことと、ホスト名、およびユーザーの名前を示す接頭部が付きます (例えば、MQExplorer_MFT_Plugin_HOST_TJWatson)。

この接頭部は、IBM MQ Explorer ・プラグインのインスタンスにより実際にはもう使用されていない永続サブスクリプションを管理者が削除する場合に備えて追加されます。

調整キュー・マネージャーで永続サブスクリプションを使用すると、SYSTEM.MANAGED.DURABLE キューにメッセージが蓄積されていく可能性があります。大容量の Managed File Transfer ネットワークを使用している場合、IBM MQ Explorer プラグインを頻繁に使用するか、または両方を使用すると、このメッセージ・データがローカル・ファイル・システムに入力される可能性があります。

これを回避するには、IBM MQ Explorer ・プラグインで調整キュー・マネージャーに対する非永続サブスクリプションを使用するように指定できます。IBM MQ Explorer で以下の手順を実行してください。

1. 「ウィンドウ」 > 「設定」 > 「MQ エクスプローラー」 > 「Managed File Transfer」を選択します。
2. 「転送ログ・サブスクリプション・タイプ」リストから「NON_DURABLE」を選択します。

IBM MQ キューへのパブリケーションの保管

IBM MQ キューにログ・メッセージまたは進行メッセージを保管するには、メッセージをこのキューに転送する調整キュー・マネージャー上にサブスクリプションを構成します。例えば、すべてのログ・メッセージを LOG.QUEUE という名前のキューに転送するには、次の MQSC コマンドを送信します。

```
define sub(MY.SUB) TOPICSTR('Log/#') TOPICOBJ(SYSTEM.FTE) DEST(LOG.QUEUE) WSCHEMA(TOPIC)
```

ログ・メッセージは IBM MQ キューに転送された後、そのキューを使用する IBM MQ アプリケーションによって処理されるまでキューに残ります。

SYSTEM.FTE トピックに対する永続サブスクリプションを管理するアプリケーションの作成

IBM MQ によってサポートされるアプリケーション・プログラミング・インターフェースのいずれかを使用して、SYSTEM.FTE トピックに対する独自の永続サブスクリプションを管理するアプリケーションを作成することができます。これらのアプリケーションは、IBM MQ キューまたはログ・メッセージを受け取り、ビジネス・ニーズに合わせて、それらを適切に処理できます。

使用可能なアプリケーション・プログラミング・インターフェースについて詳しくは、『[アプリケーションの開発](#)』を参照してください。

MFT ロガーの構成

Managed File Transfer は、ファイルを転送時に、調整キュー・マネージャーのトピックに対して、処置に関する情報をパブリッシュします。データベース・ロガーは、Managed File Transfer のオプションのコンポーネントです。このコンポーネントを使用して、分析および監査のために、この情報をデータベースにコピーできます。

ロガーには以下の 3 つのバージョンがあります。

- スタンドアロン・ファイル・ロガー
- スタンドアロン・データベース・ロガー
- Java Platform, Enterprise Edition (Java EE) ロガー

重要: Managed File Transfer ロガーは IBM i プラットフォームではサポートされません。

スタンドアロン・ファイル・ロガー

スタンドアロン・ファイル・ロガーは、調整キュー・マネージャーをホストするシステム、または調整キュー・マネージャーに接続されているキュー・マネージャーをホストするシステムで実行される Java プロセスです。スタンドアロン・ファイル・ロガーは、IBM MQ バインディングを使用して、関連付けられているキュー・マネージャーに接続します。スタンドアロン・ロガーは、**fteCreateLogger** コマンドを使用して作成します。

スタンドアロン・ファイル・ロガーを Windows サービスとして実行して、Windows セッションからログオフするときにもファイル・ロガーが継続して実行するようにできますし、システムが再始動したときに自動的に開始されるように構成できます。手順については、[107 ページの『MFT スタンドアロン・ファイル・ロガーのインストール』](#)を参照してください。

z/OS と IBM i 上では、スタンドアロン・ファイル・ロガーはサポートされていません。

スタンドアロン・データベース・ロガー

スタンドアロンのデータベース・ロガーは、キュー・マネージャーおよびデータベースをホストするシステム上にインストールする、Java アプリケーションです。スタンドアロン・データベース・ロガーは多くの場合、調整キュー・マネージャーと同じシステムにインストールされますが、調整キュー・マネージャーに接続されているキュー・マネージャーと同じシステムにインストールすることもできます。スタンドアロン・データベース・ロガーは、IBM MQ バインディングを使用して、関連するキュー・マネージャーに接続し、タイプ 2 またはタイプ 4 の JDBC ドライバーを使用して、Db2 または Oracle データベースに接続

します。スタンドアロン・データベース・ロガーはキュー・マネージャーのXAサポートを使用して、キュー・マネージャーとデータベース両方のグローバル・トランザクションを調整してデータを保護するため、これらのタイプの接続が必要になります。

Windows システムを使用している場合、スタンドアロン・ロガーを Windows サービスとして実行でき、Windows セッションからログオフするときにもロガーを継続して実行することができます。手順については、スタンドアロン・データベース・ロガーのトピック [114 ページの『MFT スタンドアロン・データベース・ロガーのインストール』](#)を参照してください。

Java EE データベース・ロガー

Java EE データベース・ロガーは、アプリケーション・サーバーにインストールするための EAR ファイルとして用意されています。データベースロガーは他のエンタープライズアプリケーションと一緒に管理できるため、既存の Java EE アプリケーションサーバー環境を利用できる場合は、スタンドアロンの Java EE データベースロガーを使用するよりも便利です。さらに、IBM MQ のサーバーとデータベースをホストするシステムとは別のシステムに Java EE データベース・ロガーをインストールすることも可能です。Java EE データベース・ロガーは Db2 および Oracle データベースでの使用がサポートされています。WebSphere Application Server 7.0 にインストールされている場合、Java EE データベース・ロガーは Oracle Real Application Clusters もサポートします。

ロガーの構成方法については、以下のトピックを参照してください。

- [107 ページの『MFT スタンドアロン・ファイル・ロガーのインストール』](#)
- [114 ページの『MFT スタンドアロン・データベース・ロガーのインストール』](#)
- [128 ページの『MFT 用の Java EE データベース・ロガーのインストール』](#)

MFT スタンドアロン・ファイル・ロガーのインストール

スタンドアロン・ファイル・ロガーは、IBM MQ バインディングを使用して調整キュー・マネージャーに接続する必要のある Java プロセスです。スタンドアロン・ファイル・ロガーを定義するには、**fteCreateLogger** コマンドを使用し、このトピックの手順に従ってください。

このタスクについて

スタンドアロン・ファイル・ロガーについて詳しくは、[106 ページの『MFT ロガーの構成』](#)を参照してください。このトピックの手順では、調整キュー・マネージャーに接続するようにロガーを構成します。ロガーの代替構成については、[127 ページの『MFT スタンドアロン・ロガーの代替構成』](#)を参照してください。

z/OS と IBM i 上では、スタンドアロン・ファイル・ロガーはサポートされていません。

手順

1. Managed File Transfer Logger コンポーネントがインストールされていることを確認します。詳しくは、[Managed File Transfer 製品のオプション](#)を参照してください。
2. 調整キュー・マネージャーを指定し **-loggerType** パラメーターを FILE に設定して **fteCreateLogger** コマンドを実行し、スタンドアロン・ファイル・ロガーを作成します。詳しくは、[408 ページの『fteCreateLogger \(MFT ファイルまたはデータベース・ロガーの作成\)』](#)を参照してください。
3. オプション: カスタム・フォーマットを使用する場合は、**fteCreateLogger** コマンドで作成された XML ファイルを変更できます。ログ・フォーマット定義は FileLoggerFormat.xml ファイルに配置されます。詳しくは、[108 ページの『MFT スタンドアロン・ファイル・ロガー・フォーマット』](#)を参照してください。
4. **fteCreateLogger** コマンドによって提供される MQSC コマンドを調整キュー・マネージャーに対して実行して、ロガー・キューを作成します。
5. ロガー・プロセスを実行するユーザーを識別し、そのユーザーの権限を構成します。詳しくは、[113 ページの『MFT スタンドアロン・ファイル・ロガーのユーザー・アクセス権限の構成』](#)を参照してください。

6. オプション: **fteCreateLogger** コマンドを実行したときに作成される `logger.properties` ファイルを編集すると、スタンドアロン・ファイル・ロガーをさらに構成できます。このファイルは、キーと値のペアで構成されている Java プロパティ・ファイルです。 `logger.properties` ファイルは、 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` ディレクトリにあります。使用可能なプロパティとそれらの効果については、118 ページの『MFT ロガーの構成プロパティ』を参照してください。
7. **Windows**
 オプション: Windows システムを使用している場合には、スタンドアロン・ファイル・ロガーを Windows サービスとして実行できます。 **-s** パラメーターを指定して、 **fteModifyLogger** コマンドを実行します。詳しくは、492 ページの『fteModifyLogger (Windows サービスとしての MFT ロガーの実行)』を参照してください。
8. **fteStartLogger** コマンドを使用してスタンドアロン・ファイル・ロガーを開始します。詳しくは、530 ページの『fteStartLogger』を参照してください。
 前のステップを実行し、Windows で **-s** パラメーターを指定して **fteModifyLogger** コマンドを使用した場合、スタンドアロン・ファイル・ロガーは、Windows サービスとして開始されます。
9. ロガー出力を確認します。スタンドアロン・ファイル・ロガーによって 2 タイプの出力 (ファイル転送監査データおよびロガー診断データ) が生成されます。ファイル転送監査データは `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs` 内にあります。ロガー診断データは `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name` 内にあります。
10. **fteStopLogger** コマンドを使用すると、ロガーを停止できます。詳しくは、533 ページの『fteStopLogger』を参照してください。

タスクの結果

MFT スタンドアロン・ファイル・ロガー・フォーマット

ファイル・ロガーにより書き込まれるメッセージ情報のフォーマットは、 `FileLoggerFormat.xml` ファイルで定義できます。

ロガーの構成ディレクトリは、 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` にあります。新規ファイル・ロガーを作成すると、そのファイル・ロガーによって使用されるデフォルトの定義セットが含まれたバージョンのファイルが作成されます。デフォルトのログ・フォーマット定義については、612 ページの『MFT スタンドアロン・ファイル・ロガーのデフォルトのログ・フォーマット』を参照してください。

独自のカスタム・ログ・フォーマットを指定する場合、 `FileLoggerFormat.xml` ファイルを編集します。

カスタム・ログ・フォーマット定義

ログ・フォーマット定義には、メッセージ・タイプのセットと各メッセージ・タイプのフォーマット定義が含まれています。メッセージ・タイプのフォーマット定義は、XPath フォーマットで指定されている挿入項目と各挿入項目を分離する分離文字で構成されます。挿入項目の配列によって、ログ・ファイルに出力される行中の内容の配列が決まります。例えば、 `callStarted` メッセージ・タイプの定義は次のようになります。

```
<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>
```

このフォーマットでは、ログ・ファイルの行が次のように作成されます。

```
2011-11-25T10:53:04;414d5120514d5f67627468696e6b20206466cf4e20004f02;[CSTR];
AGENT1;AGENT_QM;Managed Call;executable;echo;call test;
```

フォーマット定義では、各情報がログ・ファイルの行中に出現する順序で挿入項目が指定されます。[FileLoggerFormat.xml](#) ファイルのフォーマットを定義する XML スキーマについて詳しくは、[616 ページの『スタンドアロン・ファイル・ロガーのフォーマット XSD』](#)を参照してください。

メッセージ・タイプ

FTE エージェントは、複数の異なるメッセージ・タイプを SYSTEM.FTE/Log サブトピックに書き込みます。詳細内容は [を参照してください](#)。ログ・ファイル定義には、次のようなメッセージ・タイプのフォーマット定義を含めることができます。

```
callCompleted
callStarted
monitorAction
monitorCreate
monitorFired
notAuthorized
scheduleDelete
scheduleExpire
scheduleSkipped
scheduleSubmitInfo
scheduleSubmitTransfer
scheduleSubmitTransferSet
transferStarted
transferCancelled
transferComplete
transferDelete
transferProgress
```

メッセージのフォーマットはそれぞれ異なるものにすることができます。大部分のメッセージ・タイプでは、SYSTEM.FTE/Log サブトピックからコンシュームされるログ・メッセージごとに 1 行がログ・ファイルに書き込まれます。この場合、ログ・フォーマット定義内の XPATH アドレスは、メッセージのルートからの相対アドレスで指定する単純なものになります。この方式で出力が書き込まれるのは、次のメッセージ・タイプの場合です。

```
callCompleted
callStarted
monitorAction
monitorCreate
monitorFired
notAuthorized
scheduleDelete
scheduleExpire
scheduleSkipped
scheduleSubmitInfo
scheduleSubmitTransfer
transferStarted
transferCancelled
transferComplete
transferDelete
```

ログ・メッセージの書き込みに使用される他の方式では、ログ・メッセージ内の 1 つの転送セットの項目を表すのに、複数の行を使用します。この場合、指定されるフォーマットは、ログ・メッセージ内の転送セットの各項目に適用されます。転送セット内の各項目に固有の情報を組み込むには、その項目を XPATH のルートとして XPATH を指定する必要があります。この方式で出力が書き込まれるのは、次のメッセージ・タイプの場合です。

```
scheduleSubmitTransferSet
transferProgress
```

転送セットの項目ごとに出力が1行書き込まれます。転送セット内のすべての項目で固定にする情報については、ログ・メッセージのルートからの相対 XPATH アドレスを使用することができます。次の `transferProgress` フォーマット定義の簡単な例では、タイム・スタンプと転送 ID が固定されます。項目をルートとして相対で指定される情報の内容は、書き込まれる行によって異なります。この例では、項目ごとにソースと宛先のファイル情報が書き込まれます。

```
<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>
```

この例では、次のように、このフォーマットで1行以上のログ・ファイル・エントリーが作成されます。

```
2011-11-25T13:45:16;414d5120514d5f67627468696e6b20206466cf4e20033702;[TPR0];0
;/src/test1.file;3575;file;leave ;/dest/test1.file;3575;file;overwrite;;
2011-11-25T13:45:16;414d5120514d5f67627468696e6b20206466cf4e20033702;[TPR0];0
;/src/test2.file;3575;file;leave ;/dest/test2.file;3575;file;overwrite;;
```

挿入のフォーマット

メッセージ・タイプのフォーマットを定義する際に使用可能な挿入タイプは、`user` と `system` の2つがあります。挿入タイプは、insert エレメントの `type` 属性で定義します。どちらの挿入タイプも、insert エレメントの `width` 属性および `ignoreNull` 属性を使用してレイアウトをカスタマイズすることができます。以下に例を示します。

```
<insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
```

この例の挿入では、ログ・メッセージの `/transaction/@ID` の場所にある情報が、48文字になるように切り取りまたは埋め込みの処理が行われた後にログに書き込まれます。`/transaction/@ID` の内容がヌルの場合、`ignoreNull` 属性が `false` に設定されているため、48文字になるように埋め込みが行われてからヌル・ストリングが書き込まれます。`ignoreNull` が `true` に設定されている場合には、48文字になるように埋め込まれた空ストリングが書き込まれます。`width="0"` を設定すると、列の幅がトリムされないことを意味します。幅が0にトリムされることではない。この方法で `ignoreNull` 属性を使用して、予期しないときに、NULL が検出された場合にログを検出することができます。これは、新しいログ・ファイル定義のデバッグを行う際に役立つ場合があります。

ユーザー定義の挿入

ユーザー定義の挿入には、その挿入で書き込まれる情報の XPATH アドレスが含まれます。このアドレスは、FTE ログ・メッセージ内の情報の一部を参照します。ログ・メッセージのフォーマットについて詳しくは、以下を参照してください。

- [748 ページの『ファイル転送ログ・メッセージ・フォーマット』](#)
- [769 ページの『スケジュール済みファイル転送ログ・メッセージ・フォーマット』](#)
- [774 ページの『MFT モニター・ログ・メッセージ・フォーマット』](#)

システム定義の挿入

システム定義の挿入には、ログ・メッセージ内にない情報、または XPATH 言語では定義するのが容易でない情報のいずれかの情報の一部を参照するキーワードが含まれます。

システム定義の挿入では、以下のものがサポートされています。

- `type` - ログ・メッセージのタイプを、短形式で書き込みます。
- `callArguments` - 管理対象呼び出しに提供する引数のセットを、スペースで分離したフォーマットで書き込みます。
- `transferMetaData` - 転送に定義されたメタデータ・エントリーのセットを、`key=value` をコンマで区切った形式で記述します。

次の表では、メッセージ・タイプごとに、システム定義の挿入の「`type`」値をリストします。

メッセージ・タイプ	「 <code>type</code> 」システム挿入の値
<code>callCompleted</code>	[CCOM]
<code>callStarted</code>	[CSTR]
<code>monitorAction</code>	[MACT]
<code>monitorCreate</code>	[MCRT]
<code>monitorFired</code>	[MFIR]
<code>notAuthorized</code>	[AUTH]
<code>scheduleDelete</code>	[SDEL]
<code>scheduleExpire</code>	[SEXP]
<code>scheduleSkipped</code>	[SSKP]
<code>scheduleSubmitInfo</code>	[SSIN]
<code>scheduleSubmitTransfer</code>	[SSTR]
<code>scheduleSubmitTransferSet</code>	[SSTS]
<code>transferStarted</code>	[TSTR]
<code>transferCancelled</code>	[TCAN]

表 15. サポートされるメッセージ・タイプとそれらの「type」システム挿入の要約 (続き)

メッセージ・タイプ	「type」システム挿入の値
transferComplete	[TCOM]
transferDelete	[TDEL]
transferProgress	[TPRO]

MFT スタンドアロン・ファイル・ロガーからのメッセージ・タイプの除外
 特定のメッセージ・タイプをファイル・ロガーの出力から除外するために、空のメッセージ・タイプ・エレメントを使用できます。

例

例えば、次のフォーマット定義では、ファイル・ロガーから transferProgress メッセージが出力されないようにします。

```
<?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>
```

MFT スタンドアロン・ファイル・ロガーのカスタム・フォーマットの定義

ログ・フォーマット定義内でカスタム・メッセージ・タイプのサブセットを定義して、ログ・ファイル・フォーマットのカスタマイズに必要な構成を減らすことができます。

このタスクについて

FileLoggerFormat.xml ファイルに messageTypes エレメントが含まれていない場合、そのメッセージ・タイプのフォーマットにはデフォルトのフォーマットが使用されます。デフォルトと異なる場合にのみフォーマットを指定してください。

例

この例では、フォーマット定義によって、transferStarted メッセージ・タイプのデフォルトのフォーマットが、転送を開始したユーザーのみを出力するこの削減されたバージョンに置き換えられます。他のすべてのメッセージ・タイプは、このログ・フォーマット定義には含まれていないため、デフォルトのフォーマットを使用します。

```
<?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>
```

関連資料

612 ページの『MFT スタンドアロン・ファイル・ロガーのデフォルトのログ・フォーマット』
Managed File Transfer スタンドアロン・ファイル・ロガーのデフォルトのログ・ファイル・フォーマット
定義です。

616 ページの『スタンドアロン・ファイル・ロガーのフォーマット XSD』
スタンドアロン・ファイル・フォーマットのスキーマです。

MFT スタンドアロン・ファイル・ロガーの重複メッセージの削減

スタンドアロン・ファイル・ロガーのログでは、重複ログ・メッセージが発生する場合があります。
logger.properties ファイルを使用することによって、スタンドアロン・ファイル・ロガーを調整し
て、重複の数を削減できます。

ファイル・ロガー・ログの重複メッセージ

障害が発生した場合、IBM MQ にコミットされている SYSTEM.FTE/Log# トピックからログ・メッセージを
コンシュームせずに、スタンドアロン・ファイル・ロガーのログにログ・メッセージが書き込まれる場合
があります。これが発生した場合、スタンドアロン・ファイル・ロガーが再開するときに同じメッセ
ージが再度取得され、それがログ・ファイルに再度書き込まれます。ログ・ファイルを手動で見
る場合でも自動的に処理する場合でも、このような重複が発生したときの対処法を計画してくだ
さい。重複の検出を支援する機能として、スタンドアロン・ファイル・ロガーが開始するときに
以下のメッセージがログ・ファイルに出力されます。

```
BFGDB0054I: The file logger has successfully started
```

前のインスタンスが失敗する前に読み取られた最終メッセージはスタンドアロン・ファイル・ロ
ガーの開始時に処理されるため、重複が発生するとすれば常にそのときです。新規インスタ
ンスがいつ開始したかが分かれば、重複発生の可能性とそれらに対処する必要があるかど
うかを予測できます。

重複の数の削減

スタンドアロン・ファイル・ロガーは、パフォーマンス向上のため、処理するログ・メ
ッセージをグループ化して各トランザクションにまとめます。このバッチ・サイズは、障
害時に発生する可能性のある重複メッセージの最大数になります。重複の数を削減す
るため、logger.properties ファイル内の以下のプロパティを調整できます。

```
wmqfte.max.transaction.messages
```

例えば、この値を 1 に設定すると、重複メッセージの最大数が 1 に削減されます。この
値を変更すると、スタンドアロン・ファイル・ロガーのパフォーマンスに影響を与える
ので、システムに悪影響を及ぼさないように徹底的なテストを行う必要があることに注
意してください。

logger.properties ファイルは、MQ_DATA_PATH/mqft/config/coordination_qmgr_name/
loggers/logger_name ディレクトリーにあります。使用可能なプロパティとそれらの影
響について詳しくは、118 ページの『MFT ロガーの構成プロパティ』を参照してくだ
さい。

MFT スタンドアロン・ファイル・ロガーのユーザー・アクセス権限の構成

テスト環境では、通常のユーザー・アカウントに必要な、任意の特権を新たに追加
できます。実稼働環境では、ジョブを実行するために必要な最小限の許可を備
えた新しいユーザーを作成することをお勧めします。

このタスクについて

スタンドアロン・ファイル・ロガーと IBM MQ を単一のシステム上にインストールする
必要があります。次のようにして、ユーザーの許可を構成します。

手順

1. Managed File Transfer のインストールの一部としてインストールされたファイルの読み取り許可、および必要に応じて実行許可が、ユーザーに確実に付与されているようにしてください。
2. 構成ディレクトリー内の logs ディレクトリーでのファイルの作成許可および書き込み許可が、ユーザーに確実に付与されているようにしてください。このディレクトリーは、イベント・ログで使用され、必要に応じて診断トレース・ファイルおよび初期障害データ・キャプチャー機能 (FFDC) ファイルでも使用されます。
3. ユーザーが、独自のグループに所属していて、かつ調整キュー・マネージャーの広範囲の許可を備えたグループに所属していないことを確実にしてください。ユーザーは、mqm グループに所属してはなりません。特定のプラットフォームでは、staff グループに、キュー・マネージャーアクセス権限も自動的に付与されます。スタンドアロン・ファイル・ロガー・ユーザーは staff グループに入れなくてください。IBM MQ Explorer を使用すると、キュー・マネージャー自体およびキュー・マネージャー内のオブジェクトの権限レコードを表示できます。オブジェクトを右クリックして、「**オブジェクト権限**」>「**権限レコードの管理**」をクリックします。コマンド行では、コマンド `dspmqaout` (権限の表示) または `dmpmqaut` (権限のダンプ) を使用できます。
4. IBM MQ Explorer の「**権限レコードの管理**」ウィンドウまたは `setmqaut` (権限の付与または取り消し) コマンドを使用して、ユーザー自身のグループの権限を追加します (UNIX の場合、IBM MQ 権限はグループにのみ関連付けられ、個々のユーザーには関連付けられません)。必要な権限は以下のとおりです。
 - キュー・マネージャーに対する CONNECT と INQUIRE (IBM MQ Java ライブラリーを操作するには、INQUIRE 権限が必要です)。
 - SYSTEM.FTE トピックに対する SUBSCRIBE 権限。
 - SYSTEM.FTE.LOG.RJCT.logger_name キューに対する PUT 権限。
 - SYSTEM.FTE.LOG.CMD.logger_name キューに対する GET 権限。リジェクト・キュー名とコマンド・キュー名は、デフォルトの名前です。スタンドアロン・ファイル・ロガー・キューの構成時に別のキュー名を選択した場合は、それらのキュー名に対する権限を追加してください。

MFT スタンドアロン・データベース・ロガーのインストール

以下の手順に従って、スタンドアロン・データベース・ロガーをインストールし、構成します。

このタスクについて

重要: Managed File Transfer ロガーは IBM i プラットフォームではサポートされません。

スタンドアロン・データベース・ロガーについて詳しくは、106 ページの『[MFT ロガーの構成](#)』を参照してください。

注: データベース内の同じスキーマに対して一度に複数のデータベース・ロガー (スタンドアロンまたは Java EE) を実行することはできません。実行しようとする、転送ログ・データをデータベースに書き込もうとしたときに、競合が起きます。

手順

1. 使用するデータベースの資料に従って、データベース・ソフトウェアをインストールします。
使用するデータベースで、JDBC サポートがオプションのコンポーネントである場合には、ユーザーがこのコンポーネントをインストールする必要があります。
2. **-loggerType** パラメーターを DATABASE に設定して `fteCreateLogger` コマンドを実行し、スタンドアロン・データベース・ロガーを作成します。詳細については 408 ページの『[fteCreateLogger \(MFT ファイルまたはデータベース・ロガーの作成\)](#)』を参照してください。
デフォルトのスキーマ名は FTELOG です。FTELOG 以外のスキーマ名を使用する場合は、用意されている SQL ファイル (使用中のデータベースに応じて `ftelog_tables_db2.sql` または `ftelog_tables_oracle.sql`) を編集し、そのスキーマ名を反映してから、次の手順に進む必要があ

ります。詳しくは、[118 ページの『MFT ロガーの構成プロパティー』](#)の `wmqfte.database.schema` を参照してください。

3. データベースのツールを使用して、必要なデータベース表を作成します。

Multi マルチプラットフォームでは、表を作成するために実行できる SQL コマンドが `ftelog_tables_db2.sql` ファイルと `ftelog_tables_oracle.sql` ファイルに入っています。

z/OS z/OS では、実行する必要があるファイルは、使用している Db2 for z/OS のバージョンによって異なります。

- Db2 for z/OS 9.0 以前の場合は、ファイル `ftelog_tables_zos.sql` を実行して表を作成します。このファイルは、転送されるファイルのサイズと各転送に関連付けられた表 ID を示すフィールドに INTEGER データ型を使用して表を作成します。
- Db2 for z/OS 9.1 以降の場合は、ファイル `ftelog_tables_zos_bigint.sql` を実行して表を作成します。このファイルは、転送されるファイルのサイズと各転送に関連付けられた表 ID を示すフィールドに BIGINT データ型を使用して表を作成します。

4. **fteCreateLogger** コマンドによって提供される MQSC コマンドをロガー・コマンド・キュー・マネージャーに対して実行し、ロガー・キューを作成します。スタンドアロン・データベース・ロガーは、調整キュー・マネージャー上の 2 つのキューを使用します。1 つ目のキューは、スタンドアロン・データベース・ロガーの操作を制御するメッセージが置かれるコマンド・キューです。このコマンド・キューのデフォルトの名前は、`SYSTEM.FTE.LOG.CMD.logger_name` です。2 つ目のキューはリジェクト・キューです。スタンドアロン・データベース・ロガーはログ・メッセージを廃棄するということがないため、処理できないメッセージを検出すると、調査（および可能な場合は再処理）のために、そのメッセージをリジェクト・キューに置きます。キュー・マネージャーの送達不能キューをこの目的で使用することはお勧めできません。これは、リジェクトされたメッセージには DLH ヘッダーが含まれていないため、およびリジェクトされたメッセージは、他の理由で送達不能キューに書き込まれたメッセージと組み合わせるべきではないためです。リジェクト・キューのデフォルトの名前は、`SYSTEM.FTE.LOG.RJCT.logger_name` です。これらの 2 つのキューは、**fteCreateLogger** コマンドによって生成される MQSC スクリプト・ファイルで定義されます。

5. ユーザーの選択と許可の構成

6. オプション: ステップ [114 ページの『2』](#) で **fteCreateLogger** コマンドを実行したときに作成された `logger.properties` ファイルを編集すると、スタンドアロン・データベース・ロガーをさらに構成できます。このファイルは、キーと値のペアで構成されている Java プロパティー・ファイルです。`logger.properties` ファイルは、`MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` ディレクトリーにあります。使用可能なプロパティーとそれらの効果の詳細については、[『118 ページの『MFT ロガーの構成プロパティー』](#)を参照してください。

7. **Windows**

オプション: Windows システムを使用している場合には、スタンドアロン・データベース・ロガーを Windows サービスとして実行できます。-s パラメーターを指定して、**fteModifyLogger** コマンドを実行します。詳細については [492 ページの『fteModifyLogger \(Windows サービスとしての MFT ロガーの実行\)』](#)を参照してください。

8. オプション: 使用しているデータベースが Oracle である場合、または Db2 データベースにリモート接続している場合は、ロガーがデータベース・サーバーへの認証に使用するユーザー名とパスワードを指定する必要があります。このユーザー名とパスワードは、`MQMFTCredentials.xsd` スキーマに定義されている形式に従った資格情報ファイルに指定されています。詳細については [810 ページの『MFT 資格情報ファイル・フォーマット』](#)を参照してください。資格情報ファイルを作成したら、`logger.properties` ファイルで `wmqfte.database.credentials.file` プロパティーを使用して、資格情報ファイルの場所を指定する必要があります。
9. **fteStartLogger** コマンドを使用して、スタンドアロン・データベース・ロガーを開始します。デフォルトでは、スタンドアロン・データベース・ロガーはバックグラウンドで稼働し、スタンドアロン・データベース・ロガーは `logs` ディレクトリーにあるファイルに出力を書き込みます。スタンドアロン・データベース・ロガーをフォアグラウンドで実行して、ログ・ファイルに加えてコンソールにも出力を生成する場合には、**fteStartLogger** コマンドに **-F** パラメーターを追加します。

前のステップを実行し、Windows で **-s** パラメーターを指定して **fteModifyLogger** コマンドを使用した場合、スタンドアロン・データベース・ロガーは、Windows サービスとして開始されます。

リモート・データベースでの MFT の使用

Managed File Transfer のロガーを使用して、リモート・システムのデータベースと通信できます。

このタスクについて

Managed File Transfer のインストール先のマシンとは別のマシンにデータベースがインストールされている場合は、以下の手順を実行します。それぞれの手順は、特に注記がない限り、Db2 と Oracle の両方に当てはまります。

手順

1. Managed File Transfer のインストール先のシステムにデータベース・クライアントをインストールします。
2. ローカル・データベース・クライアント構成にリモート・データベース・サーバーを追加します。Managed File Transfer と IBM MQ からデータベースに正しくアクセスするために、この構成を更新する必要があります。
3. `logger.properties` ファイルに新しいプロパティを指定することにより、資格情報ファイル **wmqfte.database.credentials.file** を使用してデータベースに接続します。

注：以前のバージョンの Managed File Transfer では、プロパティとして **wmqfte.oracle.user** または **wmqfte.database.user**、および **wmqfte.oracle.password** または **wmqfte.database.password** を使用していました。それらのプロパティは、非推奨になっています。代わりに、**wmqfte.database.credentials.file** を使用してください。

4. **Oracle のみ**：データベースへのリモート接続を許可するために、調整キュー・マネージャーの `qm.ini` ファイルの `XAResourceManager` スタンザを以下のように変更します (それぞれご使用の情報に合わせて、データベース名、ユーザー名、およびユーザー・パスワードを変更してください)。
`Oracle_XA+Acc=P/ftelog/
qgw783jhT+SesTm=35+DB=FTEAUDIT1+SqlNet=FTEAUDIT1+threads=false`、変更は太字で強調表示されます。
5. **Oracle のみ**：`logger.properties` ファイルで、**wmqfte.oracle.host** プロパティと **wmqfte.oracle.port** プロパティを使用してホストとポートを指定します。ホストとポートのデフォルト値でもローカル・データベース・クライアントを操作できるので、これまで操作していたのがローカル・データベースであれば、これらの値を設定していなかった可能性があります。

関連資料

118 ページの『MFT ロガーの構成プロパティ』

Managed File Transfer ロガーには、一連の構成プロパティがあります。`logger.properties` ファイルでこれらのプロパティを指定します。このファイルは、`MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` ディレクトリーにあります。

MFT スタンドアロン・データベース・ロガーのユーザー・アクセス権限の構成

テスト環境では、通常のユーザー・アカウントに必要な、任意の特権を新たに追加できます。実稼働環境では、ジョブを実行するために必要な最小限の許可を備えた新しいユーザーを作成することをお勧めします。

このタスクについて

スタンドアロン・データベース・ロガーを実行するために必要なユーザー・アカウントの数とタイプは、使用するシステムの数によって異なります。スタンドアロン・データベース・ロガー、IBM MQ、およびデータベースを1つのシステムにインストールすることも、2つのシステムにインストールすることも可能です。スタンドアロン・データベース・ロガーは IBM MQ と同じシステム上になければなりません。それらのコンポーネントをインストールするためのトポロジーを以下に示します。

スタンドアロン・データベース・ロガー、IBM MQ、およびデータベースをすべて同じシステムに配置する場合

3つのコンポーネントすべてで使用する1つのオペレーティング・システム・ユーザーを定義できます。これは、スタンドアロン・データベース・ロガーに適した構成です。スタンドアロン・データベース・ロガーは、バインディング・モードを使用してIBM MQに接続し、ネイティブ接続を使用してデータベースに接続します。

スタンドアロン・データベース・ロガーとIBM MQを1つのシステムに配置し、データベースを別のシステムに配置する場合

この構成では、2つのユーザーを作成します。つまり、スタンドアロン・データベース・ロガーを実行するシステムのオペレーティング・システム・ユーザーと、データベース・サーバーのデータベースに対するリモート・アクセスを実行するオペレーティング・システム・ユーザーです。これは、リモート・データベースを使用するスタンドアロン・データベース・ロガーに適した構成です。スタンドアロン・データベース・ロガーは、バインディング・モードを使用してIBM MQに接続し、クライアント接続を使用してデータベースにアクセスします。

以下の説明では、例としてftelogというユーザーを想定しますが、任意のユーザー名を使用できます。次のようにして、ユーザーの許可を構成します。

手順

1. Managed File Transfer Remote Tools and Documentation のインストールの一部としてインストールされたファイルの読み取り許可、および必要に応じて実行許可が、ユーザーに確実に付与されているようにしてください。
2. 構成ディレクトリー内のlogs ディレクトリーにある任意のファイルの作成許可および書き込み許可が、ユーザーに確実に付与されているようにしてください。このディレクトリーは、イベント・ログで使用され、必要に応じて診断トレース・ファイルおよびFFDC ファイルでも使用されます。
3. ユーザーが、独自のグループに所属していて、かつ調整キュー・マネージャーの広範囲の許可を備えたグループに所属していないことを確実にしてください。ユーザーは、mqm グループに所属してはなりません。特定のプラットフォームでは、staff グループに、キュー・マネージャーアクセス権限も自動的に付与されます。スタンドアロン・データベース・ロガー・ユーザーは staff グループに入れないでください。IBM MQ Explorer を使用すると、キュー・マネージャー自体およびキュー・マネージャー内のオブジェクトの権限レコードを表示できます。オブジェクトを右クリックして、「**オブジェクト権限**」>「**権限レコードの管理**」をクリックします。コマンド行では、コマンド `dspmqaout` (権限の表示) または `dmpmqaut` (権限のダンプ) を使用できます。
4. IBM MQ Explorer の「**権限レコードの管理**」ウィンドウまたは `setmqaut` (権限の付与または取り消し) コマンドを使用して、ユーザー自身のグループの権限を追加します (UNIX の場合、IBM MQ 権限はグループにのみ関連付けられ、個々のユーザーには関連付けられません)。必要な権限は以下のとおりです。
 - キュー・マネージャーに対する CONNECT と INQUIRE (IBM MQ Java ライブラリーを操作するには、INQUIRE 権限が必要です)。
 - SYSTEM.FTE トピックに対する SUBSCRIBE 権限。
 - SYSTEM.FTE.LOG.RJCT.logger_name キューに対する PUT 権限。
 - SYSTEM.FTE.LOG.CMD.logger_name キューに対する GET 権限。リジェクト・キュー名とコマンド・キュー名は、デフォルトの名前です。スタンドアロン・データベース・ロガー・キューの構成時に別のキュー名を選択した場合は、それらのキュー名に対する権限を追加してください。
5. 使用中のデータベースに該当する固有のユーザー構成を実行します。
 - データベースが Db2 の場合は、以下の手順を実行します。

Db2 でデータベース・ユーザーを管理するメカニズムはいくつかあります。ここでの説明は、オペレーティング・システム・ユーザーに基づいたデフォルト・スキームに適用されます。

 - ftelog ユーザーが Db2 管理グループ (例: db2iadm1、db2fadm1、dasadm1) に所属していない状態にしてください

- このユーザーに、データベースに接続する許可、および『[ステップ 2: 必要なデータベース表の作成](#)』で作成した表に対する select、insert、および update の許可を付与します
- データベースが Oracle の場合は、以下の手順を実行します。
 - fteelog ユーザーが Oracle 管理グループ (例: Windows での ora_dba や UNIX での dba) に所属していない状態にしてください
 - このユーザーに、データベースに接続する許可、および『[ステップ 2: 必要なデータベース表の作成](#)』で作成した表に対する select、insert、および update の許可を付与します

MFT ロガーの構成プロパティ

Managed File Transfer ロガーには、一連の構成プロパティがあります。logger.properties ファイルでこれらのプロパティを指定します。このファイルは、MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name ディレクトリーにあります。

IBM WebSphere MQ 7.5 以降の場合は、ファイルまたはディレクトリーの場所を表す Managed File Transfer の一部のプロパティで環境変数を使用できます。これにより、製品の一部の実行時に使用されるファイルまたはディレクトリーの場所を、環境の変更 (プロセスを実行しているユーザーなど) に合わせて変更することができます。詳しくは、[535 ページ](#)の『[MFT プロパティでの環境変数の使用](#)』を参照してください。

注: Windows 上でファイル・パスを指定する場合、円記号 (¥) 分離文字は、二重の円記号 (¥¥) (つまり、エスケープ・シーケンスの ¥) で表す必要があります。代わりに、分離文字として単一スラッシュ文字 (/) を使用できます。Oracle の Java プロパティ・ファイルでの文字のエスケープ・シーケンスについて詳しくは、[Javadoc for the Properties class](#) を参照してください。

バインディング・モード接続のプロパティ

プロパティ名	説明	デフォルト値
wmqfte.logger.type	使用しているロガー・タイプ (file または database)。この値は FILE または DATABASE に設定してください。	デフォルト値なし
wmqfte.max.transaction.messages	トランザクションがコミットされる前にトランザクションで処理されるメッセージの最大数。循環ロギング・モードでは、未完了データに使用可能な固定スペース量はキュー・マネージャーが持っています。使用可能なスペースを使い尽くさないよう十分に低い値をこのプロパティに設定するようにしてください。	50
wmqfte.max.transaction.time	トランザクションのコミット間に経過する時間の最大長 (ミリ秒)。	5000
wmqfte.max.consecutive.reject	連続して (つまり、有効メッセージが出現することなく) リジェクト可能なメッセージの最大数。 この数値を超えると、ロガーは問題がメッセージ自体にあるのではなく構成にあると見なします。例えば、使用しているすべてのエージェント名よりデータベース内のエージェント名列を狭く作ると、エージェントを参照しているすべてのメッセージがリジェクトされます。	50
wmqfte.reject.queue.name	ロガーがメッセージを処理できない場合にロガーがメッセージを置くキューの名前。データベース・ロガーを使用している場合、このキューに置かれる可能性のあるメッセージについて詳しくは、 データベース・ロガーのエラー処理とリジェクト を参照してください。	SYSTEM.FTE.LOG.RJCT.logger_name
wmqfte.command.queue.name	ロガーがその動作を制御するコマンド・メッセージを読み取る際に、その読み取り元となるキューの名前。	SYSTEM.FTE.LOG.CMD.logger_name

表 16. バインディング・モードの接続プロパティ (続き)		
プロパティ名	説明	デフォルト値
wmqfte.queue.manager	<p>ローガの接続先キュー・マネージャー。このパラメーターは必須で、キュー・マネージャーへのバインディング・モード接続に必要となるものはこのパラメーターのみです。(リモート・キュー・マネージャーに接続するためのプロパティについては、125 ページの表 17を参照してください。)</p>	デフォルト値なし
wmqfte.message.source.type	<p>次のいずれかの値。</p> <p>automatic subscription デフォルト値。ローガは、SYSTEM.FTE/Log/# で定義される独自の永続的な管理対象サブスクリプションをキュー・マネージャー上に作成して使用します。これは、ほとんどのシナリオで適切な値となります。</p> <p>administrative subscription 自動サブスクリプションが適切でない場合は、(例えば、IBM MQ Explorer、MQSC、または PCF を使用して) 異なるサブスクリプションを定義し、ローガにそのサブスクリプションを使用するよう命令できます。例えば、この値を使用してログ・スペースを分割し、1 番目のローガは A から H のエージェントを、2 番目のローガは I から P のエージェントを、3 番目のローガは Q から Z のエージェントを処理するようにします。</p> <p>queue IBM MQ トポロジーを使用するとローガのサブスクリプションを作成することが不都合であることがわかった場合は、代わりにキューを使用できます。IBM MQ を構成して、通常は調整キュー・マネージャー上の SYSTEM.FTE/Log/# へのサブスクリプションによって受信されるメッセージをキューが受信するようにします。</p>	automatic subscription
wmqfte.message.source.name	<p>メッセージ送信元タイプが administrative subscription または queue である場合に使用するサブスクリプションまたはキューの名前。このプロパティは、送信元タイプが automatic subscription である場合は無視されます。</p>	デフォルト値なし
wmqfte.database.credentials.file	<p>データベースに接続するためのユーザー名とパスワードを格納するファイル。</p> <p>IBM WebSphere MQ 7.5 以降の場合は、このプロパティの値に環境変数を含めることができます。</p> <p>詳しくは、810 ページの『MFT 資格情報ファイル・フォーマット』を参照してください。</p>	<p> 認証資格情報ファイルの作成に関する詳細は、MFT 資格情報ファイルの作成を参照してください</p> <p> このファイルの場所およびアクセス権については、MQMFTCredentials.xml の構成を参照してください。</p> <p> 認証資格情報ファイルの作成について詳しくは、MFT と IBM MQ の接続認証を参照してください。</p>

表 16. バインディング・モードの接続プロパティ (続き)

プロパティ名	説明	デフォルト値
<p>wmqfte.database.driver</p>	<p>データベースの JDBC ドライバー・クラスの場合。これは通常、JAR ファイルのパスおよびファイル名です。例えば、AIX® システム上の Db2 用の Type 2 ドライバーは、ファイル /opt/IBM/db2/V9.5/java/db2jcc.jar を必要とします。Windows システムでは、パス分離文字をスラッシュ文字 (/) として指定します (例: C:/Program Files/IBM/SQLLIB/java/db2jcc.jar)。</p> <p>z/OS の場合は、db2jcc.jar ファイルの絶対パスを指定します。例えば、wmqfte.database.driver=/db2/db2v10/jdbc/classes/db2jcc.jar などです。</p> <p>z/OS システムの場合、以下のすべての JAR ファイルを参照する必要があります。</p> <ul style="list-style-type: none"> • db2jcc.jar • db2jcc_license_cisuz.jar • db2jcc_javax.jar <p>データベース・ドライバーが複数の JAR ファイルで構成されている場合 (例えば、Db2 9.1 の場合は、ドライバー JAR ファイルとライセンス JAR ファイルが必要)、これらの JAR ファイルをすべてこのプロパティに含めます。複数のファイル名は、ご使用のプラットフォームのクラスパス分離文字、つまり、Windows システムの場合はセミコロン文字 (;)、他のプラットフォームの場合はコロン文字 (:) を使用して分離します。</p>	<p>デフォルト値なし</p>
<p>wmqfte.database.exclude_duplicate_metadata</p>	<p>データベース・ロガー・スキーマ内の他の表にある情報を含むメタデータ表にエントリを保管するかどうかを制御します。この値は true または false に設定してください。これらのメタデータ・エントリは、既存のデータと重複しており、データベース・ストレージ容量を無駄に占有することになるため、デフォルトでは保管されません。同じデータが出現するプロパティ・エントリおよび表は以下のとおりです。</p> <ul style="list-style-type: none"> • com.ibm.wmqfte.SourceAgent TRANSFER_EVENT または CALL_REQUEST • com.ibm.wmqfte.DestinationAgent TRANSFER_EVENT • com.ibm.wmqfte.MqmdUser TRANSFER_EVENT または CALL_REQUEST • com.ibm.wmqfte.OriginatingUser TRANSFER_EVENT または CALL_REQUEST • com.ibm.wmqfte.OriginatingHost TRANSFER_EVENT または CALL_REQUEST • com.ibm.wmqfte.TransferId TRANSFER または CALL_REQUEST • com.ibm.wmqfte.JobName TRANSFER または CALL_REQUEST <p>このプロパティの値を false に設定すると、これらのメタデータ・エントリがメタデータ表に保管されます。</p>	<p>true</p>

表 16. バインディング・モードの接続プロパティ (続き)

プロパティ名	説明	デフォルト値
wmqfte.database.host	<p>Db2 のみ:</p> <p>IBM WebSphere MQ 7.5 以降では、Type 4 JDBC ドライバーを使用して接続するデータベース・サーバーのホスト名。このプロパティの値を指定する場合は、<code>wmqfte.database.port</code> の値も指定する必要があります。両方のプロパティが定義されていない場合は、データベース・ロガーがデフォルトの Type 2 JDBC ドライバーを使用して接続されます。</p> <p>このプロパティに値を指定する場合は、データベースに接続するユーザー名とパスワードを定義するために、このロガー用の資格情報ファイル (<code>wmqfte.database.credentials.file</code> プロパティでファイル・パスを定義) が存在しており、アクセス可能である必要があります。資格情報ファイルは、データベースがローカル・システム上にある場合でも必要です。</p>	デフォルト値なし
wmqfte.database.name	Managed File Transfer ログ表を含むデータベース・インスタンス (または、Db2 for z/OS を使用する場合はサブシステム) の名前。	デフォルト値なし
wmqfte.database.type	使用中のデータベース管理システム。Db2 または Oracle。この値を <code>db2</code> または <code>oracle</code> に設定してください。	db2
wmqfte.database.port	<p>Db2 のみ:</p> <p>IBM WebSphere MQ 7.5 以降では、Type 4 JDBC ドライバーを使用して接続するデータベース・サーバーのポート番号。このプロパティの値を指定する場合は、<code>wmqfte.database.host</code> の値も指定する必要があります。両方のプロパティが定義されていない場合は、データベース・ロガーがデフォルトの Type 2 JDBC ドライバーを使用して接続されます。</p> <p>このプロパティに値を指定する場合は、データベースに接続するユーザー名とパスワードを定義するために、このロガー用の資格情報ファイル (<code>wmqfte.database.credentials.file</code> プロパティでファイル・パスを定義) が存在しており、アクセス可能である必要があります。資格情報ファイルは、データベースがローカル・システム上にある場合でも必要です。</p>	デフォルト値なし
wmqfte.database.schema	Managed File Transfer ログイン表を含むデータベース・スキーマ。ほとんどの場合、デフォルト値が適切ですが、サイト固有のデータベースに関する考慮事項に応じて代替値を指定する必要がある場合があります。	FTELOG
wmqfte.database.native.library.path	<p>選択したデータベース・ドライバーが必要とするネイティブ・ライブラリーを含むパス (該当する場合)。例えば、AIX システム上の Db2 用の Type 2 ドライバーは、<code>/opt/IBM/db2/V9.5/lib32/</code> からのライブラリーを必要とします。このプロパティの代わりに、他の方法を使用して <code>java.library.path</code> システム・プロパティを設定できます。</p> <p>Solaris および HP-UX システムの場合、fteStartLogger コマンドを実行する前に、パスを含めるように <code>LD_LIBRARY_PATH</code> 環境変数を設定し、エクスポートする必要もあります。</p>	デフォルト値なし
wmqfte.file.logger.fileDirectory	ファイル・ロガーのログ・ファイルを配置するディレクトリー。	<code>mqft/logs/coordination_dir/loggers/logger_name/logs</code>

表 16. バインディング・モードの接続プロパティ (続き)

プロパティ名	説明	デフォルト値
wmqfte.file.logger.fileSize	<p>ログ・ファイルが大きくなることが許可される最大サイズ。サイズの値は、ゼロより大きい正整数で、単位 KB、MB、GB、m (分)、h (時)、d (日)、w (週) のいずれか 1 つを後に付けます。例えば、<code>wmqfte.file.logger.fileSize=5MB</code> のように指定します。最大ファイル・サイズは 5MB です。<code>wmqfte.file.logger.fileSize=2d</code> ファイルの最大サイズを 2 日分のデータに指定します。</p>	10MB
wmqfte.file.logger.fileCount	<p>作成するログ・ファイルの最大数。データ量がこの数のファイルに保管できる最大量を超えた場合、最も古いファイルが削除されます。そのため、ファイル数が指定された値より大きくなることはありません。</p>	3

表 16. バインディング・モードの接続プロパティー (続き)

プロパティー名	説明	デフォルト値
wmqfte.file.logger.mode	<p>使用しているローガー・モード (circular または linear)。この値は CIRCULAR または LINEAR に設定してください。</p> <p>CIRCULAR - ファイル・ローガーは、wmqfte.file.logger.fileSize プロパティーを使用して定義される最大サイズにファイルが到達するまで、そのファイルに情報を書き込みます。最大サイズに到達すると、ファイル・ローガーは新規ファイルを開始します。このモードで書き込まれるファイルの最大数は、wmqfte.file.logger.fileCount プロパティーを使用して定義される値によって制御されます。このファイル最大数に到達すると、ファイル・ローガーは最初のファイルを削除して再作成し、それを現行のアクティブ・ファイルとして使用します。wmqfte.file.logger.fileSize プロパティーで定義された値が固定サイズ・バイト単位 (例えば、KB、MB、GB) である場合、このモードで使用されるディスク・スペースの上限は、fileSize に fileCount を乗算した値です。wmqfte.file.logger.fileSize プロパティーで定義された値が時間単位 (例えば、m、h、d、w) である場合、最大サイズは、ご使用のシステムにおけるその時間内のログ・メッセージのスループットによって決まります。このモードで実行される場合に使用されるログ・ファイル命名規則は、logger_namenumber-timestamp.log です。それぞれの説明は以下のとおりです。</p> <ul style="list-style-type: none"> • logger_name は、fteCreateLogger コマンドでローガーに付けられた名前です。 • number は、セット内のファイルの番号です。 • timestamp はは、ファイルが作成されたときのタイム・スタンプです。 <p>LOGGER1-20111216123430147.log に例を示します</p> <p>LINEAR - ファイル・ローガーは、wmqfte.file.logger.fileSize プロパティーを使用して定義される最大サイズにファイルが到達するまで、そのファイルに情報を書き込みます。最大サイズに到達すると、ファイル・ローガーは新規ファイルを開始します。以前に書き込まれたファイルは削除されないため、それらをログ・メッセージの履歴レコードとして保持できます。linear モードで実行している場合、ファイルは削除されず、作成可能なファイル数の上限もないため、wmqfte.file.logger.fileCount プロパティーは無視されます。このモードで実行している場合には上限がないため、ディスク・スペースが少なくならないように、ログ・ファイルによって使用されているディスク・スペースの量を追跡する必要があります。このモードで実行される場合に使用されるログ・ファイル命名規則は、logger_name-timestamp.log です。それぞれの説明は以下のとおりです。</p> <ul style="list-style-type: none"> • logger_name は、fteCreateLogger コマンドでローガーに付けられた名前です。 • timestamp はは、ファイルが作成されたときのタイム・スタンプです。 <p>LOGGER-20111216123430147.log に例を示します</p>	デフォルト値なし

表 16. バインディング・モードの接続プロパティ (続き)

プロパティ名	説明	デフォルト値
wmqfte.max.retry.interval	<p>ロガーが永続エラーを検出した場合の、再試行間の最大時間 (ミリ秒)。</p> <p>エラー条件によっては (例えば、データベース接続が失われた場合)、ロガーが続行できなくなります。このタイプの条件が発生した場合、ロガーは現行トランザクションをロールバックし、ある一定の時間待機した後、再試行します。ロガーの待機時間は最初非常に短いため、一時的なエラーであればすぐに克服できます。しかし、ロガーの再試行のたびに、待機時間は増大します。これにより、エラー条件がより長く続く場合、例えば、データベースが保守のためのサービス停止になっている場合に、あまりに多くの不要な作業が発生しないようになっています。</p> <p>このプロパティを使用して待機時間の長さに制限を設けると、エラー条件が解決される妥当な時間で再試行が発生します。</p>	600
loggerQMgrRetryInterval	ロガーのプロセス・コントローラーがキュー・マネージャーの使用可否を検査する間隔 (秒単位)。	30
maxRestartCount	maxRestartInterval プロパティの値によって指定された時間間隔内に行える再開の最大回数。この値を超えると、ロガーのプロセス・コントローラーはロガーの再開を中止し、その代わりに maxRestartDelay プロパティの値に基づいたアクションを実行します。	4
maxRestartInterval	ロガーのプロセス・コントローラーがロガーの再開数を測定する間隔 (秒単位)。この間隔内に行われた再開の回数が maxRestartCount プロパティの値を超えると、ロガーのプロセス・コントローラーはそのロガーの再開を中止します。ロガーのプロセス・コントローラーは、その代わりに maxRestartDelay プロパティの値に基づいたアクションを実行します。	120
maxRestartDelay	ロガーの再開率が maxRestartCount プロパティおよび maxRestartInterval プロパティの値を超過した場合にロガーのプロセス・コントローラーがとる動作を決定します。ゼロ以下の値を指定すると、ロガーのプロセス・コントローラーは停止します。ゼロより大きな値を指定した場合、この秒数だけ待機した後、ロガーのプロセス・コントローラーが保持する再開履歴情報がリセットされ、ロガーが再開されます。	-1
wmqfte.oracle.port	ロガーが Oracle インスタンスへの接続に使用するポート。このポートを TNS リスナーとも言います。	1521
wmqfte.oracle.host	ロガーが Oracle インスタンスへの接続に使用するホスト。	localhost
armELEMTYPE	オプションのプロパティ。ロガーを自動リスタート・マネージャー (ARM) で再始動するように構成する場合は、関連付けられた ARM ポリシー内で指定された ARM ELEMTYPE パラメーター値をこのプロパティに設定します。ロガーの場合、ELEMTYPE は SYSBFGLG に設定します。	設定なし
armELEMENT	オプションのプロパティ。ロガーを自動リスタート・マネージャー (ARM) で再始動するように構成する場合は、関連付けられた ARM ポリシー内で指定された ARM ELEMENT パラメーター値をこのプロパティに設定します。ELEMENT 値には、ロガー名に対応する値を設定できます。	設定なし

プロパティ名	説明	デフォルト値
loggerQMGrAuthenticationCredentialsFile	ロガーの調整キュー・マネージャーに接続するためのMQ 接続資格情報が入ったファイルへのパス。	<p>Z/OS 認証資格情報ファイルの作成に関する詳細は、MFT 資格情報ファイルの作成を参照してください</p> <p>ULW このファイルの場所およびアクセス権については、MQMFTCredentials.xml の構成を参照してください。</p> <p>ULW 認証資格情報ファイルの作成について詳しくは、MFT と IBM MQ の接続認証を参照してください。</p>
トレース	<p>オプションのプロパティ。ロガーの開始時にトレースを有効にしてロガーを実行するトレース仕様。トレース仕様は、クラス、等号、およびトレース・レベルのコンマ区切りのリストです。</p> <p>例えば、<code>com.ibm.wmqfte.databaselogger</code>、<code>com.ibm.wmqfte.databaselogger.operation=all</code> です。</p> <p>コロン区切りのリストで、複数のトレース仕様を指定できます。例えば、<code>com.ibm.wmqfte.databaselogger=moderate:com.ibm.wmqfte.databaselogger.operation=all</code></p>	なし
traceFiles	オプションのプロパティ。保持するトレース・ファイルの合計数。この値は、ロガーのプロセス・コントローラーに加え、ロガー自体にも適用されます。	5
traceSize	オプションのプロパティ。トレースが次のファイルで折り返す前の、各トレース・ファイルの最大サイズ (MB 単位)。この値は、ロガーのプロセス・コントローラーとロガー自体に適用されます。	20

クライアント・モード接続のプロパティ

V 9.0.4

次に表に、ロガー・キュー・マネージャーへのクライアント・モード接続をサポートするために必要な追加のロガー・プロパティを示します。

プロパティ名	説明	デフォルト値
wmqfte.queue.manager.host	ロガー・キュー・マネージャーのホスト名または IP アドレス。	デフォルト値なし
wmqfte.queue.manager.port	ロガー・キュー・マネージャーが listen するポート。	1414
wmqfte.queue.manager.channel	ロガー・キュー・マネージャーのサーバー接続チャンネルの名前。	SYSTEM.DEF.SVRCONN
wmqfte.Ssl.CipherSuite	<p>ロガーとロガー・キュー・マネージャーのデータ交換方法に関する TLS の動作を指定します。</p> <p>wmqfte.Ssl.CipherSuite の値は、CipherSuite 名です。この CipherSuite 名は、ロガー・キュー・マネージャー・チャンネルで使用される CipherSpec 名にマップされます。</p> <p>詳しくは、CipherSuite と CipherSpec の名前のマッピングを参照してください。</p>	デフォルト値なし

表 17. クライアント・モードの接続プロパティ (続き)		
プロパティ名	説明	デフォルト値
wmqfte.Ssl.PeerName	ロガー・キュー・マネージャーによって提供される名前と合致しなければならない識別名の骨組みを指定します。その識別名に基づいて、接続時にキュー・マネージャーによって提示される識別用証明書が検査されます。	デフォルト値なし
wmqfte.Ssl.TrustStore	ロガーが信頼する証明書の場所を指定します。 wmqfte.Ssl.TrustStore の値は、ファイル・パスです。ファイル・パスが Windows のファイル・パスの場合、円記号 (¥) はさらに円記号を追加してエスケープ (¥¥) する必要があります。このプロパティの値には環境変数を含めることができる点にご注意ください。	デフォルト値なし
wmqfte.Ssl.TrustStoreCredentialsFile	wmqfte.Ssl.TrustStore 資格情報が含まれているファイルのパス。 このプロパティの値には環境変数を含めることができる点にご注意ください。	デフォルト値なし
wmqfte.Ssl.TrustStoreType	使用する SSL 鍵ストアのタイプ。JKS および PKCS#12 の鍵ストアがサポートされています。このプロパティの値は、jks または pkcs12 のいずれかになります。	jks
wmqfte.Ssl.KeyStore	ロガーの秘密鍵の場所を指定します。 wmqfte.Ssl.KeyStore の値はファイル・パスです。ファイル・パスが Windows のファイル・パスの場合、円記号 (¥) はさらに円記号を追加してエスケープ (¥¥) する必要があります。このプロパティの値には環境変数を含めることができる点にご注意ください。	デフォルト値なし
wmqfte.Ssl.KeyStore.CredentialsFile	wmqfte.Ssl.KeyStore 資格情報が含まれているファイルのパス。 このプロパティの値には環境変数を含めることができる点にご注意ください。	デフォルト値なし
wmqfte.Ssl.KeyStoreType	使用する SSL 鍵ストアのタイプ。JKS および PKCS#12 の鍵ストアがサポートされています。このプロパティの値は、jks または pkcs12 のいずれかになります。	jks
wmqfte.Ssl.FipsRequired	ロガーのレベルで FIPS サポートを使用可能にすることを指定します。このプロパティの値は、true または false のいずれかになります。詳しくは、667 ページの『MFT の FIPS サポート』を参照してください。	false

関連資料

535 ページの『MFT プロパティでの環境変数の使用』

IBM WebSphere MQ 7.5 以降、ファイルまたはディレクトリーの場所を表す Managed File Transfer プロパティで環境変数を使用できます。これにより、製品の一部の実行時に使用されるファイルまたはディレクトリーの場所を、環境の変更に合わせて変えることができます。例えば、プロセスを実行するユーザーの変更などです。

605 ページの『MFT の SSL プロパティ』

SSL または TLS を IBM MQ および Managed File Transfer とともに使用して、エージェントとキュー・マネージャーとの間の許可されない接続を防止し、エージェントとキュー・マネージャーとの間のメッセージ・トラフィックを暗号化します。

MFT スタンドアロン・ロガーの代替構成

通常、Managed File Transfer スタンドアロン・ロガーは、ファイルかデータベース・タイプかにかかわらず、調整キュー・マネージャーと同じシステム上にあり、IBM MQ バインディング・モードで調整キュー・マネージャーに接続されます。ただし、調整キュー・マネージャーに接続されているキュー・マネージャーと同じシステムにインストールすることもできます。スタンドアロン・ロガーは、そのスタンドアロン・ロガーが自動的に作成するサブスクリプションを使用して、メッセージを受信します。これは、インストール指示に記載されている構成です。

しかし、サイト固有の考慮事項がある場合は、`wmqfte.message.source.type` プロパティによって制御される別の2つの方法でメッセージを受信するようにスタンドアロン・ロガーを構成できます。このプロパティについては、『[データベース・ロガー・プロパティ](#)』で説明されています。

管理サブスクリプション

デフォルトでは、スタンドアロン・ロガーは、デフォルトの永続サブスクリプション・オプションおよび管理対象サブスクリプションを使用して、SYSTEM.FTE/Log/# トピックへの独自のサブスクリプションを作成します(つまり、キュー・マネージャーは、アプリケーションにメッセージが渡されるまでそれらの保持に使用するバッキング・キューを制御します)。サブスクリプションまたはキューで他のオプションが必要になった場合は、代わりに自分でサブスクリプションを作成して必要なオプションを設定し、そのサブスクリプションを使用するようスタンドアロン・ロガーを構成できます。作成したサブスクリプションを使用するために、スタンドアロン・ロガーの許可を追加することを忘れないでください。

この構成を使用する例として、2つのワイルドカード・サブスクリプションを使用することによるログ・スペースの分割が挙げられます。FINANCE で始まる名前を持つエージェントから一方のデータベースにログを送信し、ACCOUNTING で始まるエージェントから他方のデータベースにログを送信します。このタイプの構成では2つのスタンドアロン・ロガー・インスタンスが必要となります。それぞれのインスタンスは、必要なサブスクリプションと独自のコマンド・キューおよびリジェクト・キューを参照する独自の `logger.properties` ファイルを持っています。

名前が ACCOUNTING で始まるエージェントからのみログ・メッセージを収集するには、SYSTEM.FTE/Log/ACCOUNTING* というトピック・ストリングで調整キュー・マネージャー上にサブスクリプション・オブジェクトを作成します。「ワイルドカード使用」値を「文字レベル・ワイルドカード」に設定します。さらに、ご使用のロガーの `logger.properties` ファイルに項目を追加する必要もあります。例えば、上記の設定で ACCOUNTING.LOGS というサブスクリプション・オブジェクトを作成した場合、`logger.properties` ファイルに次の項目を追加します。

```
wmqfte.message.source.type=administrative subscription
wmqfte.message.source.name=ACCOUNTING.LOGS
```

スタンドアロン・ロガーは、トピック・ストリングが SYSTEM.FTE/Log/ でのみ始まるログ・メッセージを扱います。制限がより多いトピック・ストリングを指定することはできますが、制限がより少ないストリングを指定することはできません。制限がより少ないストリングを間違えて指定した場合、SYSTEM.FTE/Log/ 以外のトピック・ストリングに関連するすべてのパブリケーションはリジェクト・キューに移動し、スタンドアロン・ロガーはエラー・メッセージ BFGDB0002E を生成します。このエラー・メッセージは、スタンドアロン・ロガー構成に問題があることを暗黙に示します。

キュー

典型的なトポロジーでは、スタンドアロン・ロガーは調整キュー・マネージャーと同じシステム上で実行されます。これが不可能な場合は、サブスクリプション宛先として別のキュー・マネージャー上のキューを使用して調整キュー・マネージャー上にサブスクリプションを作成します(リモート・キュー定義を使用するか、サブスクリプションの DESTQMGR プロパティを使用して行います)。そうすれば、2番目のキュー・マネージャーをホストしているシステムでロガーを実行して、キューからメッセージを読み取ることができるようになります。トランザクションの保全性を保証するには、スタンドアロン・ロガーは常にバインディング・モードでそのキュー・マネージャーと接続している必要があります。リジェクト・キューとコマンド・キューは、スタンドアロン・ロガーの接続先となる同じキュー・マネージャー上で定義する必要があります。キュー・マネージャーは、IBM WebSphere MQ 7.5 以降でなければなりません。

例えば、サブスクリプションによってキュー USER.QUEUE に置かれたログ・メッセージを収集するには、以下のエントリーを `logger.properties` ファイルに追加します。

```
wmqfte.message.source.type=queue  
wmqfte.message.source.name=USER.QUEUE
```

MFT 用の Java EE データベース・ロガーのインストール

以下の手順を実行して、Managed File Transfer 用の Java EE データベース・ロガーをインストールして構成します。

このタスクについて

Java EE データベース・ロガーについて詳しくは、[106 ページの『MFT ロガーの構成』](#)のトピックを参照してください。

注：Java EE データベース・ロガーとスタンドアロン・ロガーは、それぞれデータベースの別個のインスタンスを使用している場合を除き、同時に実行することができません。

手順

1. Java EE データベース・ロガーをインストールする前に、環境を準備する必要があります。[129 ページの『MFT 用の Java EE データベース・ロガーをインストールするための準備』](#)のトピックの指示に従ってください。
2. Java EE データベース・ロガーを Java Platform, Enterprise Edition (Java EE) 準拠のアプリケーション・サーバーにインストールします。手順については、以下のトピックを参照してください。
 - [132 ページの『WebSphere Application Server 7.0 を使用した MFT 用 Java EE データベース・ロガーのインストール』](#)
 - [136 ページの『MFT with WebSphere Application Server Community Edition 用の Java EE データベース・ロガーのインストール』](#)

関連タスク

[129 ページの『MFT 用の Java EE データベース・ロガーをインストールするための準備』](#)

以下の手順を実行して、Java EE データベース・ロガーをインストールする前に Managed File Transfer 環境を準備します。

[132 ページの『WebSphere Application Server 7.0 を使用した MFT 用 Java EE データベース・ロガーのインストール』](#)

WebSphere Application Server 7.0 を使用して Managed File Transfer 用の Java Platform, Enterprise Edition (Java EE) データベース・ロガーをインストールおよび構成するには、以下の手順に従います。

[136 ページの『MFT with WebSphere Application Server Community Edition 用の Java EE データベース・ロガーのインストール』](#)

以下の手順を実行して、Managed File Transfer 用の Java Platform, Enterprise Edition (Java EE) データベース・ロガーをインストールし、構成します (WebSphere Application Server Community Edition)。

[140 ページの『MFT 用の Java EE データベース・ロガーのユーザー・アクセスの構成』](#)

Managed File Transfer Java Platform, Enterprise Edition (Java EE) データベース・ロガーを構成する場合、IBM MQ、データベース、およびオペレーティング・システムにアクセスするためのユーザー・アカウントが必要です。必要なオペレーティング・システム・ユーザーの数は、それらのコンポーネントのホストとして使用するシステムの数によって異なります。

[142 ページの『スタンドアロン・データベース・ロガーから MFT 用の Java EE データベース・ロガーへのマイグレーション』](#)

スタンドアロン・データベース・ロガーから Java EE データベース・ロガーにマイグレーションできます。スタンドアロン・データベース・ロガーを停止して、Java EE データベース・ロガーをインストールする必要があります。ログ項目の消失または重複を回避するには、スタンドアロン・データベース・ロガーを停止する前に、SYSTEM.FTE トピックへのメッセージのパブリッシュを停止し、JEE データベース・ロガーを

インストールした後にこれを再始動する必要があります。マイグレーション前にデータベースをバックアップしてください。

関連資料

362 ページの『MFT ロガーの権限』

ロガーを実行するオペレーティング・システム・ユーザーには、ロガー・キューと SYSTEM.FTE トピックに対する特定の IBM MQ 権限が必要です。

MFT 用の Java EE データベース・ロガーをインストールするための準備

以下の手順を実行して、Java EE データベース・ロガーをインストールする前に Managed File Transfer 環境を準備します。

このタスクについて

Java EE データベース・ロガーについて詳しくは、106 ページの『MFT ロガーの構成』のトピックを参照してください。

手順

1. 使用するデータベースの資料に従って、データベース・ソフトウェアをインストールします。
使用するデータベースで、JDBC サポートがオプションのコンポーネントである場合には、ユーザーがこのコンポーネントをインストールする必要があります。
2. データベースに用意されているツールを使用してデータベースを作成します。データベースのテーブル・スペース・サイズとバッファ・プール・ページ・サイズは少なくとも 8K である必要があります。デフォルトのスキーマ名は FTELOG です。FTELOG 以外のスキーマ名を使用する場合は、用意されている SQL ファイル (使用中のデータベースに応じて `ftelog_tables_db2.sql` または `ftelog_tables_oracle.sql`) を編集し、そのスキーマ名を反映してから、次の手順に進む必要があります。
3. データベースのツールを使用して、必要なデータベース表を作成します。

Multi マルチプラットフォームでは、表を作成するために実行できる SQL コマンドが `ftelog_tables_db2.sql` ファイルと `ftelog_tables_oracle.sql` ファイルに入っています。

z/OS z/OS では、実行する必要があるファイルは、使用している Db2 for z/OS のバージョンによって異なります。

- Db2 for z/OS 9.0 以前の場合は、ファイル `ftelog_tables_zos.sql` を実行して表を作成します。このファイルは、転送されるファイルのサイズと各転送に関連付けられた表 ID を示すフィールドに INTEGER データ型を使用して表を作成します。
 - Db2 for z/OS 9.1 以降の場合は、ファイル `ftelog_tables_zos_bigint.sql` を実行して表を作成します。このファイルは、転送されるファイルのサイズと各転送に関連付けられた表 ID を示すフィールドに BIGINT データ型を使用して表を作成します。
4. FTELOG 以外のスキーマ名に変更した場合は、EAR ファイルでスキーマ名を変更する必要があります。詳細内容は [を参照してください](#)。
 5. IBM MQ でリジェクト・キューを作成します。
ロガーはログ・メッセージを廃棄するということがないため、処理できないメッセージを検出すると、調査 (および可能な場合は再処理) のために、そのメッセージをリジェクト・キューに置きます。キュー・マネージャの送達不能キューをこの目的で使用しないでください。これは、リジェクトされたメッセージには DLH ヘッダーが含まれていないため、およびリジェクトされたメッセージは、他の理由で送達不能キューに書き込まれたメッセージと組み合わせるはならないためです。
fteCreateLogger コマンドは、リジェクト・キューを作成します。このリジェクト・キューのデフォルトの名前は、SYSTEM.FTE.LOG.RJCT.`logger_name` です。
 6. [JEE ロガーのユーザー・アクセス権限の構成](#)のトピックにある手順を実行します。

次のタスク

これで、Java EE 準拠のアプリケーション・サーバーに Java EE データベース・ロガーをインストールできます。ご使用のアプリケーション・サーバーに基づいて、以下のトピックの指示に従ってください。

- [132 ページの『WebSphere Application Server 7.0 を使用した MFT 用 Java EE データベース・ロガーのインストール』](#)
- [136 ページの『MFT with WebSphere Application Server Community Edition 用の Java EE データベース・ロガーのインストール』](#)

MFT 用の Java EE データベース・ロガーで使用するスキーマ名の変更

Java Platform, Enterprise Edition (Java EE) データベース・ロガーでは、デフォルト以外のスキーマ名を持ったデータベースを使用できます。Java EE データベース・ロガーの EAR ファイルでスキーマ名を変更する必要があります。

このタスクについて

Java EE データベース・ロガーで使用するスキーマ名を変更するには、以下の手順を実行します。

手順

1. 以下のコマンドを使用して、その EAR ファイルから JPA の JAR ファイルを抽出します。

```
jar -xvf ear_file lib/jpa_file
```

ここで、

- *ear_file* は、Db2 を使用しているか Oracle を使用しているかに応じて、`com.ibm.wmqfte.databaselogger.jee.oracle.ear` または `com.ibm.wmqfte.databaselogger.jee.ear` のいずれかになります。
 - *jpa_file* は、Db2 を使用しているか Oracle を使用しているかに応じて、`com.ibm.wmqfte.web.jpa.oracle.jar` または `com.ibm.wmqfte.web.jpa.jar` のいずれかになります。
2. 以下のコマンドを使用して、JPA の JAR ファイルから `persistence.xml` ファイルを抽出します。

```
jar -xvf lib/jpa_file META_INF/persistence.xml
```

ここで、

- *jpa_file* は、Db2 を使用しているか Oracle を使用しているかに応じて、`com.ibm.wmqfte.web.jpa.oracle.jar` または `com.ibm.wmqfte.web.jpa.jar` のいずれかになります。
3. `persistence.xml` ファイルを編集して、以下の行を変更します。

```
<property name="openjpa.jdbc.Schema" value="schema_name"/>
```

説明:

- *schema_name* は、使用するスキーマ名です。

4. 以下のコマンドを使用して、変更後の `persistence.xml` ファイルで JPA の JAR を更新します。

```
jar -uvf lib/jpa_file META_INF/persistence.xml
```

ここで、

- *jpa_file* は、Db2 を使用しているか Oracle を使用しているかに応じて、`com.ibm.wmqfte.web.jpa.oracle.jar` または `com.ibm.wmqfte.web.jpa.jar` のいずれかになります。

5. 以下のコマンドを使用して、変更後の JPA の JAR ファイルで EAR ファイルを更新します。

```
jar -uvf ear_file lib/jpa_file
```

ここで、

- `ear_file` は、Db2 を使用しているか Oracle を使用しているかに応じて、`com.ibm.wmqfte.databaselogger.jee.oracle.ear` または `com.ibm.wmqfte.databaselogger.jee.ear` のいずれかになります。
- `jpa_file` は、Db2 を使用しているか Oracle を使用しているかに応じて、`com.ibm.wmqfte.web.jpa.oracle.jar` または `com.ibm.wmqfte.web.jpa.jar` のいずれかになります。

次のタスク

変更後の EAR ファイルを使用して、Java EE データベース・ロガーをインストールします。

関連タスク

[132 ページの『WebSphere Application Server 7.0 を使用した MFT 用 Java EE データベース・ロガーのインストール』](#)

WebSphere Application Server 7.0 を使用して Managed File Transfer 用の Java Platform, Enterprise Edition (Java EE) データベース・ロガーをインストールおよび構成するには、以下の手順に従います。

[136 ページの『MFT with WebSphere Application Server Community Edition 用の Java EE データベース・ロガーのインストール』](#)

以下の手順を実行して、Managed File Transfer 用の Java Platform, Enterprise Edition (Java EE) データベース・ロガーをインストールし、構成します (WebSphere Application Server Community Edition)。

WebSphere Application Server 7.0 でのネイティブ・ライブラリー・パスの設定

Java Platform, Enterprise Edition (Java EE) データベース・ロガー・アプリケーションを WebSphere Application Server 7.0 にデプロイし、アプリケーションと IBM MQ の間でバインディング・モード接続を使用する場合は、システム上の IBM MQ ネイティブ・ライブラリーの場所を使用して IBM MQ メッセージング・プロバイダーを構成する必要があります。

このタスクについて

アプリケーション・サーバーにネイティブ・ライブラリー・パスを設定しない場合、WebSphere Application Server 7.0 のシステム・アウト・ログで以下のエラー・メッセージを受け取る可能性があります。

```
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]
```

WebSphere Application Server 7.0 管理コンソールを使用して、以下のステップを実行します。

手順

1. ナビゲーション・ペインで、「リソース」 > 「JMS」 > 「JMS プロバイダー」を展開します。
2. 接続ファクトリーまたはバインディング・モード接続を作成するアクティベーション・スペック用に、適切な有効範囲の IBM MQ メッセージング・プロバイダーを選択します。
注: Server 有効範囲のネイティブ・パス情報は、より高い有効範囲のネイティブ・パス情報よりも優先して使用され、Node 有効範囲のネイティブ・パス情報は Cell 有効範囲のネイティブ・パス情報よりも優先して使用されます。
3. 「一般プロパティー」の下で、「ネイティブ・ライブラリー・パス」フィールドに、IBM MQ ネイティブ・ライブラリーが含まれるディレクトリーの絶対パス名を入力します。
例えば、Linux では、`/opt/mqm/java/lib` と入力します。ディレクトリー名を 1 つのみ入力します。
4. 「OK」をクリックします。
5. アプリケーション・サーバーを再始動して、構成をリフレッシュします。
6. 必須: アプリケーション・サーバーをもう一度再始動して、ライブラリーをロードします。

WebSphere Application Server 7.0 を使用した MFT 用 Java EE データベース・ロガーのインストール

WebSphere Application Server 7.0 を使用して Managed File Transfer 用の Java Platform, Enterprise Edition (Java EE) データベース・ロガーをインストールおよび構成するには、以下の手順に従います。

始める前に

JEE データベース・ロガー・アプリケーションをインストールする前に、[129 ページの『MFT 用の Java EE データベース・ロガーをインストールするための準備』](#) および [131 ページの『WebSphere Application Server 7.0 でのネイティブ・ライブラリー・パスの設定』](#) のトピックに記載された手順に従ってください。

このタスクについて

Java EE データベース・ロガーについて詳しくは、[106 ページの『MFT ロガーの構成』](#) を参照してください。

手順

1. XA JDBC プロバイダーをセットアップします。
 - a) WebSphere Application Server 7.0 管理コンソールのナビゲーションから、「リソース」 > **JDBC** > **JDBC プロバイダー** を選択します。
 - b) 「新規作成」をクリックし、コンソール・ウィザードを使用して JDBC プロバイダーを作成します。
 - c) ウィザードのステップ 1 で、「データベース・タイプ」リストから使用中のデータベースを選択し、「プロバイダー・タイプ」リストから関連するプロバイダー・タイプを選択します。「実装タイプ」リストから、「XA データ・ソース」を選択します。「次へ」をクリックします。
 - d) ウィザードのステップ 2 で、必要なデータベース JAR ファイルのディレクトリー・ロケーションが正しく設定されていることを確認します。「次へ」をクリックします。
 - e) 要約ページで「終了」をクリックして、JDBC プロバイダーを作成します。
2. 認証別名を作成します。データ・ソースの別名 1 つと IBM MQ の別名 1 つを作成します。
 - a) WebSphere Application Server 7.0 管理コンソールのナビゲーションから「セキュリティ」 > 「グローバル・セキュリティ」を選択します。
 - b) 「認証」の見出しの下で、「Java 認証・承認サービス (JAAS)」を展開します。
 - c) 「J2C 認証データ」をクリックします。認証別名のページが開きます。
 - d) データ・ソースの認証別名を作成します。
 - i) 「新規」をクリックします。
 - ii) 「別名」、「ユーザー ID」、「パスワード」、および「説明」の詳細情報を入力します。「ユーザー ID」フィールドと「パスワード」フィールドに入力する詳細情報は、データベース・ユーザーを作成した時に入力した詳細情報と一致していなければなりません。詳しくは、[JEE データベース・ロガーのユーザー・アクセス権限の構成](#) を参照してください。
 - iii) 「OK」をクリックします。
 - e) IBM MQ の認証別名を作成します。
 - i) 「新規」をクリックします。
 - ii) 「別名」、「ユーザー ID」、「パスワード」、および「説明」の詳細情報を入力します。「ユーザー ID」フィールドと「パスワード」フィールドで入力する詳細情報は、IBM MQ のインストール済み環境のユーザーとパスワード設定に一致していなければなりません。
 - iii) 「OK」をクリックします。
3. 次のようにしてデータ・ソースを作成します。
 - a) WebSphere Application Server 7.0 管理コンソール・ナビゲーションから 「リソース」 > **JDBC** > 「データ・ソース」を選択します。
 - b) 「有効範囲」 ドロップダウン・リストを選択し、有効範囲を適切な値に変更します。例えば、Node=yourNode, Server=yourServer です。

- c) 「**新規作成**」をクリックし、コンソール・ウィザードを使用してデータ・ソースを作成します。
 - d) ウィザードのステップ 1 で、「**データ・ソース名**」フィールドに `wmqfte-database` と入力し、「**JNDI 名**」フィールドに `jdbc/wmqfte-database` と入力します。「**次へ**」をクリックします。
 - e) ウィザードのステップ 2 で、「**既存 JDBC プロバイダーを選択**」ドロップダウン・リストを使用して、前のステップで作成した JDBC プロバイダーを選択します。「**次へ**」をクリックします。
 - f) **Db2**: ウィザードのステップ 3 で、「**ドライバ・タイプ**」フィールドに 4 と入力します。
 - g) **Db2**: 「**データベース名**」、「**サーバー名**」、および「**ポート番号**」フィールドに詳細を入力して、「**次へ**」をクリックします。

Oracle: 「**URL**」フィールドに接続 URL を入力し、「**データ・ストアのヘルパー・クラス名**」フィールドでデータ・ストアの正しいヘルパーを選択します。

Oracle RAC: Oracle Real Application Cluster に接続する場合、接続 URL には、データベースの使用可能なすべてのインスタンスに接続するために必要なホスト情報が組み込まれている必要があります。
 - h) ウィザードのステップ 4 で、ステップ 2d で定義したデータ・ソースの認証別名の名前を「**XA リカバリーの認証別名**」リストから選択します。「**コンポーネント管理認証別名**」リストと「**コンテナ管理認証別名**」リストから同じ名前を選択してください。
 - i) 要約ページで「**終了**」をクリックして、データ・ソースを作成します。
4. オプション: データ・ソースの構成を検証します。
- a) WebSphere Application Server 7.0 管理コンソール・ナビゲーションから「**リソース**」 > **JDBC** > 「**データ・ソース**」を選択します。
 - b) 「**テスト接続**」ボタンをクリックします。
5. トピックを作成します。
- a) WebSphere Application Server 7.0 の管理コンソールのナビゲーションから、「**リソース**」 > 「**JMS**」 > 「**トピック**」をクリックします。
 - b) 「**有効範囲**」ドロップダウン・リストを選択し、有効範囲を適切な値に変更します。例えば、`Node=yourNode, Server=yourServer` です。
 - c) 「**新規**」をクリックします。
 - d) 「**IBM MQ メッセージング・プロバイダー**」をクリックします。
 - e) トピックのプロパティ・ページの「**管理**」パネルで、「**Name**」フィールドおよび「**JNDI name**」フィールドに固有の値を選択します。これらのフィールドは、後で構成内で参照します。
 - f) 「**IBM MQ トピック**」パネルの「**トピック名**」フィールドに `SYSTEM.FTE/Log/#` と入力します。
6. 次のようにしてアクティベーション・スペックを作成します。
- a) WebSphere Application Server 7.0 の管理コンソールのナビゲーションから、「**リソース**」 > 「**JMS**」 > 「**アクティベーション・スペック**」をクリックします。
 - b) 「**有効範囲**」ドロップダウン・リストを選択し、有効範囲を適切な値に変更します。例えば、`Node=yourNode, Server=yourServer` です。
 - c) 「**新規**」をクリックします。
 - d) 「**IBM MQ メッセージング・プロバイダー**」をクリックします。
 - e) ウィザードのステップ 1 で、「**名前**」フィールドと「**JNDI 名**」フィールドに固有値を選択します。これらの値は、構成の後の段階で再び参照することになります。
 - f) ステップ 1.1 で、ステップ 5 でセットアップしたトピックの JNDI 名を「**宛先 JNDI 名**」フィールドに入力します。
 - g) 「**宛先タイプ**」リストから、「**トピック**」を選択します。
 - h) ウィザードのステップ 1.2 で、「**永続サブスクリプション**」を選択します。「**サブスクリプション名**」フィールドに `SYSTEM.FTE.DATABASELOGGER.AUTO` と入力します。
 - i) ウィザードのステップ 2 で、「**このウィザードに必要な情報をすべて入力**」を選択します。

- j) ステップ 2.1 で、「キュー・マネージャーまたはキュー共用グループ名」フィールドにキュー・マネージャーの名前を入力します。
- k) ステップ 2.2 で、「トランスポート」リストから対象のトランスポート方式を選択します。「バイインディング」を選択すると、他の情報が不要になります。「クライアント」または「バイインディングとクライアント」を選択した場合は、「ホスト名」、「ポート」、および「サーバー接続チャンネル」の詳細情報を入力してください。
- l) オプション: キュー・マネージャーの有無を確認するには、「テスト接続」をクリックします。ただし、ステップ 6n で認証別名の参照を済ませるまでは、NOT_AUTHORIZED を受信することになるはずです。
- m) 「保存」をクリックします。
- n) 作成したアクティベーション・スペックの名前をクリックします。「構成」タブの「一般プロパティ」セクションで、「拡張」パネルにスクロールダウンして、「クライアント ID」フィールドに IBM MQ 接続を識別するための固有の名前を入力します。このステップを完了しないと、IBM MQ によって JMSSC0101 エラー・コードで接続が拒否されます。
- o) トランスポート方式として「クライアント」を選択した場合は、「セキュリティ設定」パネルにスクロールダウンして、ステップ 8 で定義した認証別名を「認証別名」リストから選択します。
- p) 「適用」をクリックします。
- q) 「構成」タブの「追加プロパティ」セクションで、「拡張プロパティ」をクリックします。「拡張プロパティ」パネルの「接続コンシューマー」セクションで、「最大サーバー・セッション」フィールドに 1 を入力します。

注: 先に進む前に、このステップを確実に実行しておきます。そうしないと、ロガーが正しく作動しない可能性があります。

- r) 「構成」タブの「追加プロパティ」セクションで、「拡張プロパティ」をクリックします。「メッセージ送信失敗時はエンドポイントを停止する」の値を最小値の 1 に設定します。

`_numberOfFailedAttemptsBeforeReject` プロパティの値を 1 より大きい値に設定すると (詳細については 9j を参照)、「メッセージ送信失敗時はエンドポイントを停止する」を `_numberOfFailedAttemptsBeforeReject` プロパティの値以上に設定します。これにより、処理できないメッセージ (誤った形式の転送ログ・メッセージ) を受け取った場合でも、エンドポイントは停止されません。詳しくは、331 ページの『MFT ロガーのエラー処理とリジェクト』を参照してください。

7. キュー接続ファクトリーを作成します。

- a) WebSphere Application Server 7.0 の管理コンソールのナビゲーションから、「リソース」 > 「JMS」 > 「キュー接続ファクトリー」をクリックします。
- b) 「有効範囲」ドロップダウン・リストを選択し、有効範囲を適切な値に変更します。例えば、Node=yourNode, Server=yourServer です。
- c) 「新規」をクリックします。
- d) 「IBM MQ メッセージング・プロバイダー」をクリックします。
- e) ウィザードのステップ 1 で、「名前」フィールドと「JNDI 名」フィールドに固有値を選択します。これらの値は、構成の後の段階で再び参照することになります。
- f) ステップ 2 で、「このウィザードに必要な情報をすべて入力」を選択します。
- g) ステップ 2.1 で、「キュー・マネージャーまたはキュー共用グループ名」フィールドにキュー・マネージャーの名前を入力します。
- h) ステップ 2.2 で、「トランスポート」リストから対象のトランスポート方式を選択します。「バイインディング」を選択すると、他の情報が不要になります。「クライアント」または「バイインディングとクライアント」を選択した場合は、「ホスト名」、「ポート」、および「サーバー接続チャンネル」の詳細情報を入力してください。
- i) オプション: キュー・マネージャーの有無を確認するには、「テスト接続」をクリックします。ただし、ステップ 7h で認証別名の参照を済ませるまでは、NOT_AUTHORIZED を受信することになるはずです。

- j) トランスポート方式として「クライアント」または「バインディングとクライアント」を選択した場合は、作成したキュー接続ファクトリーの名前をクリックします。「構成」タブの「セキュリティ設定」パネルにスクロールダウンして、ステップ 2e で定義した認証別名を「XA リカバリーの認証別名」リストと「コンテナ管理認証別名」リストから選択します。

8. WebSphere Application Server でリジェクト・キューを作成します。

- a) WebSphere Application Server 7.0 の管理コンソールのナビゲーションから、「リソース」 > 「JMS」 > 「キュー」をクリックします。
- b) 「有効範囲」ドロップダウン・リストを選択し、有効範囲を適切な値に変更します。例えば、Node=yourNode, Server=yourServer です。
- c) 「新規」をクリックします。
- d) 「IBM MQ メッセージング・プロバイダー」をクリックします。
- e) 「名前」フィールドと「JNDI 名」フィールドに固有値を選択します。これらの値は、構成の後の段階で再び参照することになります。
- f) 「キュー名」フィールドに、SYSTEM.FTE.LOG.RJCT.logger_name と入力します。このキューを調整キュー・マネージャー上に作成したことを確認します。
- g) 「キュー・マネージャー名」フィールドにキュー・マネージャーの名前を入力します。
- h) 「OK」をクリックします。

9. JEE データベース・ロガー・アプリケーションをインストールします。

- a) WebSphere Application Server 7.0 の管理コンソールから、「アプリケーション」 > 「新規アプリケーション」を選択します。
- b) 「有効範囲」ドロップダウン・リストを選択し、有効範囲を適切な値に変更します。例えば、Node=yourNode, Server=yourServer です。
- c) オプション・リストから、「新規エンタープライズ・アプリケーション」を選択します。
- d) 「アプリケーション・インストールの準備」ページで、Managed File Transfer Service のインストール済み環境の MQ_INSTALLATION_PATH/mqft/web ディレクトリーから com.ibm.wmqfte.databaselogger.jee.ear ファイルまたは com.ibm.wmqfte.databaselogger.jee.oracle.ear ファイルを選択し、「次へ」をクリックします。
- e) 次の画面で、「詳細」を選択して、すべてのインストール・オプションおよびパラメーターを表示し、「次へ」をクリックします。
- f) ウィザードのステップ 1 から 4 で、「次へ」をクリックして、デフォルト値を受け入れます。
- g) ウィザードのステップ 5 の「メッセージ駆動型 Bean のリスナーをバインド」で、「リスナー・バインディング」セクションにスクロールします。「アクティベーション・スペック」をクリックします。

以下のフィールドに必要な値を入力します。

ターゲット・リソース JNDI 名

ステップ 6e でアクティベーション・スペックを作成したときに指定した JNDI 名。

宛先 JNDI 名

ステップ 5e でトピックを作成したときに指定した JNDI 名。

「次へ」をクリックします。

- h) ウィザードのステップ 6 の「リソース参照をリソースにマップ」で、「ターゲット・リソース JNDI 名」フィールドに詳細情報を入力します。この名前は、ステップ 7c でリジェクト・キュー接続ファクトリーに指定した JNDI 名です。「次へ」をクリックします。
- i) ウィザードのステップ 7 の「リソース環境エントリー参照をリソースにマップ」で、「ターゲット・リソース JNDI 名」フィールドに詳細情報を入力します。この名前は、ステップ 8d で作成したリジェクト・キューの JNDI 名です。「次へ」をクリックします。
- j) ウィザードのステップ 8 で、EJB モジュールの環境エントリーをマップのデフォルト値である 1 を受け入れます。「次へ」をクリックします。

Oracle RAC:Oracle Real Application Cluster に接続する場合は、`_numberOfFailedtsBeforeReject` プロパティの値を少なくとも 2 に設定する必要があります。このプロパティは、障害が発生した後にロガーが監査メッセージの処理を試行する回数を決定します。データベースのフェイルオーバーでは、少なくとも 1 つの障害が発生する可能性があります。リジェクト・キューへのメッセージの不要な移動を回避するには、この値を大きくして、再度試行されるようにします。これにより、新しいデータベース・インスタンスに接続されるため、通常は成功します。それでもまだ、データベース・インスタンスのフェイルオーバー中にメッセージがリジェクト・キューに移動されることが、テストでわかった場合には、この値をさらに大きくします。インスタンス間の切り替えを行うタイミングにより、同じメッセージに対して複数の障害が引き起こされる場合があるからです。ただし、この値を大きくすると、データベースのフェイルオーバーだけでなく、すべての障害事例 (例えば、誤った形式のメッセージ) に影響します。そのため、値を大きくする際には、不要な再試行が行われないように注意してください。

- k) ウィザードのステップ 9 の「モジュールのメタデータ」で、「次へ」をクリックします。
 - l) ウィザードのステップ 10 の「要約」で、「終了」をクリックします。
10. これで、WebSphere Application Server 7.0 管理コンソールからアプリケーションを開始できるようになりました。
- a) コンソール・ナビゲーションから「アプリケーション」>「アプリケーション・タイプ」>「WebSphere エンタープライズ・アプリケーション」と選択します。
 - b) コレクション表でロガー・エンタープライズ・アプリケーションのチェック・ボックスを選択し、「開始」をクリックします。

MFT with WebSphere Application Server Community Edition 用の Java EE データベース・ロガーのインストール

以下の手順を実行して、Managed File Transfer 用の Java Platform, Enterprise Edition (Java EE) データベース・ロガーをインストールし、構成します (WebSphere Application Server Community Edition)。

始める前に

Java EE データベース・ロガー・アプリケーションをインストールする前に、129 ページの『MFT 用の Java EE データベース・ロガーをインストールするための準備』のトピックに記載された手順に従ってください。

このタスクについて

Java EE データベース・ロガーについて詳しくは、106 ページの『MFT ロガーの構成』のトピックを参照してください。

手順

1. IBM MQ リソース・アダプター `wmq.jmsra.rar` をデプロイします。
 - 調整キュー・マネージャー `QM_JUPITER` を使用して JEE データベース・ロガー用に IBM MQ リソース・アダプターをデプロイするために、以下のステップを実行します。この例は、ご使用の WebSphere Application Server Community Edition インスタンスが、接続先となる IBM MQ キュー・マネージャーと同じシステムで実行している場合に当てはまります。
 - a. MFT 調整キュー・マネージャーに対する接続を定義する、計画ファイルを作成します。以下の計画ファイルの例では、`QM_JUPITER` というキュー・マネージャーへの接続と、そのキュー・マネージャーにある `SYSTEM.FTE.LOG.RJCT.LOGGER1` というキューへの参照を定義しています。

```
<?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>

```

ご使用の環境でこの計画ファイルを使用するには、QM_JUPITER を、調整キュー・マネージャーの名前に変更します。

- b. WebSphere Application Server CE 管理コンソールを開きます。
 - c. ウェルカム・ページの「共通コンソール・アクション」リストから、「新規アプリケーションのデプロイ」 > 「新規デプロイ」をクリックします。
 - d. 「アーカイブ」フィールドに、`mq_install_root/java/lib/jca/wmq.jmsra.rar` と入力します。
 - e. 「計画」フィールドに、ステップ 1a で作成した計画ファイルのパスを入力します。
- ご使用の WebSphere Application Server Community Edition インスタンスが、接続先となる IBM MQ キュー・マネージャーとは異なるシステムで実行されている場合には、以下の手順を実行して IBM MQ リソース・アダプターをデプロイしてください。
 - a. WMQFTE 調整キュー・マネージャーに対する接続を定義する、計画ファイルを作成します。以下の計画ファイルの例では、WebSphere Application Server Community Edition インストール済み環境とは異なるシステムにあるキュー・マネージャー QM_SATURN に対する接続と、そのキュー・マネージャーにある SYSTEM.FTE.LOG.RJCT.LOGGER1 というキューへの参照を定義しています。QM_SATURN のホスト名は `saturn.example.com` です。QM_SATURN のポートは 1415 です。QM_SATURN のチャンネルは SYSTEM.DEF.SVRCONN です。

アプリケーション・サーバーとキュー・マネージャーは異なるシステム上にあるため、キュー・マネージャーにはクライアント・モードの接続を使用する必要があります。次の計画ファイルは、`transportType` という名前を持つ `<config-property-setting>` エレメントの値を、CLIENT に設定します。

```

<?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>
  </resourceadapter>
  <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/WMQFTEJEEDBLoggerRejectQueue</message-destination-name>
    <config-property-setting name="baseQueueManagerName">QM_SATURN</config-property-setting>
    <config-property-setting name="baseQueueName">SYSTEM.FTE.LOG.RJCT.LOGGER1</config-property-setting>
  </adminobject-instance>
</adminobject>
</connector>

```

ご使用の環境でこの計画ファイルを使用するには、QM_SATURN を、調整キュー・マネージャーの名前に変更します。ホスト名、ポート、およびチャネルの値を、ご使用の調整キュー・マネージャーの値に変更してください。

- b. ファイル `mq_install_root/java/lib/jca/wmq/jmsra.rar` を、IBM MQ がインストールされているシステムから WebSphere Application Server CE がインストールされているシステムにコピーします。
 - c. WebSphere Application Server CE 管理コンソールを開きます。
 - d. ウェルカム・ページの「共通コンソール・アクション」リストから、「新規アプリケーションのデプロイ」 > 「新規デプロイ」をクリックします。
 - e. 「アーカイブ」フィールドに、入手した `wmq.jmsra.rar` ファイルのコピーのパスを入力します。
 - f. 「計画」フィールドに、作成した計画ファイルのパスを入力します。
2. WebSphere Application Server Community Edition 環境内から JEE データベース・ロガー・アプリケーションが必要なデータベースにアクセスできるように、データベース・コネクタを定義する必要があります。

WebSphere Application Server Community Edition 管理コンソールから、以下のステップを実行してください。

- a) 使用している WebSphere Application Server Community Edition のレベルに応じて、「コンソール・ナビゲーション」から「サービス」 > 「データベース・プール」を選択するか、「リソース」 > 「データ・ソース」を選択します。
- b) Geronimo データベース・プール・ウィザードを使用して、データベース・プールを作成します。「データベース・プールの名前」フィールドに、`jdbc/wmqfte-database` と入力します。
- c) 「データベース・タイプ」で、ご使用のデータベースに応じて「DB2 XA」または「Oracle Thin」を選択します。
- d) 「次へ」をクリックします。
- e) 「ドライバー JAR」フィールドで、ご使用のデータベースに適した JAR を選択します。
- f) 「データベース名」フィールドに、転送状況情報のために、接続先となるデータベースの名前を入力します。
- g) 「ユーザー名」フィールドに、接続先となってデータベースで認証するユーザー名を入力します。
- h) 「パスワード」および「パスワードの確認」フィールドに、データベースで認証するためのパスワードを入力します。

- i) 「ポート番号」フィールドに、使用するポート番号を入力します (それがデフォルトのポートでない場合)。
 - j) 「ドライバー・タイプ」の値が 4 であることを確認します。
 - k) 「トランザクション・タイプ」リストから XA を選択します。
 - l) 「デプロイ」をクリックします。
3. Managed File Transfer JEE データベース・ロガー・アプリケーションの `openejb-jar.xml` ファイルを、使用環境に応じて更新します。Java SDK の jar ユーティリティを使用して以下のステップを実行してください。
- a) 以下のコマンドを実行して、提供されている EAR ファイルから EJB JAR ファイルを抽出します。

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

`ear_file_name` は、Db2 を使用しているか Oracle を使用しているかに応じて、`com.ibm.wmqfte.databaselogger.jee.ear` または `com.ibm.wmqfte.databaselogger.jee.oracle.ear` のいずれかになります。EAR ファイルは、IBM WebSphere MQ File Transfer Edition Server インストール済み環境の `MQ_INSTALLATION_PATH/mqft/web` ディレクトリーにあります。

- b) 以下のコマンドを実行して、先に解凍した EJB jar ファイル `com.ibm.wmqfte.databaselogger.jee.ejb.jar` から `META-INF/openejb-jar.xml` ファイルを抽出します。

```
jar -xf com.ibm.wmqfte.databaselogger.jee.ejb.jar META-INF/openejb-jar.xml
```

- c) 抽出した `META-INF/openejb-jar.xml` ファイルを、テキスト・エディターを使用して編集します。以下の `activation-config-property` 値を、使用環境に一致するように変更してください。

queueManager

JEE データベース・ロガーによって使用される IBM MQ キュー・マネージャーの名前。

hostName

指定された IBM MQ キュー・マネージャーへの接続に使用されるホスト名。バインディング・モードでキュー・マネージャーに接続する場合、この値は不要です。

transportType

指定された IBM MQ キュー・マネージャーにクライアント・モードで接続するかバインディング・モードで接続するか。

port

transportType に `bindings` を指定した場合は不要。指定された IBM MQ キュー・マネージャーへの接続に使用されるポート。

channel

transportType に `bindings` を指定した場合は不要。指定された IBM MQ キュー・マネージャーへの接続に使用されるサーバー・チャンネル。

- d) 以下のコマンドを実行して、EJB jar ファイルを、変更後の `META-INF/openejb-jar.xml` ファイルで更新します。

```
jar -uf com.ibm.wmqfte.databaselogger.jee.ejb.jar META-INF/openejb-jar.xml
```

- e) 以下のコマンドを実行して、提供されている ear ファイルを、更新後の EJB jar ファイルで更新します。

```
jar -uf ear_file_name com.ibm.wmqfte.databaselogger.jee.ejb.jar
```

`ear_file_name` は、使用しているデータベースに応じて、`com.ibm.wmqfte.databaselogger.jee.ear` または `com.ibm.wmqfte.databaselogger.jee.oracle.ear` のいずれかになります。

4. EAR ファイルをアプリケーション・サーバーにデプロイするには、WebSphere Application Server Community Edition 管理コンソールから、以下のステップを実行します。
 - a) 「**コンソール・ナビゲーション**」メニューから「**アプリケーション**」 > 「**新規デプロイ**」と選択します。
 - b) 「**アーカイブ**」フィールドで EAR ファイルを指定します。使用しているデータベースに応じて、`com.ibm.wmqfte.databaselogger.jee.ear` または `com.ibm.wmqfte.databaselogger.jee.oracle.ear` のいずれかになります。
 - c) 「**計画**」フィールドはブランクのままにしておきます。
 - d) 「**インストール後にアプリケーションを開始**」ボックスが選択されていることを確認します。
 - e) 「**インストール**」をクリックします。JEE データベース・ロガー・アプリケーションがインストールされ、開始します。

MFT 用の Java EE データベース・ロガーのユーザー・アクセスの構成

Managed File Transfer Java Platform, Enterprise Edition (Java EE) データベース・ロガーを構成する場合、IBM MQ、データベース、およびオペレーティング・システムにアクセスするためのユーザー・アカウントが必要です。必要なオペレーティング・システム・ユーザーの数は、それらのコンポーネントのホストとして使用するシステムの数によって異なります。

このタスクについて

Java EE データベース・ロガーを実行するために必要なユーザー・アカウントの数とタイプは、使用するシステムの数によって異なります。以下の 3 つの環境にアクセスするためにユーザー・アカウントが必要です。

- ローカル・オペレーティング・システム
- IBM MQ
- データベース

Java EE データベース・ロガー、IBM MQ、およびデータベースは、1 つのシステムにインストールすることも、複数のシステムに分けてインストールすることも可能です。それらのコンポーネントをインストールするためのトポロジーの例を以下に示します。

Java EE データベース・ロガー、IBM MQ、およびデータベースをすべて同じシステムに配置する場合

3 つのコンポーネントすべてで使用する 1 つのオペレーティング・システム・ユーザーを定義できます。ロガーは、**バインディング・モード**を使用して IBM MQ に接続し、**ネイティブ接続**を使用してデータベースに接続します。

Java EE データベース・ロガーと IBM MQ を 1 つのシステムに配置し、別のシステムにデータベースを配置する場合

この構成では、2 つのユーザーを作成します。つまり、ロガーを実行するシステムのオペレーティング・システム・ユーザーと、データベース・サーバーのデータベースに対するリモート・アクセスを実行するオペレーティング・システム・ユーザーです。ロガーは、**バインディング・モード**を使用して IBM MQ に接続し、**クライアント接続**を使用してデータベースにアクセスします。

Java EE データベース・ロガー、IBM MQ、データベースをそれぞれ別々のシステムに配置する場合

この構成では、3 つのユーザーを作成します。つまり、アプリケーション・サーバーを開始するオペレーティング・システム・ユーザーと、使用するキューとトピックにアクセスする IBM MQ ユーザーと、データベース表にアクセスしてデータを挿入するデータベース・サーバー・ユーザーです。ロガーは、**クライアント・モード**を使用して IBM MQ にアクセスし、**クライアント接続**を使用してデータベースにアクセスします。

以下の説明では、例として `ftelog` というユーザーを想定しますが、任意の新規または既存のユーザー名を使用できます。次のようにして、ユーザーの許可を構成します。

手順

1. オペレーティング・システムのユーザーが、独自のグループに所属していて、かつ調整キュー・マネージャーの広範囲の許可を備えたグループに所属していないことを確実にしてください。ユーザーは、

mqm グループに所属してはなりません。特定のプラットフォームでは、staff グループに、キュー・マネージャーアクセス権限も自動的に付与されます。ロガー・ユーザーは staff グループに入れないでください。IBM MQ Explorer を使用すると、キュー・マネージャー自体およびキュー・マネージャー内のオブジェクトの権限レコードを表示できます。オブジェクトを右クリックして、「**オブジェクト権限**」>「**権限レコードの管理**」をクリックします。コマンド行では、コマンド `dspmqaout` (権限の表示) または `dmpmqaut` (権限のダンプ) を使用できます。

2. IBM MQ Explorer の「**権限レコードの管理**」ウィンドウまたは `setmqaut` (権限の付与または取り消し) コマンドを使用して、IBM MQ ユーザー自身のグループの権限を追加します (UNIX の場合、IBM MQ 権限はグループにのみ関連付けられ、個々のユーザーには関連付けられません)。必要な権限は以下のとおりです。

- キュー・マネージャーに対する CONNECT と INQUIRE (IBM MQ Java ライブラリーを操作するには、INQUIRE 権限が必要です)。
- SYSTEM.FTE トピックに対する SUBSCRIBE 権限。
- SYSTEM.FTE.LOG.RJCT.*logger_name* キューに対する PUT 権限。

リジェクト・キュー名とコマンド・キュー名は、デフォルトの名前です。ロガー・キューの構成時に別のキュー名を選択した場合は、それらのキュー名に対する権限を追加してください。

3. 使用中のデータベースに該当する固有のデータベース・ユーザー構成を実行します。

- データベースが Db2 の場合は、以下の手順を実行します。

注：Db2 でデータベース・ユーザーを管理するメカニズムはいくつかあります。ここでの説明は、オペレーティング・システム・ユーザーに基づいたデフォルト・スキームに適用されます。

- `ftelog` ユーザーが Db2 管理グループ (例: `db2iadm1`、`db2fadm1`、`dasadm1`) に所属していない状態にしてください
 - このユーザーに、データベースに接続する許可、および『[ステップ 2: 必要なデータベース表の作成](#)』で作成した表に対する `select`、`insert`、および `update` の許可を付与します
- データベースが Oracle の場合は、以下の手順を実行します。
 - `ftelog` ユーザーが Oracle 管理グループ (例: Windows での `ora_dba` や UNIX での `dba`) に所属していない状態にしてください
 - このユーザーに、データベースに接続する許可、および『[ステップ 2: 必要なデータベース表の作成](#)』で作成した表に対する `select`、`insert`、および `update` の許可を付与します

Java EE データベース・ロガーのマイグレーション

WebSphere Application Server 7.0 上の Java EE データベース・ロガーを IBM WebSphere MQ File Transfer Edition 7.0 から IBM WebSphere MQ 7.5 以降にマイグレーションするには、以下の手順を実行します。

手順

1. WebSphere Application Server コンソールを開きます。
2. 「**アプリケーション**」>「**アプリケーション・タイプ**」>「**エンタープライズ・アプリケーション**」をクリックします。アプリケーションのリストで IBM WebSphere MQ File Transfer Edition データベース・ロガー・アプリケーションを探します。データベース・ロガー・アプリケーションがまだ停止されていない場合は、アプリケーションを選択して「**停止**」をクリックします。
3. 以前にその JEE データベース・ロガー用にセットアップした構成設定をメモしておきます。このメモは、この後のステップ [142](#) ページの『[7](#)』で必要になります。
 - a) データベース・ロガーをインストールしながら最初に EJB モジュールのデフォルト設定から変更している場合は、(詳細はステップ [9](#) を参照)、「**エンタープライズ・アプリケーション**」>「**WebSphere MQ File Transfer Edition データベース・ロガー**」>「**EJB モジュールの環境エントリ**」をクリックして、ペインの設定をメモします。
 - b) 「**エンタープライズ・アプリケーション**」>「**WebSphere MQ File Transfer Edition データベース・ロガー**」>「**メッセージ駆動型 Bean リスナー・バインディング**」をクリックし、使用されているアクティベーション・スペック、「**ターゲット・リソース JNDI 名**」、および「**宛先 JNDI 名**」をメモしておきます。

- c) 「エンタープライズ・アプリケーション」 > 「WebSphere MQ File Transfer Edition データベース・ロガー」 > 「リソース参照」をクリックし、リジェクト・キュー接続ファクトリーの詳細をメモしておきます。
 - d) 「エンタープライズ・アプリケーション」 > 「WebSphere MQ File Transfer Edition データベース・ロガー」 > 「リソース環境エントリ参照」をクリックし、リジェクト・キューの詳細をメモしておきます。
4. 「アプリケーション」 > 「アプリケーション・タイプ」 > 「エンタープライズ・アプリケーション」をクリックして、IBM WebSphere MQ File Transfer Edition データベース・ロガー・アプリケーションをアンインストールします。データベース・ロガー・アプリケーションを選択して、「アンインストール」をクリックします。
 5. オプション: IBM WebSphere MQ 7.5 以降にマイグレーションするインストール環境を複数使用していて、ネイティブ・ライブラリーのパスが異なっている場合は、「リソース」 > 「JMS プロバイダー」 > 「WebSphere MQ メッセージング・プロバイダー」をクリックして、パスを変更します。
例えば、ネイティブ・ライブラリー・パスが C:\Program Files\IBM\WebSphere MQ\java\lib であった場合は、パスを C:\Program Files\IBM\New MQ Installation Location\java\lib に変更します。
 6. オプション: IBM WebSphere MQ 7.5 以降にマイグレーションするインストール環境を複数使用している場合は、setmqm コマンドを使用して、キュー・マネージャーを新しいインストール環境に関連付ける必要があります。
 7. 「WebSphere Application Server 7.0 を使用した IBM MQ Managed File Transfer Java EE データベース・ロガーのインストール」の情報、および前のステップ 141 ページの『3』で記録した情報を使用して、データベース・ロガー・アプリケーションを再インストールします。
 8. 「アプリケーション」 > 「アプリケーション・タイプ」 > 「エンタープライズ・アプリケーション」をクリックして、新しいデータベース・ロガーを開始します。データベース・ロガー・アプリケーションを選択して、「開始」をクリックします。
 9. マイグレーションの検査を行うには、データベースを調べて、各項目が書き込まれていることを確認します。

スタンドアロン・データベース・ロガーから MFT 用の Java EE データベース・ロガーへのマイグレーション

スタンドアロン・データベース・ロガーから Java EE データベース・ロガーにマイグレーションできます。スタンドアロン・データベース・ロガーを停止して、Java EE データベース・ロガーをインストールする必要があります。ログ項目の消失または重複を回避するには、スタンドアロン・データベース・ロガーを停止する前に、SYSTEM.FTE トピックへのメッセージのパブリッシュを停止し、JEE データベース・ロガーをインストールした後にこれを再始動する必要があります。マイグレーション前にデータベースをバックアップしてください。

このタスクについて

手順

1. データベースを停止する前に、調整キュー・マネージャーに対して次の MQSC コマンドを実行します。
ALTER QM PSMODE(COMPAT)
これによって SYSTEM.FTE/Log トピックへのメッセージのパブリッシュが停止します。ロガーがそのサブスクリプション上のメッセージをすべて処理するまで待ちます。デフォルトでは、このサブスクリプションは SYSTEM.FTE.LOGGER.AUTO という名前です。
2. **fteStopLogger** コマンドを使用して、データベース・ロガーを停止します。
3. データベース・ソフトウェアに付属のツールを使用して、データベースをバックアップします。
4. スタンドアロンのデータベース・ロガーに属するサブスクリプションを削除します。
デフォルトでは、このサブスクリプションは SYSTEM.FTE.LOGGER.AUTO という名前です。
5. 使用中のデータベース・スキーマが以前のバージョンの場合、スキーマを順番に以降のレベルにマイグレーションする必要があります。例えば、データベース・スキーマが 7.0.1 であり、7.0.4 にマイグレーションする必要があります。

ーションする場合は、7.0.1 から 7.0.2 に、次に 7.0.2 から 7.0.3 に、さらに 7.0.3 から 7.0.4 に、それぞれマイグレーションする必要があります。マイグレーション時に通過するスキーマのバージョンごとに以下のいずれかの操作を実行して、バージョン *old* からバージョン *new* にデータベース・スキーマをマイグレーションします (*old* と *new* は、スキーマの各バージョンに相当する変数です)。

- **z/OS** ご使用のデータベースが z/OS 上の Db2 であり、7.0.2 スキーマと 7.0.3 スキーマの間、または 7.0.3 スキーマと 7.0.4 スキーマの間でマイグレーションする場合は、新しいデータベース・スキーマを作成し、既存のデータをそこにコピーする必要があります。詳しくは、[22 ページの『z/OS 上の Db2 上のデータベース表の IBM MQ 8.0 へのマイグレーション』](#)を参照してください。
 - データベースが Db2 でない場合や、8K を超えるページ・サイズでデータベースを作成していた場合は、以下の手順を実行して、他のバージョンの場合と同じ要領でスキーマをマイグレーションできます。
 - その他の環境でデータベース表をマイグレーションする場合は、以下の手順を実行します。
 - a. ご使用のデータベース・プラットフォームに適した、名前にストリング *old-new* が含まれているファイルを選択します。このファイルは、Remote Tools and Documentation インストール済み環境の `MQ_INSTALLATION_PATH/mqft/sql` ディレクトリにあります。
 - b. 初期スキーマに対して変更を行った場合は、マイグレーション・ファイルを調べて、ファイルと変更済みデータベースとの互換性が確保されるようにしてください。
 - c. ご使用のデータベースに対して SQL ファイルを実行します。
6. Java EE データベース・ロガー EAR ファイルをインストールします。
 7. Java EE データベース・ロガーをデプロイします。詳細については [128 ページの『MFT 用の Java EE データベース・ロガーのインストール』](#)を参照してください。
 8. 調整キュー・マネージャーに対して次の MQSC コマンドを実行します。ALTER QMGR PSMODE(ENABLED)
これによってメッセージが SYSTEM.FTE/Log トピックにパブリッシュされるようになります。

タスクの結果

Connect:Direct ブリッジの構成

Managed File Transfer ネットワークと Connect:Direct ネットワークの間でファイルを転送するために、Connect:Direct ブリッジを構成します。Connect:Direct ブリッジのコンポーネントは、Connect:Direct ノードと、そのノードと通信するための専用の Managed File Transfer エージェントです。このエージェントのことを Connect:Direct ブリッジ・エージェントといいます。

このタスクについて

以下の手順を実行して、Connect:Direct ブリッジを構成します。

手順

1. [144 ページの『Connect:Direct ブリッジのエージェントおよびノードのオペレーティング・システムを選択する』](#)。
2. [144 ページの『Connect:Direct ノードを選択および構成する』](#)。
3. [145 ページの『Connect:Direct ブリッジ・エージェントを作成および構成する』](#)。
4. [145 ページの『ConnectDirectNodeProperties.xml ファイルを構成して、リモート Connect:Direct ノードに関する情報を組み込む』](#)。
5. [146 ページの『Connect:Direct ブリッジ・エージェントと Connect:Direct ノードの間のセキュア接続を構成する』](#)。

Connect:Direct ブリッジのエージェントおよびノードのオペレーティング・システムを選択する

始める前に

Connect:Direct ブリッジを構成するエージェントとノードは、同じシステムに存在しているか、共有 NFS マウントなどによって同じファイル・システムにアクセスできる状態になっている必要があります。このファイル・システムは、Connect:Direct ブリッジがかかっているファイル転送中に、**cdTmpDir** パラメーターで定義されているディレクトリーに一時的にファイルを保管するために使用されます。

Connect:Direct ブリッジ・エージェントと Connect:Direct ブリッジ・ノードでは、同じパス名を使用してこのディレクトリーを指定する必要があります。例えば、エージェントとノードが別個の Windows システムにある場合、共有ファイル・システムをマウントするためにそれらのシステムで同じドライブ名が使用されている必要があります。以下の構成を使用すると、エージェントとノードで同じパス名を使用できます。

- エージェントとノードが、Windows または Linux for System x のいずれかを実行する同じシステム上にある
- エージェントが Linux for System x 上にあり、ノードが UNIX 上にある
- エージェントが Windows システム上にあり、ノードがそれとは別の Windows システム上にある

以下の構成を使用すると、エージェントとノードで同じパス名を使用できません。

- エージェントが Linux for System x 上にあり、ノードが Windows 上にある
- エージェントが Windows 上にあり、ノードが UNIX 上にある

Connect:Direct ブリッジのインストールを計画する際には、これらの制約事項を考慮してください。

Connect:Direct ブリッジでサポートされているオペレーティング・システム・バージョンの詳細については、Web ページ [IBM MQ System Requirements](#) を参照してください。

手順

1. Connect:Direct ブリッジ・エージェントをインストールするシステムとして Windows または Linux on System x のいずれかを実行するシステムを選択します。
2. Connect:Direct ブリッジ・ノードをインストールするオペレーティング・システムとして Connect:Direct for Windows または Connect:Direct for UNIX でサポートされているものを選択します。

Connect:Direct ノードを選択および構成する

始める前に

以下の手順を実行する前に、Connect:Direct ノードをインストールしておく必要があります。

手順

1. Managed File Transfer エージェントと通信する Connect:Direct ノードを選択します。
2. 選択した Connect:Direct ノードのネットワーク・マップを確認します。Windows オペレーティング・システム上で実行中のリモート・ノードの項目がネットワーク・マップに含まれている場合は、それらのノードが Windows で実行されていることが、それらの項目で指定されていることを確認します。
 - a) Connect:Direct ブリッジに対して選択した Connect:Direct ノードが Windows 上で実行されている場合、Connect:Direct Requester を使用してネットワーク・マップを編集します。Windows 上で実行されているリモート・ノードの「オペレーティング・システム」フィールドが「**Windows**」に設定されていることを確認します。

Connect:Direct ブリッジ・エージェントを作成および構成する

このタスクについて

Connect:Direct ブリッジ・エージェントは、Connect:Direct ノードと通信するための専用の Managed File Transfer エージェントです。

手順

1. **fteCreateCDAgent** コマンドを使用して、Connect:Direct ブリッジ・エージェントを作成します。
 - a) **cdNode** パラメーターには値を指定する必要があります。このパラメーターでは、エージェントが Connect:Direct ブリッジの一部になる Connect:Direct ノードに対して使用する名前を指定します。前のセクションで選択した Connect:Direct ノードの名前を使用します。
 - b) **cdNodeHost** パラメーターおよび **cdNodePort** パラメーターの値を指定します。これらは、エージェントが通信する Connect:Direct ノードを定義します。
cdNodeHost パラメーターに値を指定しないと、ローカル・システムのホスト名または IP アドレスが使用されます。**cdNodePort** パラメーターに値を指定しないと、値 1363 が使用されます。
 - c) [402 ページの『fteCreateCDAgent \(Connect:Direct ブリッジ・エージェントの作成\)』](#)の情報をを使用して、**cdTmpDir** パラメーターの値を指定する必要があるかどうかを判別します。
2. Managed File Transfer で使用するユーザー資格情報を、Connect:Direct ノード上のユーザー資格情報にマップします。以下のいずれかの方法を使用して資格情報をマップできます。
 - **ConnectDirectCredentials.xml** ファイルを作成して、資格情報マッピング情報を定義する。詳細については、[146 ページの『ConnectDirectCredentials.xml ファイルを使用した Connect:Direct の資格情報のマップ』](#)を参照してください。
 - 使用する Connect:Direct ブリッジ用に資格情報マッピングを実行するユーザー出口を作成する。詳細については、[149 ページの『出口クラスを使用した Connect:Direct の資格情報のマップ』](#)を参照してください。

ConnectDirectNodeProperties.xml ファイルを構成して、リモート Connect:Direct ノードに関する情報を組み込む

始める前に

以下の手順を実行する前に、Connect:Direct ブリッジ・エージェントを作成しておく必要があります。

手順

Connect:Direct ブリッジ・エージェントの構成ディレクトリーにある **ConnectDirectNodeProperties.xml** テンプレートを編集します。情報を定義する Connect:Direct ノードまたはノード・グループごとに、以下の手順を実行します。

- a) **nodeProperties** エレメントの内部に、**node** エレメントを作成します。
- b) **node** 要素に **name** 属性を追加する。1つ以上のリモート Connect:Direct ノードの名前に合致するパターンとして、この属性の値を指定します。
- c) オプション: **node** 要素に **pattern** 属性を追加し、**name** 属性内の値がどのような種類のパターンであるかを指定します。有効な値は、**regex** と **wildcard** です。デフォルト・オプションは **wildcard** です。
- d) **node** 要素に、**name** 属性で指定されたリモート Connect:Direct ノードが動作するオペレーティングシステムを指定する **type** 属性を追加する。
有効な値は、以下のとおりです。
 - Windows - ノードは Windows で実行されています
 - UNIX - ノードは UNIX または Linux で実行されています

- **z/OS** z/OS、zos、os/390、os390 - ノードは z/OS で実行されています

この属性の値には、大/小文字の区別がありません。Connect:Direct ブリッジは、その他のオペレーティング・システムのリモート・ノードへの転送には対応していません。

詳しくは、[588 ページの『Connect:Direct ノードのプロパティ・ファイルのフォーマット』](#)を参照してください。

Connect:Direct ブリッジ・エージェントと Connect:Direct ノードの間のセキュア接続を構成する

このタスクについて

デフォルトでは、Connect:Direct ブリッジ・エージェントは、TCP/IP プロトコルを使用して Connect:Direct ノードに接続します。Connect:Direct ブリッジ・エージェントと Connect:Direct ノードの間にセキュア接続を確立する場合は、SSL プロトコルまたは TLS プロトコルを使用できます。

手順

セキュア接続を構成します。その方法を示す例については、[54 ページの『Connect:Direct ブリッジ・エージェントと Connect:Direct ノードの間の SSL または TLS の構成』](#)を参照してください。

Connect:Direct の資格情報のマップ

Connect:Direct ブリッジ・エージェントのデフォルトの資格情報マッピング機能を使用するか、独自のユーザー出口を作成することによって、Managed File Transfer 内のユーザー資格情報を Connect:Direct ノード上のユーザー資格情報にマップします。Managed File Transfer には、ユーザー資格情報マッピングを実行するサンプルのユーザー出口が用意されています。

ConnectDirectCredentials.xml ファイルを使用した Connect:Direct の資格情報のマップ

Connect:Direct ブリッジ・エージェントのデフォルトの資格情報マッピング機能を使用して、Managed File Transfer 内のユーザー資格情報を Connect:Direct ノード上のユーザー資格情報にマップします。Managed File Transfer で提供される XML ファイルを編集して、ユーザーの資格情報を組み込むことができます。

このタスクについて

fteCreateCDAgent コマンドを使用して Connect:Direct ブリッジ・エージェントが作成された後、ConnectDirectCredentials.xml ファイルを手動で作成する必要があります。Connect:Direct ブリッジ・エージェントを使用するには、まずこのファイルを編集して、ホスト、ユーザー、および資格情報を含める必要があります。詳細内容は [を参照してください](#)。デフォルトでは、このファイルは現在のホーム・ディレクトリー (例えば、/home/fteuser/ConnectDirectCredentials.xml) からロードされます。別の場所を使用するには、ConnectDirectNodeProperties.xml ファイル内の <credentialsFile> エレメントで指定します。

手順

1. <tns:pnode name="Connect:Direct node host" pattern="wildcard"> エレメントの name 属性には、必ず Connect:Direct ブリッジ・エージェントが接続する Connect:Direct ノードの名前の値を含めてください。この値は、**fteCreateCDAgent -cdNode** パラメーターに指定した値と同じ値でなければなりません。
pattern 属性の値は、wildcard または regex のいずれかです。この属性を指定しない場合、デフォルトは wildcard です。
2. <tns:pnode> の子エレメントとして、ユーザー ID および資格情報をファイルに挿入します。

以下の<tns:user>エレメントの1つ以上のインスタンスをファイルに挿入することができます:

```
<tns:user name="name"
  pattern="pattern"
  ignorecase="ignorecase"
  cdUserId="cdUserId"
  cdPassword="cdPassword"
  pnodeUserId="pnodeUserId"
  pnodePassword="pnodePassword">
</tns:user>
```

ここで、

- *name* は、MFT 転送要求に関連付けられた MQMD ユーザー ID と一致するパターンです。
- *pattern* は、*name* 属性に指定されたパターンがワイルドカード表現であるか、Java 正規表現であるかを指定します。 *pattern* 属性の値は、wildcard または regex のいずれかです。 この属性を指定しない場合、デフォルトは wildcard です。
- *ignorecase* は、*name* 属性で指定されたパターンを大/小文字の区別があるものとして扱うかどうかを指定します。 この属性を指定しない場合、デフォルトは true です。
- *cdUserId* は、Connect:Direct ブリッジ・エージェントが、<tns:pnode> エレメントの *name* 属性で指定された Connect:Direct ノードに接続するために使用するユーザー ID です。 可能な場合は、*cdUserId* に Connect:Direct 管理者ユーザー ID を指定してください。 *cdUserId* に Connect:Direct 管理者を指定できない場合は、使用するユーザー ID が Connect:Direct ブリッジ・ノードにおいて以下の機能権限を持つようにしてください。
 - Windows ノードの場合、以下の権限を設定します。 この例では、読みやすくするために復帰文字を使用してフォーマットしています。

```
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
```

- UNIX ノードの場合、*userfile.cfg* ファイルに以下のパラメーターを設定します。

```
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* は、*cdUserId* 属性で指定されたユーザー ID に関連付けられているパスワードです。

- オプションで、`pnodeUserId` 属性を指定できます。この属性の値は、`<tns:pnode>` エレメントの `name` 属性で指定された `Connect:Direct` ノードが `Connect:Direct` プロセスを実行依頼するために使用するユーザー ID です。`pnodeUserId` 属性を指定しない場合、`Connect:Direct` ノードは、`cdUserId` 属性で指定されたユーザー ID を使用して `Connect:Direct` プロセスを実行依頼します。
- オプションで、属性 `pnodePassword` を指定できます。この属性の値は、`pnodeUserId` 属性で指定されたユーザー ID に関連付けられているパスワードです。

MQMD ユーザー ID に一致するユーザー・エレメントがない場合、転送は失敗します。

3. オプション: 1 つ以上の `<tns:snode>` エレメントを、`<tns:user>` エレメントの子エレメントとして含めることができます。`<tns:snode>` エレメントでは、`Connect:Direct` ブリッジの一部である `Connect:Direct` ノードで使用する資格情報を指定します。この資格情報は、`Connect:Direct` ブリッジ・ノードが、ファイル転送のソースまたは宛先である `Connect:Direct` ノードに接続するとき使用するユーザー ID とパスワードです。

ファイルに以下のエレメントを 1 つ以上挿入します。

```
<tns:snode name="name"
           pattern="pattern"
           userId="userId"
           password="password" />
```

ここで、

- `name` は、ファイル転送の転送元または転送先である `Connect:Direct` ノードの名前とマッチングさせるパターンです。
- `pattern` は、`name` 属性に指定されたパターンがワイルドカード表現であるか、Java 正規表現であるかを指定します。`pattern` 属性の値は、`wildcard` または `regex` のいずれかです。この属性を指定しない場合、デフォルトは `wildcard` です。
- `userId` は、`<tns:pnode>` エレメントの `name` 属性で指定された `Connect:Direct` ノードが、`<tns:snode>` の `name` 属性で指定されたパターンに一致する `Connect:Direct` ノードに接続するために使用されるユーザー ID です。
- `password` は、`userId` 属性で指定されたユーザー ID に関連付けられているパスワードです。

ファイル転送の二次ノードと一致する `<tns:snode>` エレメントがない場合は、転送が失敗することはありません。転送が開始され、`snode` で使用するユーザー ID およびパスワードは指定されません。

タスクの結果

ユーザー名または `Connect:Direct` ノード名をパターン・マッチで検索するとき、`Connect:Direct` ブリッジ・エージェントは、ファイルの先頭から末尾に向かって検索します。最初に見つかった一致が使用されます。

関連タスク

[143 ページの『Connect:Direct ブリッジの構成』](#)

Managed File Transfer ネットワークと `Connect:Direct` ネットワークの間でファイルを転送するために、`Connect:Direct` ブリッジを構成します。`Connect:Direct` ブリッジのコンポーネントは、`Connect:Direct` ノードと、そのノードと通信するための専用の Managed File Transfer エージェントです。このエージェントのことを `Connect:Direct` ブリッジ・エージェントといいます。

関連資料

[584 ページの『Connect:Direct 資格情報ファイル・フォーマット』](#)

Managed File Transfer Agent 構成ディレクトリーの `ConnectDirectCredentials.xml` ファイルは、`Connect:Direct` エージェントが `Connect:Direct` ノードに対して自身を許可するのに使用するユーザー名および資格情報を定義します。

[402 ページの『fteCreateCDAgent \(Connect:Direct ブリッジ・エージェントの作成\)』](#)

`fteCreateCDAgent` コマンドは、Managed File Transfer Agent、および `Connect:Direct` ブリッジのための関連する構成を作成します。

出口クラスを使用した **Connect:Direct** の資格情報のマップ

Connect:Direct ブリッジ・エージェントのデフォルトの資格情報マッピング機能を使用しない場合は、独自のユーザー出口を作成して、Managed File Transfer 内のユーザー資格情報を Connect:Direct ノード上のユーザー資格情報にマップできます。資格情報マッピングの独自のユーザー出口を構成すると、デフォルトの資格情報マッピング機能が使用不可になります。

このタスクについて

Connect:Direct の資格情報のマッピングのために作成するユーザー出口では、`com.ibm.wmqfte.exitroutine.api.ConnectDirectCredentialExit` インターフェースを実装する必要があります。詳細内容は [を参照してください](#)。

Connect:Direct ブリッジ・エージェントと Connect:Direct ノードの間の SSL または TLS の構成

Connect:Direct ブリッジ・エージェントと Connect:Direct ノードが SSL プロトコルを使用して相互に接続するように構成します。そのためには、鍵ストアとトラストストアを作成し、Connect:Direct ブリッジ・エージェントのプロパティ・ファイルでプロパティを設定します。

このタスクについて

ここでは、認証局から鍵の署名を得るための手順を含めています。認証局を使用しない場合は、自己署名証明書を生成できます。自己署名証明書を生成する方法の詳細については、[UNIX および Windows システムでの SSL または TLS の取り扱いを参照してください](#)。

ここでは、Connect:Direct ブリッジ・エージェントの新しい鍵ストアとトラストストアを作成するための手順を含めています。Connect:Direct ブリッジ・エージェントに、IBM MQ キュー・マネージャーへのセキュア接続で使用できる鍵ストアとトラストストアが既にある場合は、Connect:Direct ノードへのセキュア接続で既存の鍵ストアとトラストストアを使用できます。詳細については [51 ページの『MFT の SSL または TLS 暗号化の構成』を参照してください](#)。

手順

Connect:Direct ノードの場合、以下のステップを実行します。

1. Connect:Direct ノードの鍵と署名付きの証明書を生成します。
そのためには、IBM MQ に用意されている IBM 鍵管理ツールを使用できます。詳細については、[SSL または TLS の取り扱いを参照してください](#)。
2. 鍵の署名を得るための要求を認証局に送信します。返ってくる証明書を受け取ります。
3. 認証局の公開鍵を組み込んだテキスト・ファイル (例えば `/test/ssl/certs/CAcert`) を作成します。
4. Connect:Direct ノードに Secure+ オプションをインストールします。
ノードが既に存在している場合は、インストーラーを再び実行し、既存のインストール環境の場所を指定し、Secure+ オプションだけのインストールを選択することによって、Secure+ オプションをインストールできます。
5. 新しいテキスト・ファイル (例えば `/test/ssl/cd/keyCertFile/node_name.txt`) を作成します。
6. 認証局から受け取った証明書と、`/test/ssl/cd/privateKeys/node_name.key` にある秘密鍵をそのテキスト・ファイルにコピーします。
`/test/ssl/cd/keyCertFile/node_name.txt` の内容は、以下の形式になっていなければなりません。

```
-----BEGIN CERTIFICATE-----
MIICnzCCAgigAwIBAgIBGjANBgkqhkiG9w0BAQUFADBeMQswCQYDVQQGEwJH0jES
MBAGA1UECBMJSjSGFtcHNoaXJ1MRAdBgYDVQQHEwIdXJzbGV5M0wwCgYDVQQKEwNJ
Qk0xDjAMBGNVBAoTBU1RSVBUMQswCQYDVQQDEwJDTAeFw0xMTAzMDEwNDZa
Fw0yMTAyMjYxNjIwNDZaMFAsCzAJBgNVBAYTAkdCMRIwEAYDVQQIEw1IYW1wc2hp
cmUxDDAKBgNVBAoTA01CTTE0MAwGA1UECzMFTVFGVUEuXzANBgNVBAMTBmJpbmJh
ZzCBnzANBgkqhkiG9w0BAQEFAA0BjQAwGyKCGYEAvgP1QIklu9ypSKD1XoDo1yk
```

```
EyMFXB0UpZr1DVxjoSEC0vtWncJ199e+Vc4UpNybdyBu+Nkd1MNoF4QxeQcLAFj
WnhakqCiQ+JIAD5AurhnriwChe0MV3kjA84GKH/r0SVqt1984mu/1DyS819XcfSSn
c00MsK1KbneVSCIV2XECaWEAAa7MHkwCQYDVR0TBAlwADAsBglghkgBhvhCAQ0E
HxYdT3Blb1NTTCBHZW51cmF0ZWQgQ2VydG1maWNoGdUwHQYDVR00BBYEFNXMIpSc
csBXUniW4A3UzrZnCRsv3MB8GA1UdIwQYMBaAFDXY8rmj41Vz5+FVAoQb++cns+B4
MA0GCSqGSIsB3DQEBBQUAA4GBAFc7k1Xa4pGKYgwhchxKpE3ZF6FNwy4vBXS216/ja
8h/v18+iv010CL8t0ZOKSU95fyZLzOPKnCH7v+ItFSE3CIIEk9D1z2U6W091ICwn
17PL72Tdfal3kabwHYVf17IVcuL+VZsZ3HjLggP2qH09ZuJPspeT9+AxFVMLiaAb
8eHw
```

```
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,64A02DA15B6B6EF9
```

```
57kqxL0J/gRU0IQ6hVK2YN13B4E1jAi1gSme0I5ZpEIG8CHXISKB7/0cke2FTqsV
lvI99QyCxsDwoMnt5fj51v7aPmVeS60b0m+U1Gre8B/Ze18JVj204K2U72rDCXE
5e6eFxsDUM207sQDy20euBVELJtM2k0kL1R0doQQS1U3XQNgJw/t3ZIx5hPXWEQT
rjRQ064BEhb+PzzxPF8uwwZ9IrkUK9BJ/UUnqC60dBR87IeA4pnJD1Jvb2ML7EN9Z
5Y+50hTKI80GvBvWX04fHyvIX5as1whBoArXIS1AtNTprtPvoaP1zyIAeZ60Cvo/
SFo+A2UhmteJe0JaZG2XZ3H495fAw/EHmjehzIACwukQ9nSIEtgu4A1+CV64RJED
aYBCM8UjaAkbZDH5gn7+eBov0ssXAXWdyJBVhU0jXjvAj/e1h+kcSF1hax5D//AI
66nRMZzboSxNqkjcVd8wfdwP+bEjDzUaaarJTS71IFeLW7eJ8MNAkMGicDkycL0
EPBU9X5QnHKLKOfYHN/1WgUk8qt3UytFXXfzTXGF3EbsWbBupkT5e5+1YcX80VZ6
sHFPN1HluCny/riUcBy9iviVeodX8Iom0chSy05DK18bwZNjYtUP+CtYHNFU5BaD
I+1uU0AeJ+wjYKT1WaeIGZ3VxuNITJu18y5qDTXXfX7vxM50oWxa6U5+AYuGUMg
/itPZmUmNrhjT7ghT6i1IQ0aBowXXKJB1Mmq/6BQXN2IhkD9ys2qrvM1hdi5nAf
egmdiG50l0LnBRqWbFR+DykpAhK4SaDi2F52Uxovw3Lh1w8dQp71zQ==
-----END RSA PRIVATE KEY-----
```

7. Secure+ 管理ツールを開始します。

- Linux または UNIX のシステムでは、**spadmin.sh** コマンドを実行します。
- Windows のシステムでは、「スタート」 > 「プログラム」 > 「Sterling Commerce Connect:Direct」 > 「CD Secure+ 管理ツール」をクリックします。

CD Secure+ 管理ツールが開始します。

8. CD Secure+ 管理ツールで、**.Local** の行をダブルクリックして、SSL または TLS のメイン設定を編集します。

- a) 使用するプロトコルに応じて、「**SSL プロトコルを有効にする**」または「**TLS プロトコルを有効にする**」を選択します。
- b) 「**オーバーライドを無効にする**」を選択します。
- c) 少なくとも 1 つの暗号スイートを選択します。
- d) 双方向認証を設定する場合は、「**クライアント認証を有効にする**」の値を「はい」に変更します。
- e) 「**トラステッド・ルート証明書**」フィールドに、認証局の公開証明書ファイルのパス (/test/ssl/certs/CAcert) を入力します。
- f) 「**鍵証明書ファイル**」フィールドに、作成したファイルのパス (/test/ssl/cd/keyCertFile/node_name.txt) を入力します。

9. **.Client** の行をダブルクリックして、SSL または TLS のメイン設定を編集します。

- a) 使用するプロトコルに応じて、「**SSL プロトコルを有効にする**」または「**TLS プロトコルを有効にする**」を選択します。
- b) 「**オーバーライドを無効にする**」を選択します。

Connect:Direct ブリッジ・エージェントの場合は、以下の手順を実行します。

10. トラストストアを作成します。そのためには、ダミーの鍵を作成してから、そのダミーの鍵を削除します。

以下のコマンドを使用できます。

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

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

11. 認証局の公開証明書をトラストストアにインポートします。

以下のコマンドを使用できます。

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

12. Connect:Direct ブリッジ・エージェントのプロパティ・ファイルを編集します。
ファイルの任意の場所に以下の行を組み込みます。

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

この手順の例では、*protocol* は使用するプロトコル (SSL または TLS) で、*password* はトラストストアの作成時に指定したパスワードです。

13. 双方向認証を設定する場合は、Connect:Direct ブリッジ・エージェントの鍵と証明書を作成します。
- a) 鍵ストアと鍵を作成します。
以下のコマンドを使用できます。

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

- b) 署名要求を生成します。
以下のコマンドを使用できます。

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

- c) 前の手順で受け取った証明書を鍵ストアにインポートします。証明書は、x.509 形式でなければなりません。
以下のコマンドを使用できます。

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

- d) Connect:Direct ブリッジ・エージェントのプロパティ・ファイルを編集します。
ファイルの任意の場所に以下の行を組み込みます。

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

この手順の例では、*password* は鍵ストアの作成時に指定したパスワードです。

関連タスク

143 ページの『Connect:Direct ブリッジの構成』

Managed File Transfer ネットワークと Connect:Direct ネットワークの間でファイルを転送するために、Connect:Direct ブリッジを構成します。Connect:Direct ブリッジのコンポーネントは、Connect:Direct ノードと、そのノードと通信するための専用の Managed File Transfer エージェントです。このエージェントのことを Connect:Direct ブリッジ・エージェントといいます。

ConnectDirectProcessDefinition.xml ファイルを使用して、開始する Connect:Direct プロセスを指定する操作

Managed File Transfer 転送の一部として開始する Connect:Direct プロセスを指定します。Managed File Transfer には、プロセス定義を指定するために編集できる XML ファイルが用意されています。

このタスクについて

fteCreateCDAgent コマンドを使用すると、エージェントの構成ディレクトリー `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name` にファイル `ConnectDirectProcessDefinitions.xml` が作成されます。Connect:Direct ブリッジ・エージェントからユーザー定義 Connect:Direct プロセスを呼び出すには、まずこのファイルを編集してプロセス定義をセットアップする必要があります。

Connect:Direct ブリッジを経由した転送の一部として呼び出すように指定するプロセスごとに、以下の手順を実行します。

手順

1. 転送の一部として Connect:Direct ブリッジ・エージェントから呼び出す Connect:Direct プロセスを定義し、プロセス・テンプレートをファイルに保存します。
2. テキスト・エディターで `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name/ConnectDirectProcessDefinitions.xml` ファイルを開きます。
3. `<processSet>` エlementを作成します。
4. `<processSet>` Elementの内部に、`<condition>` Elementを作成します。
5. `<condition>` Element内で、ステップ1で定義した Connect:Direct プロセスを呼び出すために転送要求が一致する必要がある条件を定義するElementを1つ以上作成します。これらのElementは、`<match>` Elementまたは`<defined>` Elementのいずれかです。

- `<match>` Elementを使用して、変数の値がパターンに一致する必要があることを指定します。以下の属性を使用して`<match>` Elementを作成します。
 - `variable` - 値を比較する変数の名前。この変数は、組み込みシンボルです。詳しくは、[660 ページの『ユーザー定義 Connect:Direct プロセスで使用する置換変数』](#)を参照してください。
 - `value` - 指定した変数の値と比較するパターン。
 - オプション: `pattern - value` 属性の値で使用するパターンのタイプ。このパターン・タイプは、`wildcard` または `regex` のいずれかになります。この属性は任意指定であり、デフォルトは `wildcard` です。
- `<defined>` Elementを使用して、変数に値を定義する必要があることを指定します。以下の属性を使用して`<defined>` Elementを作成します。
 - `variable` - 値が定義されていなければならない変数の名前。この変数は、組み込みシンボルです。詳しくは、[660 ページの『ユーザー定義 Connect:Direct プロセスで使用する置換変数』](#)を参照してください。

`<condition>` Element内に指定された条件は、論理 AND と結合されます。Connect:Direct ブリッジ・エージェントがこの `<processSet>` Elementによって指定されたプロセスを呼び出すには、すべての条件を満たす必要があります。`<condition>` Elementを指定しない場合、プロセス・セットはすべての転送に一致します。

6. `<processSet>` Elementの内部に、`<process>` Elementを作成します。
7. `<process>` Elementの内部に、`<transfer>` Elementを作成します。

`transfer` Elementでは、Connect:Direct ブリッジ・エージェントが転送の一部として呼び出す Connect:Direct プロセスを指定します。以下の属性を使用して`<transfer>` Elementを作成します。

- `process` - -ステップ1で定義した Connect:Direct プロセスの場所です。このファイルの場所は、絶対パスで指定されます。または、`MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name` ディレクトリーに対する相対パスで指定します。

タスクの結果

Connect:Direct ブリッジ・エージェントは、条件に合致する項目を検索するときに、ファイルの先頭から末尾に向かって検索します。最初に見つかった一致が使用されます。

関連タスク

[143 ページの『Connect:Direct ブリッジの構成』](#)

Managed File Transfer ネットワークと Connect:Direct ネットワークの間でファイルを転送するために、Connect:Direct ブリッジを構成します。Connect:Direct ブリッジのコンポーネントは、Connect:Direct ノードと、そのノードと通信するための専用の Managed File Transfer エージェントです。このエージェントのことを Connect:Direct ブリッジ・エージェントといいます。

関連資料

[590 ページの『Connect:Direct プロセスの定義ファイルのフォーマット』](#)

Connect:Direct ブリッジ・エージェントの構成ディレクトリーにある

ConnectDirectProcessDefinitions.xml ファイルでは、ファイル転送の一部として開始するユーザー定義 Connect:Direct プロセスを指定します。

[402 ページの『fteCreateCDAgent \(Connect:Direct ブリッジ・エージェントの作成\)』](#)

fteCreateCDAgent コマンドは、Managed File Transfer Agent、および Connect:Direct ブリッジのための関連する構成を作成します。

MSCS による MFT エージェントの構成

MFT でサポートされるプラットフォームで、Windows のいずれかのバージョンを実行している場合は、Managed File Transfer (MFT) エージェントの Microsoft Cluster Service (MSCS) セットアップが可能です。

このタスクについて

このタスクでは、MFT エージェントのフェイルオーバーを実現できる 2 つのシナリオについて説明します。

- シナリオ 1: エージェントを MSCS リソースとして構成する。
- シナリオ 2: エージェント・キュー・マネージャーとエージェントを MSCS リソースとして構成する。

手順

シナリオ 1: エージェントを MSCS リソースとして構成する

- エージェントを MSCS リソースとして構成するには、以下の手順を実行します。
 - a) クラスタ内の各マシンで Managed File Transfer をローカルにインストールします。
[Managed File Transfer のインストール](#)を参照してください。
 - b) クラスタ内の 1 次マシン上にエージェントを作成します。
このエージェントは、CLIENT トランスポートを使用してエージェント・キュー・マネージャーに接続するように構成する必要があります。キュー・マネージャー上にこのエージェントのためのすべてのオブジェクトを作成してください。これを行う方法については、[エージェントのセットアップ](#)を参照してください。
 - c) Windows サービスとして実行するようにエージェントを変更し、Windows の「サービス」ツールでエージェント・サービスの「**スタートアップの種類**」フィールドを「手動」に設定して、Windows の再始動時にエージェントを自動開始しないように構成します。
詳しくは、[Windows サービスとしての MFT エージェントの開始](#)を参照してください。
 - d) 二次マシン上で、シナリオ 1 のステップ [153 ページの『2』](#) およびステップ [153 ページの『3』](#) を繰り返します。
これにより、クラスタ内のもう一方のマシンにも、ログやプロパティなどのためのファイル構造が存在するようになります。手順 [153 ページの『2』](#) のキュー・マネージャー・オブジェクトの作成は行う必要がないので注意してください。
 - e) 1 次マシンで、エージェントを「汎用サービス」として MSCS の制御対象に追加します。
そのためには、次のようにします。
 - a. クラスタを右クリックし、「役割」->「リソースの追加」->「汎用サービス」を選択します。

- b. Windows サービスのリストから、エージェント・サービスを選択し、「次へ」をクリックして構成ウィザードを実行します。

これで、エージェント・サービスが MSCS リソースとして追加されました。フェイルオーバーが発生した場合は、もう一方のマシン上でエージェント・サービスが開始されます。

シナリオ 2: エージェント・キュー・マネージャーとエージェントを MSCS リソースとして構成する

- エージェント・キュー・マネージャーおよびエージェントを MSCS リソースとして構成するには、以下の手順を実行します。
 - a) MSCS リソースとして実行するようにエージェント・キュー・マネージャーを構成します。

その方法について詳しくは、[MSCS 制御下にキュー・マネージャーを置く](#)を参照してください。
 - b) クラスタ内の 1 次マシン上にエージェントを作成します。

このエージェントは、BINDINGS トランSPORTを使用してエージェント・キュー・マネージャーに接続するように構成する必要があります。キュー・マネージャー上にこのエージェントのためのすべてのオブジェクトを作成してください。これを行う方法については、[エージェントのセットアップ](#)を参照してください。
 - c) Windows サービスとして実行するようにエージェントを変更し、Windows の「サービス」ツールでエージェント・サービスの「**スタートアップの種類**」フィールドを「手動」に設定して、Windows の再始動時にエージェントを自動開始しないように構成します。

詳しくは、[Windows サービスとしての MFT エージェントの開始](#)を参照してください。
 - d) (MSCS 制御下にある) エージェント・キュー・マネージャーが 2 次マシンで実行されていることを確認します。

このマシンに作成するエージェントは、BINDINGS トランSPORTを使用してキュー・マネージャーに接続するので、エージェントの作成時にキュー・マネージャーが使用可能でなければなりません。
 - e) 二次マシン上で、シナリオ 2 のステップ [154 ページの『2』](#) およびステップ [154 ページの『3』](#) を繰り返します。

これにより、クラスタ内のもう一方のマシンにも、ログやプロパティなどのためのファイル構造が存在するようになります。手順 [154 ページの『2』](#) のキュー・マネージャー・オブジェクトの作成は行う必要がないので注意してください。
 - f) エージェントを「汎用サービス」として MSCS の制御対象に追加します。

そのためには、次のようにします。
 - a. クラスタを右クリックし、「役割」->「リソースの追加」->「汎用サービス」を選択します。
 - b. Windows サービスのリストから、エージェント・サービスを選択し、「次へ」をクリックして構成ウィザードを実行します。
 - g) エージェント・サービスのリソース・プロパティを変更して、依存関係リストにキュー・マネージャー・リソースを追加します。

これにより、エージェントが開始される前にキュー・マネージャー・リソースが開始されるようになります。
 - h) キュー・マネージャー・リソースをオフラインにしてから、エージェント・リソースをオンラインにします。キュー・マネージャー・リソースとエージェントの両方が開始されていることを確認してください。

フェイルオーバーが発生した場合は、エージェント・サービスとエージェント・キュー・マネージャーが 2 次マシン上で開始されます。

管理 MFT

Managed File Transfer を管理するには、Managed File Transfer コマンドを使用します。また、IBM MQ Explorer を使用して管理用タスクの一部を行うこともできます。

エージェント・コマンド・キューにメッセージを入れて転送を開始する

ファイル転送メッセージをソース・エージェントのコマンド・キューに書き込むことにより、ファイル転送を開始することもできます。例示コマンド・キュー名は SYSTEM.FTE.COMMAND.AGENT01 です。正し

いソース・エージェントのコマンド・キューにメッセージが届くようにする必要があります。XML のソース情報と一致しないエージェントによってメッセージが受け取られる場合、メッセージは拒否されます。

転送要求 XML は、FileTransfer.xsd スキーマに準拠している必要があります。ルート・エレメントとして <request> エレメントを使用する必要があります。転送要求メッセージの構造と内容に関する情報については、『ファイル転送要求メッセージ・フォーマット』を参照してください。エージェントのコマンド・キューにどのように転送要求メッセージを書き込むかは、タスクにより異なります。例えば、IBM MQ Java API を使用して、メッセージをキューにプログラマチックに書き込むことができます。

MFT エージェントの開始

Managed File Transfer エージェントを使用してファイル転送を行うには、まずエージェントを開始する必要があります。

このタスクについて

Managed File Transfer Agent は、コマンド行から開始できます。この場合、エージェント・プロセスはユーザーがシステムからログオフすると停止します。

ULW UNIX, Linux, and Windows では、ユーザーがシステムからログオフしてもエージェントが実行を続行し、ファイル転送を受信し続けるようにエージェントを構成することができます。

z/OS z/OS では、対話式セッションがなくても、JCL から開始したタスクとしてエージェントを開始するようにエージェントを構成できます。

エージェントを実行中にリカバリー不能エラーが発生した場合、初期障害データ・キャプチャー (FDC) が生成され、エージェントは停止することにご注意ください。

手順

- コマンド行からエージェントを開始するには、**fteStartAgent** コマンドを使用します。
詳細については [528 ページの『fteStartAgent』](#) を参照してください。
- **ULW**
システムからログオフしてもエージェントが実行を続行するように構成するには、次のようにします。
 - **Windows** Windows では、エージェントを Windows サービスとして実行するように構成します。
詳細については [155 ページの『Windows サービスとしての MFT エージェントの開始』](#) を参照してください。
 - **Linux** **UNIX** UNIX および Linux では、リブート時にスクリプト・ファイルを使用してエージェントが自動的に開始するように構成します。詳細については [157 ページの『UNIX および Linux システム始動時の MFT エージェントの開始』](#) を参照してください。
- **z/OS**
z/OS では、対話式セッションがなくても、JCL から開始したタスクとしてエージェントを開始するようにエージェントを構成します。
詳細については [159 ページの『z/OS での MFT エージェントの開始』](#) を参照してください。

Windows Windows サービスとしての MFT エージェントの開始

エージェントを Windows サービスとして開始することにより、Windows からログオフしても、引き続きエージェントを実行し、ファイル転送を受け取ることができます。

このタスクについて

Windows 上のコマンド行からエージェントを開始すると、エージェント・プロセスは、Windows にログオンするために使用したユーザー名を使用して実行されます。システムからログオフすると、エージェント・プロセスは停止します。エージェントが停止しないようにするには、Windows サービスとして実行さ

れるようにエージェントを構成することができます。Windows サービスとして実行させることにより、エージェントを、Windows 環境の始動または再始動時に自動的に開始するように構成することもできます。

以下の手順に従って、Windows サービスとして実行するエージェントを開始します。Windows サービスとしてエージェントを実行するには、サポートされる Windows バージョンのいずれかで、Managed File Transfer を実行している必要があります。サポートされる環境のリストについては、『[IBM MQ のシステム要件](#)』を参照してください。

実際のステップは、既にエージェントを作成しているか、あるいはエージェントを作成中であるかによって異なります。どちらのオプションも以下のステップで説明されています。

手順

1. Managed File Transfer エージェントを作成する場合は、**fteCreateAgent**、**fteCreateCDAgent**、または **fteCreateBridgeAgent** コマンドを使用します。Windows サービスとしてエージェントを実行するには、**-s** パラメーターを指定します。以下の例では、エージェント・キュー・マネージャー QMGR1 を含むエージェント AGENT1 が作成されます。Windows サービスは、関連パスワード ftepassword を含む、ユーザー名 fteuser を使用して実行されます。

```
fteCreateAgent -agentName AGENT1 -agentQMGR1 QMGR1 -s -su fteuser -sp ftepassword
```

オプションで、**-s** パラメーターの後にサービスの名前を指定することができます。名前を指定しなかった場合、サービスの名前は mqmftAgentAGENTQMGR1 となります。ここで、AGENT はユーザーが指定したエージェント名であり、QMGR はエージェント・キュー・マネージャー名です。この例では、サービスのデフォルト名は mqmftAgentAGENT1QMGR1 です。

注: **-su** パラメーターを使用して指定する Windows ユーザー・アカウントには、**Log on as a service** 権限が必要です。これを構成する方法については、[325 ページ](#)の『[MFT エージェントまたはロガーを Windows サービスとして実行するためのガイダンス](#)』を参照してください。

詳しくは、[388 ページ](#)の『[fteCreateAgent \(MFT エージェントの作成\)](#)』、[-->402 ページ](#)の『[fteCreateCDAgent \(Connect:Direct ブリッジ・エージェントの作成\)](#)』、または [393 ページ](#)の『[fteCreateBridgeAgent \(MFT プロトコル・ブリッジ・エージェントの作成および構成\)](#)』を参照してください。

2. 前のステップに従ってエージェントを作成した場合は、**fteCreateAgent**、**fteCreateCDAgent**、または **fteCreateBridgeAgent** コマンドによって生成された MQSC コマンドを実行します。これらのコマンドは、エージェントが必要とする IBM MQ キューを作成します。

例えば、エージェントの名前が AGENT1、エージェント・キュー・マネージャーの名前が QMGR1、および調整キュー・マネージャーの名前が COORDQMGR1 の場合、以下のコマンドを実行します。

```
runmqsc QMGR1 MQ_DATA_PATH\mqft\config\COORDQMGR1\agents\AGENT1\AGENT1_create.mqsc
```

3. 前のステップでエージェントを作成しておらず、代わりに既存のエージェントを Windows サービスとして実行するように構成する場合、エージェントが実行中であれば最初にエージェントを停止してから、その構成を変更します。

- a) 以下の例では、AGENT1 という名前のエージェントを使用します。以下のコマンドを実行します。

```
fteStopAgent AGENT1
```

- b) **fteModifyAgent** コマンドを使用して、Windows サービスとして実行されるようにエージェントを構成します。

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

詳しくは、[490 ページ](#)の『[fteModifyAgent \(Windows サービスとしての MFT エージェントの実行\)](#)』を参照してください。

4. **fteStartAgent** コマンドを使用してエージェントを開始します。代わりに、Windows デスクトップのスタート・メニューから選択した「コントロールパネル」の「管理ツール」から選択可能な Windows の「サービス」ツールを使用して、サービスを開始することもできます。

```
fteStartAgent AGENT1
```

Windows からログオフしても、サービスは引き続き実行されます。Windows がシャットダウン後に再始動したときにサービスも再開されるようにするため、Windows の「サービス」ツールの「スタートアップの種類」フィールドはデフォルトで「自動」に設定されています。Windows の再始動時にサービスが再開されないようにするには、これを「手動」に変更します。

5. オプション: エージェントを停止するには、**fteStopAgent** コマンドを使用するか、あるいは Windows の「サービス」ツールを使用します。例えば、コマンド行から、以下のコマンドを実行します。

```
fteStopAgent AGENT1
```

- **fteStopAgent** コマンドをサービスとして実行すると、このコマンドは **-i** パラメーターが指定されているかどうかに関わりなく、常にこのパラメーターを使用して実行されます。**-i** パラメーターは、進行中の転送を完了せずにエージェントを即時停止します。これは、Windows サービスの制限によるものです。

次のタスク

Windows サービスの開始に問題がある場合は、325 ページの『MFT エージェントまたはロガーを Windows サービスとして実行するためのガイダンス』を参照してください。このトピックでは、Windows サービス・ログ・ファイルの場所についても説明します。

Linux UNIX UNIX および Linux システム始動時の MFT エージェントの開始

Managed File Transfer Agent は、UNIX のシステム始動時に開始するように構成できます。UNIX システムからログオフしても、エージェントは引き続き実行され、ファイル転送を受け取ることができます。

fteCreateAgent、**fteCreateCDAgent**、または **fteCreateBridgeAgent** のいずれかの Managed File Transfer コマンドを使用してエージェントを作成および構成した場合、単純に実行するスクリプト・ファイルを使用して、UNIX マシンでのリブート中に自動的に開始するようにエージェントを構成できます。

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

`mq_install_root` は、必要な Managed File Transfer インストール済み環境のルート・ディレクトリーです。デフォルトは次のとおりです。/opt/mqm および エージェント名は、開始される Managed File Transfer Agent の名前です。このスクリプト・ファイルの使用法は、具体的な UNIX オペレーティング・システムに応じて異なります。

Solaris

Solaris Solaris の場合は、以下の手順を実行します。

1. 次のコマンドを実行し、返されるパスを記録します。

```
which sh
```

例えば、パスが /usr/bin/sh であるとします。

2. システム root ユーザー ID として、独自の /etc/init.d/startmqmft ファイルを作成します。
3. このファイルを編集し、次のスクリプト行を追加します。このとき、手順 1 で返されたパスをスクリプトの 1 行目として使用します。

```
#!/usr/bin/sh
su mqmft_user mq_install_root/bin/fteStartAgent agent_name
```

4. ファイル実行可能モジュールを作成します。例えば、次のようにします。

```
chmod 755 /etc/init.d/startmqmft
```

5. このファイルのシンボリック・リンクを rc3.d ディレクトリーに作成します。

```
ln -s /etc/init.d/startmqmft /etc/rc3.d/S98startmqmft
```

接頭部 S は、デフォルト状態の「Solaris 用に開始」を意味します。98 はシーケンス番号です。接尾部は `init.d` にあるファイル名です。

HP-UX

HP-UX

HP-UX の場合は、以下の手順を実行します。

1. 次のような内容の `/sbin/init.d/mqmft` という名前のファイルを作成します。

```
#!/bin/sh
su -l mqmft_user -c mq_install_root/bin/fteStartAgent agent_name
```

2. 次のような内容の `/etc/rc.config.d/mqmft` という名前のファイルを作成します。

```
MQMFT=1
```

3. このファイルのシンボリック・リンクを rc3.d ディレクトリーに作成して、エージェントを開始します。

```
ln -s /sbin/init.d/mqmft /sbin/rc3.d/S84mqmft
```

Linux

Linux

Linux システムの場合、システム・ブート・プロセス中にアプリケーションを開始する方法は複数あります。一般に、以下の手順の実行を検討してください。

1. 次のような内容の `/etc/rc.mqmft` という名前のファイルを作成します。

```
#!/bin/sh
su -l mqmft_user -c mq_install_root/bin/fteStartAgent agent_name"
```

ここで、`mqmft_user` はエージェント・プロセスを実行するユーザー ID です。このユーザー ID は `mqm` グループのメンバーである必要があります。

2. ファイル実行可能モジュールを作成します。例えば、次のようにします。

```
chmod 755 /etc/rc.mqmft
```

3. 次に、`/etc/inittab` に、次の行を追加します。

```
mqmft:5:boot:/etc/rc.mqmft
```

Linux でのブート時にエージェントを開始するその他の方法では、`/etc/rc.d/rc.local` ファイルにスクリプト行を追加するか、Linux SuSe にスクリプト行を追加して、スクリプト行を `/etc/init.d/`

boot.local ファイルに追加します。ご使用の環境に最も適した方法を選択してください。サポートされている特定の Linux ディストリビューションで始動時にエージェントを開始するその他の方法について、以下にさらに説明します。

SLES 10 および 11

SUSE Linux Enterprise Server (SLES) 10 および 11 システムの場合は、以下の手順を実行します。

1. システム root ユーザー ID として、独自の /etc/init.d/rc.rclocal ファイルを作成します。
2. rc.rclocal ファイルに次の行を追加します。

```
#!/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. 以下のコマンドを実行します。

```
chmod 755 rc.rclocal
chkconfig --add rc.rclocal
```

z/OS z/OS での MFT エージェントの開始

z/OS でエージェントを実行するには、UNIX System Services セッションから **fteStartAgent** コマンドを実行する方法に加えて、JCL から開始済みタスクとして開始する方法もあり、この場合には対話式セッションを必要としません。

開始タスクは、特定のユーザー ID の下で実行され、ユーザーのログオフに影響されないため、使用されます。

注: **V9.0.0.1** → **V9.0.2** 通常、開始済みタスクはログオン特権がない可能性がある管理ユーザーの下で実行されるため、エージェントを実行しているユーザーとして z/OS システムにログオンすることはできません。そのエージェントに対して、**fteStartAgent**、**fteStopAgent**、および **fteSetAgentTraceLevel** の各コマンド、および **-d** パラメーターが指定されている **fteShowAgentDetails** コマンドを実行することはできません。

IBM MQ 9.0.2 および IBM MQ 9.0.0 Fix Pack 1 以降、z/OS 上の Managed File Transfer エージェントでエージェント・プロパティー **adminGroup** を使用できるようになりました。例えば、MFTADMIN というセキュリティ・マネージャー・グループを定義し、開始済みタスクのユーザー ID と管理者 TSO ID をこのグループに追加できます。エージェント・プロパティー・ファイルを編集し、**adminGroup** プロパティーの値をこのセキュリティ・マネージャー・グループの名前に設定します。

```
adminGroup=MFTADMIN
```

これにより、このグループのメンバーは、開始済みタスクとして実行されているエージェントに対して、**fteStartAgent**、**fteStopAgent**、**fteSetAgentTraceLevel** の各コマンド、および **-d** パラメーターが指定されている **fteShowAgentDetails** コマンドを実行できます。

詳しくは、**adminGroup** プロパティー (550 ページの『MFT agent.properties ファイル』) を参照してください。

エージェントは、Java アプリケーションであり、JCL から実行できる UNIX System Services アプリケーションです。これを実行するには、エージェントに対して生成された Managed File Transfer コマンド PDSE ライブラリー・データ・セットから、BFGAGSTP メンバーを使用します。MFT コマンド PDSE ライブラリ

ー・データ・セットを作成し、必要なエージェント用にカスタマイズする方法について詳しくは、[63 ページの『MFT エージェントまたはロガーのコマンド・データ・セットの作成』](#)を参照してください。

IBM MQ Advanced for z/OS, Value Unit Edition のみでエージェントのリモート接続を有効にする

V 9.0.3

MQ Adv. VUE

IBM MQ Advanced for z/OS VUE の製品 ID (PID) で実行されている、z/OS 上の MFT エージェントは、クライアント接続を使用して z/OS 上のキュー・マネージャーに接続できます。

エージェントが始動されると、BFGPR0137I メッセージがそのイベント・ログ (output0.log) に書き込まれ、実行時に使用されている PID が表示されます。このメッセージの例を以下に示します。

BFGPR0137I: 製品使用データ記録が、製品 'MQ z/OS MFT'、製品 ID '5655-MF9' に対して開始されました。

IBM MQ 製品、各製品に関連付けられている PID 値、およびエクスポート分類について詳しくは、[IBM MQ 製品 ID およびエクスポート情報](#)を参照してください。

他の PID の下で実行されている z/OS 上の MFT エージェントは、バインディング接続のみを使用して、ローカル・キュー・マネージャーに接続できます。

z/OS 上の MFT エージェントは、MFT PID に関係なく、z/OS 上で実行されるキュー・マネージャーにしか接続できません。

z/OS で実行されていないキュー・マネージャーに IBM MQ Advanced for z/OS VUE エージェントが接続しようとする、次の BFGMQ1044E メッセージが発行され、エージェントの開始が終了します。

BFGMQ1044E: z/OS でのエージェント・クライアント接続は、z/OS 上のキュー・マネージャーに対する接続でなければなりません。

関連資料

[212 ページの『z/OS 上の MFT エージェントの停止』](#)

JCL からの開始タスクとして z/OS 上で Managed File Transfer Agent を実行している場合、エージェントは、**fteStopAgent** コマンドに加えて、z/OS オペレーター・コマンド **MODIFY** および **STOP** を受け入れます。

[550 ページの『MFT agent.properties ファイル』](#)

各 Managed File Transfer Agent には、独自のプロパティ・ファイル **agent.properties** があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。agent.properties ファイルには、エージェントの動作を変更するプロパティを含めることもできます。

新規ファイル転送の開始

新規ファイル転送は、IBM MQ Explorer またはコマンド行から開始でき、単一ファイルまたは複数ファイルのグループのいずれかの転送を選択できます。

このタスクについて

新規ファイル転送をコマンド行から開始するには、[fteCreateTransfer](#) コマンドを参照してください。

新規ファイル転送を IBM MQ Explorer の「[新しいファイル転送管理の作成](#)」ウィザードを使用して開始するには、以下のステップを実行します。

手順

1. 「ナビゲーター」ビューで、「**ファイル転送管理**」をクリックします。「**ファイル転送管理 - メイン**」が「コンテンツ」ビューに表示されます。
2. すべての調整キュー・マネージャーが「ナビゲーター」ビューに表示されます。転送に使用するエージェントの登録対象となる調整キュー・マネージャーの名前を展開します。転送に使用するつもりのもの以外の調整キュー・マネージャーに現在接続している場合は、「ナビゲーター」ビューでその調整キュー・マネージャーの名前を右クリックして、「**切断**」をクリックします。使用する調整キュー・マネージャーの名前を右クリックして、「**接続**」をクリックします。

- 以下の方式のいずれかを使用して、「ファイル転送管理の新規作成」ウィザードを開始します。
 - 「ナビゲーター」ビューで、関連した調整キュー・マネージャー、「転送テンプレート」、「転送ログ」、または「保留中の転送」のいずれかのノードの名前を右クリックします。その後「新規の転送」をクリックしてウィザードを起動します。
 - 「ファイル」 > 「新規」 > 「その他」 > 「ファイル転送管理ウィザード」 > 「新規の転送ウィザード」をクリックします。
- ウィザード・パネルの指示に従います。各パネルには、コンテキスト・ヘルプも提供されています。Windows 上でコンテキスト・ヘルプにアクセスするには、F1 キーを押します。Linux 上では、Ctrl+F1 キーまたは Shift+F1 キーを押します。

転送定義ファイルの使用

ファイル転送を作成するために使用できる転送定義ファイルを指定できます。転送定義ファイルは、転送を作成するために必要な情報の一部またはすべてを定義した XML ファイルです。

転送定義ファイルは、複数のソース・ファイルと宛先ファイルを 1 つの転送操作に指定する際に便利です。転送定義ファイルを使用して、複雑なファイル転送を実行依頼できます。転送定義ファイルを再利用したり共有したりすることも可能です。

転送定義ファイルでは、2 つのフォーマットを使用できます。多少の違いはありますが、どちらのフォーマットも FileTransfer.xsd スキーマに準拠しています。このスキーマは、Managed File Transfer インストール済み環境の samples\schema ディレクトリーで見つけることができます。

以下の 2 つのフォーマットの転送定義ファイルがサポートされています。

- 転送のソース・ファイルと宛先ファイルの定義。この定義では、ルートとして <transferSpecifications> エレメントを使用します。
- 転送全体の定義。ソース・ファイルと宛先ファイル、ソース・エージェントと宛先エージェントを含みます。この定義では、ルートとして <request> エレメントを使用します。
 - このフォーマットのファイルは、**fteCreateTransfer** コマンドの **-gt** パラメーターを使用して生成できます。

転送のソース・ファイルと宛先ファイルだけを指定する転送定義ファイル・フォーマットの例を以下に示します。

```
<?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>
```

このフォーマットの転送定義ファイルを実行依頼する場合は、コマンド行でソース・エージェントと宛先エージェントを指定する必要があります。

```
fteCreateTransfer -sa AGENT1 -sm agent1qm -da AGENT2 -dm agent2qm -td
c:\definitions\example1.xml
```

転送に必要なすべての情報を指定する転送定義ファイル・フォーマットの例を以下に示します。

```
<?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>

```

fteCreateTransfer コマンドの **-gt** パラメーターを使用して、このフォーマットのファイルを作成できます。このフォーマットの転送定義ファイルを実行依頼する場合は、コマンド行で他の情報を指定する必要はありません。

```
fteCreateTransfer -td c:\definitions\example2.xml
```

コマンド行でソース・エージェントと宛先エージェントの情報をオーバーライドすることもできます。その場合は、転送定義ファイルに加えて通常のパラメーターを渡します。以下に例を示します。

```
fteCreateTransfer -da AGENT9 -dm agent9qm -td c:\definitions\example2.xml
```

この例では、コマンド行オプションを使用して、転送定義ファイル内で **AGENT9** として定義されている宛先エージェントと、転送定義ファイルで **agent9qm** として定義されている宛先キュー・マネージャーをオーバーライドします。

上記で説明したどちらのフォーマットでも、1つ以上の `<item>` エレメントを使用できます。`<item>` エレメントについて詳しくは、[ファイル転送要求メッセージ・フォーマット](#)を参照してください。これらの転送項目には、転送の動作を制御する追加属性を持つソース・ファイルと宛先ファイルのペアがそれぞれ定義されます。例えば、以下の動作を指定します。

- 転送はチェックサムを使用するかどうか
- 転送はテキストかバイナリーか
- 転送が完了した後にソース・ファイルを削除するかどうか
- ファイルが存在する場合、宛先ファイルを上書きするかどうか

転送定義ファイルを使用することの1つの利点は、コマンド行からは使用できない追加のオプションを指定できることです。例えば、メッセージからファイルへの転送を行っているときに、転送定義ファイルを使用して `groupId` 属性を指定することができます。この属性は、キューから読み取られるメッセージの IBM MQ グループ ID を指定します。転送定義ファイルの別の利点は、ファイル・ペアごとに異なるオプションを指定できることです。例えば、チェックサムを使用するかどうか、またファイルをテキスト・モードで転送するかバイナリー・モードで転送するかを、個々のファイルごとに指定することができます。コマンド行を使用する場合には、転送に含まれる各ファイルに同じオプションが適用されます。

以下に例を示します。

```

<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 分散システムから z/OS システムにファイルを転送するための項目も使用できます。

```
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 この例では、ファイル `textTransferTest.txt` をソース・エージェントから、宛先エージェント上のデータ・セット `//TEXT.TRANS.TEST` にテキスト・モードで転送します。この転送によって、ソース・データが、ソース・エージェントのデフォルトのエンコード (ソース・エンコード属性の指定なし) から、コード・ページ `IBM-1047` に変換されます。

スケジュール済みファイル転送の作成

IBM MQ Explorer エクスプローラーまたはコマンド行のいずれかを使用して新規ファイル転送をスケジュールに入れられます。スケジュール済みの転送には、単一のファイルまたは1つのグループの複数のファイルを含めることができます。スケジュール済みファイル転送は、1回実行することも複数回転送を繰り返すこともできます。

このタスクについて

ファイル転送スケジュールは、1回行うようにセットアップすることもできますし、次の間隔で行うようにセットアップすることもできます。

- 1分ごと
- 毎時
- 日次
- 週次
- 月次
- 毎年

その後、次の時点でスケジュールの発生を停止するように指定できます。

- 定義された日時
- 定義された発生回数の後

また、期限なくスケジュールの発生が継続するように指定することもできます。

V 9.0.0.9 スケジュールされた転送が毎日同じ時刻に実行される場合は、エージェント・プロパティ・ファイルの `adjustScheduleTimeForDaylightSaving` 属性を使用して、クロックが変更されたときにスケジュールが実行される時刻を調整します。詳しくは、[MFT agent.properties ファイル](#)を参照してください。

コマンド・ラインを使用してスケジュール済みファイル転送を新規作成するには、`fteCreateTransfer` コマンドのスケジュールリング・パラメーター (`-tb`、`-ss`、`-oi`、`-of`、`-oc`、および `-es`) を使用します。

IBM MQ Explorer の「ファイル転送管理の新規作成」ウィザードを使用して、スケジュールされるファイル転送を新規作成するには、以下の手順を実行します。

手順

1. 「ナビゲーター」ビューで、「ファイル転送管理」をクリックします。「ファイル転送管理 - メイン」が「コンテンツ」ビューに表示されます。
2. すべての調整キュー・マネージャーが「ナビゲーター」ビューに表示されます。転送に使用するエージェントの登録対象となる調整キュー・マネージャーの名前を展開します。転送に使用するつもの以外の調整キュー・マネージャーに現在接続している場合は、「ナビゲーター」ビューでその調整キュー・マネージャーの名前を右クリックして、「切断」をクリックします。使用する調整キュー・マネージャーの名前を右クリックして、「接続」をクリックします。
3. 以下の方式のいずれかを使用して、「ファイル転送管理の新規作成」ウィザードを開始します。
 - a) 「ナビゲーター」ビューで、関連した調整キュー・マネージャー、「転送テンプレート」、「転送ログ」、または「保留中の転送」のいずれかのノードの名前を右クリックします。その後「新規の転送」をクリックしてウィザードを起動します。
 - b) 「ファイル」 > 「新規」 > 「その他」 > 「ファイル転送管理ウィザード」 > 「新規の転送ウィザード」をクリックします。
4. ウィザード・パネルの指示に従います。「スケジュール転送を有効にする」チェック・ボックスを選択してあることを確認し、「スケジュール」タブにスケジュールの詳細を入力します。スケジュール済みファイル転送は、転送に影響する可能性がある問題がなければ、スケジュール開始時刻から 1 分以内に開始します。例えば、スケジュール済み転送の開始を妨げるネットワークまたはエージェントの問題があるかもしれません。各パネルにはコンテキスト・ヘルプがあります。Windows でコンテキスト・ヘルプにアクセスするには、F1 を押します。Linux の場合は、Ctrl+F1 または Shift+F1 を押します。

タスクの結果

スケジュール済みファイル転送に含まれるメッセージの詳細については、『[スケジュール済み転送のメッセージ・フォーマット](#)』を参照してください。

保留中のファイル転送の処理

IBM MQ Explorer から、保留中のスケジュール済みファイル転送を表示することができます。「保留中の転送」ウィンドウには、現在接続している調整キュー・マネージャーに登録されている保留中の転送がすべて表示されます。

このタスクについて

まだ開始されていないスケジュール済みファイル転送の状況を表示するには、以下のステップを実行します。

手順

1. 「ナビゲーター」ビューで「ファイル転送管理」を展開します。「ファイル転送管理 - メイン」が「コンテンツ」ビューに表示されます。
2. すべての調整キュー・マネージャーが「ナビゲーター」ビューに表示されます。スケジュール済みの転送に使用した調整キュー・マネージャーの名前を展開します。接続先の調整キュー・マネージャーを変更する場合は、「ナビゲーター」ビューで使用する調整キュー・マネージャーの名前を右クリックして、「接続」をクリックします。
3. 「保留中の転送」をクリックします。「保留中の転送」ウィンドウが「コンテンツ」ビューに表示されます。
4. 「保留中の転送」ウィンドウに、スケジュール済みのファイル転送に関する以下の詳細が表示されます。
 - a) 「名前」。スケジュール済みファイル転送の番号。この番号は自動的に割り当てられます。
 - b) 「ソース」。ソース・エージェントの名前。
 - c) 「ソース・ファイル」。ホスト・システムにおける、転送するファイルの名前。

- d) 「宛先」。宛先エージェントの名前。
- e) 「宛先ファイル」。宛先システムに転送された後のファイルの名前。
- f) 「スケジュール済みの開始時刻 (選択したタイム・ゾーン)」。ファイル転送を開始するようスケジュールされた、管理者が選択したタイム・ゾーンでの時刻と日付。表示されるタイム・ゾーンを変更するには、**ウィンドウ > 設定 > IBM MQ Explorer > ファイル転送管理** をクリックし、**タイム・ゾーン**: リストから別のタイム・ゾーンを選択します。「OK」をクリックします。
- g) 「繰り返し周期」。スケジュール済み転送を繰り返す選択をした場合、転送を繰り返す指定間隔。数値で表示されます。
- h) 「繰り返しのタイプ」。スケジュール済み転送を繰り返す選択をした場合、ファイル転送のために指定した繰り返しの間隔のタイプ。タイプは、次のいずれかの値になります。分、時間、日、週、月、または年。
- i) 「繰り返し期限」。スケジュール済み転送を繰り返す選択をした場合、ファイル転送の繰り返しを停止する時間の詳細。例えば、指定した日時、指定した発生回数の後など。

タスクの結果

「保留中の転送」ウィンドウに表示されている内容を最新表示するには、「コンテンツ」ビューのツールバーにある「リフレッシュ」ボタン  をクリックします。

保留中のファイル転送を取り消すには、特定の転送を右クリックし、「キャンセル」をクリックします。転送を取り消すと、ファイル転送要求が完全に廃棄されます。

ファイル転送のトリガー

転送を実行するために満たす必要がある特定のトリガー条件を、ファイル転送に対して設定できます。トリガー条件が満たされない場合にはファイル転送は実行されず、転送が行われなかったことを記録するためのログ・メッセージがオプションで送信されます。その後ファイル転送要求は廃棄されます。例えば、ソース・エージェントがあるシステム上の指定ファイルが設定サイズを超えた場合のみ、またはソース・エージェントがあるシステム上に特定の指定ファイルが存在する場合のみファイル転送が実行されるようにセットアップできます。トリガー・ファイル転送は、IBM MQ Explorer かコマンド行のいずれかを使用してセットアップできます。

このタスクについて

リソースを継続的にモニターして、トリガー条件が満たされるかどうかを判断することができます。リソース・モニターの詳細については、[170 ページの『MFT リソースのモニター』](#)を参照してください。

設定できるトリガー条件は 3 種類あります。条件は以下のとおりです。

- ソース・エージェントと同じシステムに特定のファイルが存在する場合
- ソース・エージェントと同じシステムに特定のファイルが存在しない場合
- ソース・エージェントがあるシステム上の特定のファイルが特定のサイズを超えている (サイズはバイト、KB、MB、または GB で指定できます) 場合。これらの単位では、 2^{10} 規則を使用します。例えば 1 KB は 1024 バイトを示し、1 MB は 1024 KB を示します。

上記のリストにあるトリガー・タイプは、次の 2 つの方法で結合できます。

- 単一の条件では、ソース・エージェントがあるシステム上の複数のファイルを指定できます。この場合、指定したいいずれかのファイルが条件を満たした場合に (ブール演算子 OR) 転送がトリガーされます。
- 複数の条件を指定できます。この場合、条件すべてが満たされた場合のみ (ブール演算子 AND) 転送はトリガーされます。

トリガー転送をスケジュール済み転送と結合させることもできます。詳しくは、[スケジュール済みファイル転送の作成](#)を参照してください。この場合、トリガー条件はスケジュールが開始する時点で評価されます。繰り返しスケジュールの場合には、スケジュールが開始する時点ごとに評価されます。

トリガー転送は、プロトコル・ブリッジ・エージェントではサポートされません。

コマンドラインを使用してトリガーしたファイル転送を作成するには、`fteCreateTransfer` コマンドで `-tr` パラメーターを使用します。

IBM MQ Explorer の「ファイル転送管理の新規作成」ウィザードを使用して、スケジュールされるファイル転送を作成するには、以下の手順を実行します。

手順

1. 「ナビゲーター」ビューで、「ファイル転送管理」をクリックします。「ファイル転送管理 - メイン」が「コンテンツ」ビューに表示されます。
2. すべての調整キュー・マネージャーが「ナビゲーター」ビューに表示されます。スケジュール済みの転送に使用した調整キュー・マネージャーの名前を展開します。接続先の調整キュー・マネージャーを変更する場合は、「ナビゲーター」ビューで使用する調整キュー・マネージャーの名前を右クリックして、「接続」をクリックします。
3. 以下の方式のいずれかを使用して、「ファイル転送管理の新規作成」ウィザードを開始します。
 - a) 「ナビゲーター」ビューで、関連した調整キュー・マネージャー、「転送テンプレート」、「転送ログ」、または「保留中の転送」のいずれかのノードの名前を右クリックします。その後「新規の転送」をクリックしてウィザードを開きます。
 - b) 「ファイル」 > 「新規」 > 「その他」 > 「ファイル転送管理ウィザード」 > 「新規の転送ウィザード」をクリックします。
4. ウィザード・パネルの指示に従います。「トリガー」タブの「トリガー転送を有効にする」チェックボックスが選択されていることを確認し、そのタブにあるフィールドすべてに入力してトリガーをセットアップします。各パネルにはコンテキスト・ヘルプがあります。Windows 上でコンテキスト・ヘルプにアクセスするには、F1 キーを押します。Linux 上では、**Ctrl+F1** キーまたは **Shift+F1** キーを押します。

進行中のファイル転送のモニター

IBM MQ Explorer の「ファイル転送管理 - 現在の転送進行状況」タブを使用して、進行中のファイル転送をモニターできます。このファイル転送は、IBM MQ Explorer またはコマンド行のいずれかから開始できます。このタブには、スケジュール済み転送が開始した時点でのスケジュール済み転送の進行も表示されます。

このタスクについて

IBM MQ Explorer を使用してリモート・システムの調整キュー・マネージャーに関連する転送をモニターする場合は、167 ページの『リモート調整キュー・マネージャーをモニターするための IBM MQ Explorer の構成』のトピックにある手順に従ってください。

直前のファイル転送情報は、IBM MQ Explorer を停止して再始動した後は保持されません。再始動すると、過去の転送に関する情報は「現在の転送進行状況」タブから消去されます。完了した転送は、IBM MQ Explorer が開いているときには「完了した転送を削除」を使用して任意の時点で消去できます。

手順

IBM MQ Explorer またはコマンド行を使用して新規のファイル転送を開始した後、「現在の転送進行状況」タブで転送の進行をモニターできます。進行中の各転送について、以下の情報が表示されます。

- a) **ソース**。ソース・システムからファイルを転送するために使用するエージェントの名前。
- b) **宛先**。宛先システムでファイルを受け取るために使用するエージェントの名前。
- c) **現在のファイル**。現在転送中のファイルの名前。既に転送されている個々のファイルの部分は、B、KiB、MiB に表示されます。GiB または TiB とともに、ファイルの合計サイズが括弧内にあります。表示される単位はファイルのサイズによって異なります。

B は 1 秒あたりのバイト数を示します。KiB/s は 1 秒あたりのキビバイト数を示します (1 キビバイトは 1024 バイト)。MiB/s は 1 秒あたりのメビバイト数を示します (1 メビバイトは 1 048 576 バイト)。GiB/s は 1 秒あたりのギビバイト数を示します (1 ギビバイトは 1 073 741 824 バイト)。TiB/s は 1 秒あたりのテビバイト数を示します (1 テビバイトは 1 099 511 627 776 バイト)。

- d) 「**ファイル数**」。複数のファイルを転送している場合、この数は、ファイルのグループ全体を通して現在の転送がどの程度進んだかを示します。
- e) 「**進行状況**」。進行状況表示バーには、現在のファイル転送の完了率(パーセント)が表示されます。
- f) 「**転送速度**」。ファイルが転送される速度。KiB/s 単位 (1 秒あたりのキビバイト数。1 キビバイトは 1024 バイト)。
- g) 「**開始 (選択したタイム・ゾーン)**」。ファイル転送が開始された時刻。管理者が選択したタイム・ゾーンで表示されます。表示されるタイム・ゾーンを変更するには、**ウィンドウ > 設定 > IBM MQ Explorer > ファイル転送管理** をクリックし、**タイム・ゾーン:** リストから別のタイム・ゾーンを選択します。「**OK**」をクリックします。
ファイルの転送中に転送がリカバリー状態に入ると、開始された時刻は更新され、ファイル転送が再開された時刻を反映します。

タスクの結果

このタブの情報は定期的に自動的に最新表示されますが、「**現在の転送進行状況**」タブに表示されている内容を強制的に最新表示するには、「コンテンツ」ビューのツールバーにある「**リフレッシュ**」 をクリックします。

「**現在の転送進行状況**」タブからファイル転送を削除するには、「コンテンツ」ビューのツールバーにある「**完了した転送を削除**」 をクリックします。このボタンをクリックしても、ファイル転送の詳細がタブから削除されるだけです。現行のまたはスケジュール済みの転送は停止またはキャンセルされません。

「**現在の転送進行状況**」タブを閉じた後にそこに戻る場合は、「**ウィンドウ**」 > 「**ビューの表示**」 > 「**その他**」 > 「**その他**」 > 「**ファイル転送管理 - 現在の転送進行状況**」をクリックすることで、タブを表示できます。「**OK**」をクリックします。

次のタスク

さらに、カスタム・ファイル転送モニター用のアプリケーションを開発することも可能です。そのためには、対象の Managed File Transfer 管理トピックのサブスクリプションをプログラマチックに作成するか、管理方式で作成します。そうすれば、モニター・アプリケーションでそのトピックの Managed File Transfer ファイル転送アクティビティ・パブリケーションを受け取れるようになります。サブスクリプション・トピックとパブリケーション・メッセージのフォーマットの詳細については、[747 ページの『ファイル転送進行メッセージの例』](#)を参照してください。

関連タスク

[167 ページの『リモート調整キュー・マネージャーをモニターするための IBM MQ Explorer の構成』](#)

リモート・システムで実行中の調整キュー・マネージャーに関連するファイル転送をモニターするには、IBM MQ Explorer を使用します。IBM WebSphere MQ 7.5 以降では、IBM MQ Explorer を実行可能なシステムが必要です。リモート調整キュー・マネージャーに接続できるように IBM MQ Explorer コンポーネントをインストールする必要があります。

[168 ページの『「転送ログ」のファイル転送の状況の表示』](#)

ファイル転送の詳細は、IBM MQ Explorer の「**転送ログ**」を使用して表示できます。対象にできるのは、コマンド行または IBM MQ Explorer のいずれかから開始された転送です。また、「**転送ログ**」に表示される内容をカスタマイズすることもできます。

リモート調整キュー・マネージャーをモニターするための IBM MQ Explorer の構成

リモート・システムで実行中の調整キュー・マネージャーに関連するファイル転送をモニターするには、IBM MQ Explorer を使用します。IBM WebSphere MQ 7.5 以降では、IBM MQ Explorer を実行可能なシステムが必要です。リモート調整キュー・マネージャーに接続できるように IBM MQ Explorer コンポーネントをインストールする必要があります。

このタスクについて

前提: リモート調整キュー・マネージャーに接続する権限があること。それには、リモート接続を許可するようにキュー・マネージャーを構成します。

このように構成する方法について詳しくは、53 ページの『クライアント・モードでチャンネル認証を使用してキュー・マネージャーに接続する操作』および 352 ページの『MFT 固有リソースの権限の管理』を参照してください。

Windows または Linux を実行していないシステム上のエージェント間でキュー・マネージャーとファイル転送をモニターするには、以下のステップを使用して、IBM MQ Explorer をリモート・システムに接続するように構成します。

手順

1. IBM MQ Explorer を開始します。
2. IBM MQ Explorer がロードされたら、「ファイル転送管理」フォルダーを右クリックして「新規構成」を選択します。
3. ウィザードに従って調整およびコマンド・キュー・マネージャーを選択し、次に構成の名前を定義します。
4. 「完了」をクリックして定義を完了します。
5. 定義を完了したら、その定義を右クリックして「接続」を選択します。

タスクの結果

これで、IBM MQ Explorer を開始して、調整キュー・マネージャーに関連する Managed File Transfer ネットワークの転送アクティビティをモニターするために使用できます。

関連タスク

166 ページの『進行中のファイル転送のモニター』

IBM MQ Explorer の「ファイル転送管理 - 現在の転送進行状況」タブを使用して、進行中のファイル転送をモニターできます。このファイル転送は、IBM MQ Explorer またはコマンド行のいずれかから開始できます。このタブには、スケジュール済み転送が開始した時点でのスケジュール済み転送の進行も表示されます。

168 ページの『「転送ログ」のファイル転送の状況の表示』

ファイル転送の詳細は、IBM MQ Explorer の「転送ログ」を使用して表示できます。対象にできるのは、コマンド行または IBM MQ Explorer のいずれかから開始された転送です。また、「転送ログ」に表示される内容をカスタマイズすることもできます。

「転送ログ」のファイル転送の状況の表示

ファイル転送の詳細は、IBM MQ Explorer の「転送ログ」を使用して表示できます。対象にできるのは、コマンド行または IBM MQ Explorer のいずれかから開始された転送です。また、「転送ログ」に表示される内容をカスタマイズすることもできます。

手順

1. 「ナビゲーター」ビューで「ファイル転送管理」を展開して、転送ログを表示する調整キュー・マネージャーの名前を展開します。
2. 「ナビゲーター」ビューで「転送ログ」をクリックします。「転送ログ」が「コンテンツ」ビューに表示されます。
3. 「転送ログ」ウィンドウに、ファイル転送に関する以下の詳細が表示されます。
 - a) 「ソース」。ソース・ファイルが格納されているシステム上のエージェントの名前。
 - b) 「宛先」。ファイルの転送先となるシステム上のエージェントの名前。
 - c) 「完了状態」。ファイル転送の状況。状態は、「開始」、「進行中」、「成功」、「一部成功」、「取り消し済み」、または「失敗」のいずれかの値です。

- d) 「所有者」。転送要求を実行依頼したホストでのユーザー ID。
- e) 「開始(選択したタイム・ゾーン)」。Managed File Transfer エージェントによってファイル転送要求が受け入れられた時刻と日付。管理者が選択したタイム・ゾーンで表示されます。表示されるタイム・ゾーンを変更するには、ウィンドウ>「設定」> **IBM MQ Explorer > Managed File Transfer** をクリックし、「タイム・ゾーン:」リストから別のタイム・ゾーンを選択します。「OK」をクリックします。
- f) 「状態の記録日時(選択したタイム・ゾーン)」(この列はデフォルトでは表示されません。「転送ログの列の構成」 ウィンドウを使用して、この列を表示するように選択できます)。完了状態が記録された、管理者が選択したタイム・ゾーンでの時刻と日付。
- g) 「ジョブ名」。f`teCreateTransfer` の `-jn` パラメーターを使用するか、Ant スクリプト内でユーザーによって指定された ID。
- h) 「転送 ID」。ファイル転送のための固有 ID。
- i) 「Connect: Direct」。「プロセス番号」、「プロセス名」、「1 次ノード」、「2 次ノード」、「ソース・タイプ」、および「宛先タイプ」の詳細がリストされます。

タスクの結果

注: 転送ログの内部形式は、APAR IC99545 対応の IBM MQ 8.0.0 Fix Pack 1 で変更されています。結果として、IBM MQ Explorer が 8.0.0.1 以降にアップグレードされてから 8.0.0.0 に復元された場合、IBM MQ Explorer が 8.0.0.1 であった間に行われた転送の監査 XML は表示されません。これらの転送の「プロパティ」ウィンドウの XML パネルには、空のテキスト・ボックスが表示されます。

完了した転送に関する詳細を表示するには、正符号 (+) をクリックすることにより、関心のある転送を展開します。その後、その転送に含まれているすべてのソース・ファイル名と宛先ファイル名を表示できます。ただし、多数のファイルから成る転送が現在進行中の場合には、これまでに既に転送されたファイルのみを表示できます。

「転送ログ」に表示されている内容を最新表示するには、「コンテンツ」ビューのツールバーにある「リフレッシュ」ボタン  をクリックします。「転送ログ」内のファイル転送情報は、IBM MQ Explorer の停止と再始動を行うまでログの中に残ります。完了したファイル転送をすべてログから削除する場合は、「コンテンツ」ビューのツールバーにある「完了した転送を削除」 をクリックします。

完了した個別のファイル転送をログから削除するには、転送を右クリックし、「削除」をクリックします。転送を削除しても、進行中またはスケジュール済みの転送は停止または取り消されることはありません。保管された履歴データのみが削除されます。

転送の固有 ID をクリップボードにコピーするには、その転送項目を右クリックしてから「ID のコピー」をクリックします。

転送のメタデータおよび完全な監査 XML は、「プロパティ」アクションの下のポップアップ・メニューから入手できます。

関連概念

246 ページの『[ファイル転送のリカバリーのタイムアウト・オプション](#)』

停止したファイル転送のリカバリーをソース・エージェントが試行し続ける時間を秒単位で設定できます。転送が成功しないままエージェントが再試行間隔のタイムアウトに達した場合、その転送は失敗します。

関連タスク

166 ページの『[進行中のファイル転送のモニター](#)』

IBM MQ Explorer の「[ファイル転送管理 - 現在の転送進行状況](#)」タブを使用して、進行中のファイル転送をモニターできます。このファイル転送は、IBM MQ Explorer またはコマンド行のいずれかから開始できます。このタブには、スケジュール済み転送が開始した時点でのスケジュール済み転送の進行も表示されます。

170 ページの『[転送ログの構成](#)』

表示される情報の内容と情報の表示方法を、IBM MQ Explorer の「[転送ログ](#)」で構成できます。

転送ログの構成

表示される情報の内容と情報の表示方法を、IBM MQ Explorer の「転送ログ」で構成できます。

このタスクについて

「転送ログ」の列の順序を再配置するには、移動する列のタイトルをクリックし、その列を新しい位置にドラッグします。列の新しい順序は、次に IBM MQ Explorer を停止して再始動するまでしか保持されません。

「転送ログ」の項目をフィルター操作するには、「表示するログ項目のフィルタリング」フィールドにストリングを入力します。すべての項目をログに復元するには、フィールドに入力したストリングを削除します。このフィールドでは、有効な任意の Java 正規表現を使用できます。詳細内容は [を参照してください](#)。

転送ログに表示される列をカスタマイズするには、「転送ログの列の構成」を使用します。以下のステップを使用して「転送ログの列の構成」ウィンドウを開始して使用します。

手順

1. 「コンテンツ」ビューで「転送ログ」が開いていることを確認します。「コンテンツ」ビュー・ツールバーで「転送ログの列の構成」をクリックします。「転送ログの列の構成」ウィンドウが開きます。
2. 「転送ログ」の表示をカスタマイズするには、表示または非表示にする列の各チェック・ボックスを選択またはクリアします。「すべて選択」をクリックして「OK」をクリックすると、すべてのチェック・ボックスを選択できます。「すべて選択解除」をクリックして「OK」をクリックすると、すべてのチェック・ボックスをクリアできます。

関連タスク

[166 ページの『進行中のファイル転送のモニター』](#)

IBM MQ Explorer の「ファイル転送管理 - 現在の転送進行状況」タブを使用して、進行中のファイル転送をモニターできます。このファイル転送は、IBM MQ Explorer またはコマンド行のいずれかから開始できます。このタブには、スケジュール済み転送が開始した時点でのスケジュール済み転送の進行も表示されます。

[168 ページの『「転送ログ」のファイル転送の状況の表示』](#)

ファイル転送の詳細は、IBM MQ Explorer の「転送ログ」を使用して表示できます。対象にできるのは、コマンド行または IBM MQ Explorer のいずれかから開始された転送です。また、「転送ログ」に表示される内容をカスタマイズすることもできます。

MFT リソースのモニター

キューやディレクトリーなどの Managed File Transfer リソースをモニターできます。そのリソースで条件が満たされると、リソース・モニターがファイル転送などのタスクを開始します。`fteCreateMonitor` コマンドを使用するか、IBM MQ Explorer に対応した Managed File Transfer プラグインの「モニター」ビューを使用して、リソース・モニターを作成できます。

このタスクについて

Managed File Transfer リソース・モニターは、以下の用語を使用します。

monitor

リソース (ディレクトリーまたはキューなど) を事前定義された定期間隔でポーリングし、リソースのコンテンツが変更されたかどうかを確認する処理。変更されている場合、コンテンツはそのモニターの条件セットと比較されます。条件が一致する場合、このモニター用のタスクが開始されます。

resource

モニターがトリガー条件と比較するために、各ポーリング間隔で検査するシステム・リソース。キュー、ディレクトリーあるいはネストされたディレクトリー構造をモニター対象のリソースにすることができます。

condition

評価される式 (通常、モニター対象リソースのコンテンツに照らして評価されます)。式の評価の結果が真であると、その条件はトリガーの全体条件に与えられます。

trigger condition

すべての条件が満たされると、満たされる全体条件。トリガー条件が満たされると、タスクは処理可能になります。

task

トリガー条件あるいは条件のセットが満たされると、開始される操作。サポートされるタスクは、ファイル転送とコマンド呼び出しです。

trigger file

タスク (通常は転送) を開始できることを示す、モニター対象ディレクトリーに置かれるファイル。例えば、処理されるすべてのファイルが、既知の場所に到着し、転送が可能であること、あるいは別の場合には処理が可能であることを示します。トリガー・ファイルの名前に基づいて、変数置換によって転送対象のファイルを指定することも可能です。詳細については [180 ページの『変数置換を使用した MFT リソース・モニター・タスクのカスタマイズ』](#) を参照してください。

トリガー・ファイルは、ready ファイルまたは go ファイルとも呼ばれます。ただし、この資料では、常にトリガー・ファイルという表現を使用しています。

リソース・モニターは、プロトコル・ブリッジ・エージェントまたは Connect:Direct ブリッジ・エージェントではサポートされません。

MFT のリソース・モニターの概念

Managed File Transfer のリソース・モニター機能の主要概念の概要。

モニター

リソース・モニターは、Managed File Transfer エージェントに関連付けられて、エージェントが開始されるときにのみアクティブになります。モニター中のエージェントが停止すると、モニターも停止します。モニターが作成されたときにエージェントが既に開始されている場合、モニターは即時開始されます。モニター・エージェントは、モニターにより開始されるタスクのソース・エージェントでもある必要があります。

モニター名は、そのエージェント内で固有である必要があります。モニター名は、1 文字以上の長さでなければならず、アスタリスク (*)、パーセント (%)、疑問符 (?) の文字は使用できません。指定されたモニター名の大文字小文字は無視されて、大文字に変換されます。既に存在する名前でのモニターを作成しようとすると、その要求は無視されて、モニター・ログのトピックにその試みが記録されます。

エージェントに作成できるモニターの数に制限はなく、すべてのモニターは同じ優先度で実行されます。モニター対象リソースのオーバーラップ、トリガー条件の矛盾、およびリソースをポーリングする頻度の影響を考慮してください。

リソース・モニターのオーバーラップがあると、以下の状態が発生する場合があります。

- ソースとなるロケーション/項目で競合が発生する可能性がある。
- 同じソース項目に対して重複した転送要求が発生する可能性がある。
- ソース項目の競合が原因で、転送において予期しないエラーや障害が発生する。

同じロケーションを複数のモニターがスキャンし、同じ項目を対象としてトリガーする可能性がある場合には、2 つの異なるモニターがその同じ項目に対する管理対象転送要求を実行依頼するという問題が発生する可能性があります。

リソース・モニターは、各ポーリング間隔の時間が過ぎると、リソースのコンテンツを調べます。リソースのコンテンツは、トリガー条件と比較されて、もし条件が満たされるとそのモニターに関連付けられているタスクが呼び出されます。

タスクは、非同期に開始されます。条件の一致があり、タスクが開始された場合、モニターはリソース・コンテンツに対してさらに変更がないかポーリングします。したがって、例えば、reports.go というファイルがモニター対象ディレクトリーに到着したために一致が起こると、タスクは直ちに開始されます。たとえそのファイルがまだ存在していても、次のポーリング間隔でタスクが再度開始されることはありません。しかし、もしファイルが削除されてディレクトリーに再び置かれるか、あるいは、そのファイルが更新される (最終変更日時属性が変更されるなど) と次のトリガー条件の検査により、再びタスクが呼び出されることとなります。

リソース

Managed File Transfer のリソース・モニターは、次の 2 つのタイプのリソースのコンテンツをポーリングできます。

ディレクトリーまたはネストされたディレクトリー構造

ディレクトリーをモニターして、トリガー・ファイルが存在するかどうかを確認する、というのが 1 つの一般的なシナリオです。外部アプリケーションは、複数のファイルを処理して既知のソース・ディレクトリーに配置する場合があります。アプリケーションが処理を完了すると、トリガー・ファイルをモニター対象の場所に配置することによって、ファイルを転送する準備ができていないか、ファイルを転送する準備ができていないことを示します。トリガー・ファイルは、Managed File Transfer リソース・モニターによって検出され、ソース・ディレクトリーから別の Managed File Transfer Agent へのそれらのファイルの転送が開始されます。

デフォルトで、指定されたディレクトリーがモニターされています。サブディレクトリーも検査するには、**fteCreateTransfer** コマンドの再帰レベルを設定します。

ディレクトリーをモニターする 2 つの例を以下に示します。

- トリガー・ファイル (例えば `trigger.file`) をモニターし、ワイルドカード (例えば `*.zip`) を転送する。
- `*.zip` をモニターし、`${FilePath}` (例えば転送を起動したファイル) を転送する。変数置換について詳しくは、[180 ページの『変数置換を使用した MFT リソース・モニター・タスクのカスタマイズ』](#)を参照してください。

注：`*.zip` をモニターし、`*.zip` を転送するモニターを作成しないでください。モニターは、システム上のすべての `.zip` ファイルについて、`*.zip` の転送を開始しようとします。つまり、`*.zip` についてモニターで `*` に相当する数の転送が発生することになります。

ディレクトリーをモニターするためのリソース・モニターを作成する例については、[178 ページの『ディレクトリーのモニターおよび変数置換の使用』](#)を参照してください。

IBM MQ キュー

キューをモニターする例としては、外部アプリケーションがメッセージを生成し、既知のキューにメッセージを同じグループ ID で書き込む場合があります。アプリケーションがキューにメッセージを入れ終わると、そのグループは完了したことが示されます。メッセージの完了グループは Managed File Transfer リソース・モニターによって検出され、ソース・キューからファイルへのメッセージのグループの転送が開始されます。キューをモニターするためのリソース・モニターを作成する例については、[180 ページの『例: MFT リソースの構成』](#)を参照してください。

注：指定できるのは、1 つのキューにつき 1 つのモニターだけです。ある IBM MQ キューをポーリングするために複数のモニターを指定した場合は、予測不能な動作が発生します。

データ・セットのモニターはサポートされていません。

トリガー条件

リソースに他のストリングまたはパターンと一致する値が含まれている場合に条件が満たされます。条件は、以下のいずれでも構いません。

- ファイル名 (パターン) が一致する。
- ファイル名 (パターン) の一致がない。
- ファイル・サイズ
- ポーリングを繰り返してもファイル・サイズが変わらない場合に一致する。

ファイル名の一致は、次のように表すことができます。

- スtring の完全一致
- 簡単なワイルドカード・マッチング ([653 ページの『MFT でのワイルドカード文字の使用』](#)を参照)
- 正規表現の一致

また、ファイル名は、一致することのないファイル名を識別するワイルドカードまたは Java 正規表現を使用して、ファイル名の一致から除外することもできます。

一致したファイルが検出されると、最終変更日時のタイム・スタンプが保存されます。その後のポーリングでファイルが変更されたことが検出されると、トリガー条件が再度満たされてタスクが開始されます。条件でファイルが存在しないときを検出するようになっている場合、モニター対象ディレクトリーにそのファイル名パターンと一致するファイルがないと、タスクが開始されます。次にファイル名パターンと一致するファイルがそのディレクトリーに追加されると、そのファイルが削除された場合にのみタスクが開始されます。

タスク

Managed File Transfer では、リソース・モニターによる以下の 2 タイプのタスク開始の構成がサポートされています。

- ファイル転送
- コマンド

ファイル転送タスクは、他のファイル転送と同じように定義されます。モニターが必要とするタスク XML を生成する便利な方法は、`fteCreateTransfer` コマンドを `-gt` パラメーターを指定して実行することです。このコマンドは、転送仕様を含むタスク定義を XML 文書として生成します。次にこのタスク XML 文書の名前を `fteCreateMonitor` コマンドの `-mt` パラメーターの値として渡します。`fteCreateMonitor` は実行時にタスク XML 文書を読み取ります。`fteCreateMonitor` の実行後にタスク XML ファイルに加えられた変更は、モニターで使用されません。

コマンド・タスクは、Ant スクリプトを実行するか、実行可能プログラムを呼び出すか、または、JCL ジョブを実行することができます。詳細については、[コマンドおよびスクリプトを呼び出すモニター・タスクの構成](#)を参照してください。

ファイル転送タスクを使用している場合、1つのタスクに一括してまとめるトリガー条件の数を選択できます。デフォルトでは、1つのトリガー条件が1つのタスクを開始します。`-bs` オプションを指定して `FteCreateMonitor` コマンドを実行すると、1つのタスクにまとめてバッチ処理されるトリガー条件の数を選択できます。

リソース・モニターのバックアップとリストア

以前に定義したリソース・モニターをバックアップしておけば、後からその定義を再利用することができます。使用できる各種オプションを以下にまとめます。

- `fteCreateMonitor` コマンドで `-ox` パラメーターを使用すれば、1つのリソース・モニター構成を XML ファイルにエクスポートできます。`-ix` パラメーターを使用すれば、XML ファイルからリソース・モニター構成をインポートしてリソース・モニターをリストアできます。
- `-ox` を指定した `fteListMonitors` コマンドを使用して、単一リソース・モニターの定義を XML ファイルにエクスポートします。
- **V 9.0.5** IBM MQ 9.0.5 から、`-od` を指定して `fteListMonitors` コマンドを使用し、複数のリソース・モニター定義を指定のディレクトリーにエクスポートします。各リソース・モニター定義が別々の XML ファイルに保存されます。`-od` オプションを使用して、指定のディレクトリーに1つのリソース・モニター定義をエクスポートすることも可能です。

詳しくは、[193 ページの『MFT リソース・モニターのバックアップとリストア』](#)を参照してください。

リソース・モニターのロギング

V 9.0.3

IBM MQ 9.0.3 以降、Managed File Transfer にはリソース・モニター・ロギングが組み込まれています。詳しくは、[191 ページの『MFT リソース・モニターのロギング』](#)を参照してください。

関連概念

[180 ページの『変数置換を使用した MFT リソース・モニター・タスクのカスタマイズ』](#)

アクティブなリソース・モニターのトリガー条件が満たされると、定義されたタスクが呼び出されます。毎回同じ宛先エージェントまたは同じ宛先ファイル名を使用して転送またはコマンド・タスクを呼び出すことができますが、実行時にタスク定義を変更することもできます。これは、タスク定義 XML に変数名を挿入することで行います。トリガー条件が満たされているとモニターが判断し、タスク定義に変数名が含まれている場合は、変数名を変数値と置換してから、タスクを呼び出します。

関連タスク

[174 ページの『コマンドおよびスクリプトを開始する MFT モニター・タスクの構成』](#)

リソース・モニターの関連タスクは、ファイル転送の実行に限定されません。また、実行可能プログラム、Ant スクリプト、または JCL ジョブなどのモニター・エージェントから他のコマンドを呼び出すようにモニターを構成することもできます。コマンドを呼び出すには、モニター・タスク定義 XML を編集して、引数およびプロパティーなど、対応するコマンド呼び出しパラメーターを指定した 1 つ以上のコマンド・エレメントを含めます。

[180 ページの『例: MFT リソースの構成』](#)

-mq パラメーターを **fteCreateMonitor** コマンドで使用することにより、リソース・モニターによってモニターされるリソースとして、IBM MQ キューを指定できます。

[186 ページの『キューのモニターおよび変数置換の使用』](#)

fteCreateMonitor コマンドを使用して、キューをモニターし、モニターしたキューからファイルにメッセージを転送できます。モニターされるキューから読み取られる最初のメッセージにある任意の IBM MQ メッセージ・プロパティーの値をタスク XML 定義に置換して、転送動作の定義に使用できます。

関連資料

[413 ページの『fteCreateMonitor: MFT リソース・モニターの作成』](#)

fteCreateMonitor コマンドは、コマンド行から新規リソース・モニターを作成して開始します。

Managed File Transfer を使用してリソース (例えば、ディレクトリーの内容) をモニターして、トリガー条件が満たされたときに、指定したタスク (ファイル転送など) を開始することができます。

[476 ページの『fteListMonitors: MFT リソース・モニターのリスト』](#)

コマンド行から **fteListMonitors** コマンドを使用すると、Managed File Transfer ネットワークの既存のリソース・モニターをすべてリストできます。

[466 ページの『fteDeleteMonitor: MFT リソース・モニターの削除』](#)

コマンド行で Managed File Transfer の既存のリソース・モニターを停止して削除する場合には、

fteDeleteMonitor コマンドを使用します。リソース・モニター・エージェントに対してこのコマンドを発行します。

コマンドおよびスクリプトを開始する MFT モニター・タスクの構成

リソース・モニターの関連タスクは、ファイル転送の実行に限定されません。また、実行可能プログラム、Ant スクリプト、または JCL ジョブなどのモニター・エージェントから他のコマンドを呼び出すようにモニターを構成することもできます。コマンドを呼び出すには、モニター・タスク定義 XML を編集して、引数およびプロパティーなど、対応するコマンド呼び出しパラメーターを指定した 1 つ以上のコマンド・エレメントを含めます。

このタスクについて

モニター・エージェントから呼び出せるようにする実行可能プログラム、Ant スクリプト、または JCL ジョブへのファイル・パスを、モニター・エージェントの `commandPath` に含める必要があります。コマンド・パスのプロパティーについては、[365 ページの『commandPath MFT プロパティー』](#)を参照してください。

以下のいずれかの方法で、タスク定義 XML 文書を作成できます。

- `FileTransfer.xsd` スキーマに従って手動でタスク定義 XML 文書を作成する。詳しくは、[175 ページの『スキーマに従って手動でタスク定義 XML を作成する』](#)を参照してください。
- **fteCreateTransfer -gt** パラメーターで生成された XML 文書を独自のタスク定義の基本として使用して、編集する。詳しくは、[177 ページの『生成済み文書の変更によるタスク定義文書の作成』](#)を参照してください。

転送タスクまたはコマンド・タスクのどちらを使用するかにかかわらず、タスク定義は <request> ルート・エレメントで開始する必要があります。<request> の子エレメントは、<managedTransfer> または <managedCall> のいずれかでなければなりません。実行するコマンドまたはスクリプトが1つの場合は、通常 <managedCall> を選択し、ファイル転送とオプションで最大4つのコマンド呼び出しをタスクに含める場合は <managedTransfer> を選択します。

スキーマに従って手動でタスク定義 XML を作成する

このタスクについて

スキーマ FileTransfer.xsd に従って、手動でタスク定義 XML ファイルを作成できます。このスキーマは、MQ_INSTALLATION_PATH/mqft/samples/schema にあります。このスキーマについて詳しくは、780 ページの『ファイル転送要求メッセージ・フォーマット』を参照してください。

例

以下の例は、<managedCall> エレメントを使用して RunCleanup.xml という Ant スクリプトを呼び出す、cleanuptask.xml、として保存されたタスク定義 XML 文書の例を示しています。

RunCleanup.xml Ant スクリプトは、モニター・エージェントの commandPath になければなりません。

```
<?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>
      <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>
  </managedCall>
</request>
```

<agent> エレメントは、commandPath で指定された Ant スクリプトを使用して構成される Managed File Transfer Agent を指定します。

<call><command>... 構造は、実行する実行可能ファイルまたはスクリプトを定義します。このコマンドは、以下の値のうち1つを指定できる type 属性をオプションで持ちます。

antscript

Ant スクリプトを別個の JVM で実行します。

executable

実行可能プログラムを呼び出します。

jcl

JCL ジョブを呼び出します。

type 属性を省略する場合、デフォルト値 executable が使用されます。

`name` 属性は、パス情報なしで、実行する Ant スクリプト、実行可能プログラム、または JCL ジョブの名前を指定します。エージェントは、そのエージェントの `agent.properties` ファイル内の `commandPath` プロパティで指定された場所にあるスクリプトまたはプログラムを検索します。

`retrycount` 属性は、プログラムが成功を示す戻りコードを戻さなかったときに、再度プログラムの呼び出しを試行する回数を指定します。この属性に指定する値は、負の値を指定することはできません。`retrycount` 属性を指定しない場合、デフォルト値のゼロが使用されます。

`retrywait` 属性は、プログラム呼び出しを再試行するまでの待機時間(秒数)を指定します。この属性に指定する値は、負の値を指定することはできません。`retrywait` 属性を指定しない場合、デフォルト値のゼロが使用されます。

`successsrc` 属性は、プログラム呼び出しが正常に実行される条件を決定するために使用される式です。コマンドの処理戻りコードは、この式を使用して評価されます。値は、ブール値の OR を表す垂直バー文字 (!)、またはブール値の AND を表すアンパーサンド (&) 文字で結合された 1 つ以上の式で構成することができます。各式は、以下のいずれかのタイプの式とすることができます。

- 処理戻りコードとの等価テストを示す数値。
- 処理戻りコードとの大なりテストを示す、接頭部に「大なり」文字 (>) が付いた数値。
- 処理戻りコードとの小なりテストを示す、接頭部に「小なり」文字 (<) が付いた数値。
- 処理戻りコードとの不等テストを示す、接頭部に感嘆符文字 (!) が付いた数値。例えば、>2&<7&!5|0|14 は、戻りコード 0、3、4、6、14 を正常と解釈します。これ以外の戻りコードは、すべて失敗と解釈されます。

`successsrc` 属性を指定しない場合、デフォルト値のゼロが使用されます。これは、ゼロの戻りコードを戻した場合にのみ、コマンドは正常に実行されたと判断されるという意味です。

Ant スクリプトの場合、通常、<target> および <property> エlement を指定します。<target> Element の値は、Ant スクリプト内のターゲット名と一致している必要があります。

実行可能プログラムの場合、<argument> Element を指定できます。ネストされた argument Element を使用すると、プログラム呼び出しの一部として呼び出されるプログラムに渡される引数が指定されます。このプログラム実行引数は、argument Element の出現する順序で argument Element により指定された値から構成されます。ゼロ個以上の argument Element をプログラム呼び出しのネストされた Element として指定できます。

管理者は、<managedCall> Element を含むタスク定義 XML 文書を使用して、モニターを通常どおりに定義し、開始します。以下に例を示します。

```
fteCreateMonitor -ma AGENT1 -mm QM1 -md /monitored -mn MONITOR01 -mt
/tasks/cleanuptask.xml -pi 30 -pu seconds -tr match,*go
```

転送定義 XML 文書へのパスは、**fteCreateMonitor** コマンドを実行するローカル・ファイル・システムにある必要があります(この例では、/tasks/cleanuptask.xml)。cleanuptask.xml 文書は、リソース・モニターの作成にのみ使用されます。cleanuptask.xml 文書が参照するタスク (Ant スクリプトまたは JCL ジョブ) も、モニター・エージェントのコマンド・パスにある必要があります。モニター・トリガー条件が満たされると、タスク定義 XML 内のすべての変数はモニターからの実際の値で置換されます。したがって、例えば、\${FilePath} は、エージェントに送信される要求メッセージで /monitored/cleanup.go に置換されます。要求メッセージは、エージェントのコマンド・キューに置かれます。コマンド・プロセッサは、要求がプログラム呼び出し用であることを検出し、指定されたプログラムを開始します。antscript タイプのコマンドが呼び出された場合は、新規 JVM が開始され、Ant タスクはその下で実行されます。変数置換の用法について詳しくは、『[変数置換を使用したタスクのカスタマイズ](#)』を参照してください。

関連概念

[180 ページの『変数置換を使用した MFT リソース・モニター・タスクのカスタマイズ』](#)

アクティブなリソース・モニターのトリガー条件が満たされると、定義されたタスクが呼び出されます。毎回同じ宛先エージェントまたは同じ宛先ファイル名を使用して転送またはコマンド・タスクを呼び出すことができますが、実行時にタスク定義を変更することもできます。これは、タスク定義 XML に変数名を

挿入することで行います。トリガー条件が満たされているとモニターが判断し、タスク定義に変数名が含まれている場合は、変数名を変数値と置換してから、タスクを呼び出します。

関連資料

780 ページの『ファイル転送要求メッセージ・フォーマット』

ファイル転送は、エージェントのコマンド・キューに到着した XML メッセージによって開始されます。通常、ユーザーがファイル転送コマンドを発行したか、IBM MQ Explorer を使用した結果、XML メッセージが到着します。転送要求 XML は、FileTransfer.xsd スキーマに準拠していて、<request> エlement をルート・Element として持っている必要があります。FileTransfer.xsd スキーマ文書は MQ_INSTALLATION_PATH/mqft/samples/schema ディレクトリーにあります。FileTransfer.xsd スキーマは、同じディレクトリーにある fteutils.xsd をインポートします。

365 ページの『commandPath MFT プロパティー』

commandPath プロパティーを使用して、Managed File Transfer がコマンドを実行できる場所を指定します。指定された commandPath にある任意のコマンドは、エージェントにコマンドを送信できるリモート・クライアント・システムから事実上呼び出すことができるため、このプロパティーを設定するときは十分な注意を払ってください。

生成済み文書の変更によるタスク定義文書の作成

このタスクについて

ftCreateTransfer の **-gt** オプションによって生成された XML 文書を変更することによって、モニター・タスク定義文書を作成できます。生成された文書の<request> の後に<managedTransfer> Element が続きます。このタスク定義を有効な<managedCall> 構造体に変換するには、以下のステップに従ってください。

手順

1. <managedTransfer> の開始タグと終了タグを <managedCall> タグで置き換えます。
2. <schedule> Element および子ノードをすべて削除します。
3. <sourceAgent> の開始タグと終了タグを <agent> と置き換えて、モニター・エージェント構成の詳細を一致させます。
4. <destinationAgent> および <trigger> Element を除去します。
5. <item> Element を削除する。
6. <transferSet> Element に、新規 <call>...</call> 構造を挿入します。この構造には、以下の例に示すコマンド定義が含まれます。

```
<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>
```

例

また、すべてのファイル転送の詳細を含む<managedTransfer> Element を保持し、4 つまでのコマンド呼び出しを挿入することもできます。この場合は、<metaDataSet> Element と<item> Element の間に、以下の呼び出し Element の選択を挿入します。

preSourceCall

ソース・エージェント上のプログラムを呼び出してから転送を開始します。

postSourceCall

転送を完了した後にソース・エージェント上のプログラムを呼び出します。

preDestinationCall

宛先エージェント上のプログラムを呼び出してから転送を開始します。

postDestinationCall

転送を完了した後に宛先エージェント上のプログラムを呼び出します。

これらの各要素は、前の例で説明されているように、<command>要素構造を取ります。FileTransfer.xsd スキーマは、さまざまな呼び出し要素で使用されるタイプを定義します。

次の例では、タスク定義文書内の preSourceCall、postSourceCall、preDestinationCall、および postDestinationCall を示します。

```
⋮
<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">
⋮
```

異なるタイプのコマンドを転送に混入できます。引数、ターゲット、およびプロパティ・要素はオプションです。

ディレクトリーのモニターおよび変数置換の使用

ftCreateMonitor コマンドを使用して、ディレクトリーをモニターすることができます。置換変数の値をタスク XML 定義に置換して、転送動作の定義に使用できます。

このタスクについて

この例では、ソース・エージェントの名前は AGENT_HOP です。AGENT_HOP がモニターするディレクトリーは、/test/monitored です。エージェントは、ディレクトリーを 5 分おきにポーリングします。

.zip ファイルがディレクトリーに書き込まれた後、そのファイルをディレクトリーに書き込んだアプリケーションは、トリガー・ファイルと同じディレクトリーに書き込みます。トリガー・ファイルの名前は、.zip ファイルの名前と同じですが、ファイル拡張子が異なります。例えば、ファイル file1.zip がディレクトリーに書き込まれた後、ファイル file1.go がディレクトリーに書き込まれます。リソース・モニターは、そのディレクトリー内でパターン *.go と一致するファイルをモニターし、変数置換を使用して、関連する .zip ファイルの転送を要求します。

手順

1. モニター起動時にモニターが実行するタスクを定義するタスク XML を作成します。

```
<?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>
```

トリガー・ファイルに関連する値に置き換えられる変数を、**太字**で強調表示しています。このタスク XML は、ファイル /home/USER1/task.xml に保存されます。

- ディレクトリー /test/monitored をモニターするリソース・モニターを作成します。以下のコマンドを実行依頼します。

```
fteCreateMonitor -ma AGENT_HOP -mm QM_HOP -md /test/monitored
                 -mn myMonitor -mt /home/USER1/task.xml
                 -tr match,*.*go -pi 5 -pu minutes
```

- ユーザーまたはプログラムが、ファイル jump.zip をディレクトリー /test/monitored に書き込んだ後、ファイル jump.go をそのディレクトリーに書き込みます。
- ファイル jump.go の存在によって、モニターがトリガーされます。エージェントは、トリガー・ファイルに関する情報を、タスク XML に置換します。この結果、タスク XML は以下のように変換されます。

```
<?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>
```

タスクの結果

タスク XML によって定義された転送が実行されます。jump.zip ファイルが AGENT_HOP によって /test/monitored ディレクトリーから読み取られ、AGENT_SKIP が実行されているシステム上の /out/jump.zip というファイルに転送されます。

関連概念

180 ページの『変数置換を使用した MFT リソース・モニター・タスクのカスタマイズ』

アクティブなリソース・モニターのトリガー条件が満たされると、定義されたタスクが呼び出されます。毎回同じ宛先エージェントまたは同じ宛先ファイル名を使用して転送またはコマンド・タスクを呼び出すことができますが、実行時にタスク定義を変更することもできます。これは、タスク定義 XML に変数名を挿入することで行います。トリガー条件が満たされているとモニターが判断し、タスク定義に変数名が含まれている場合は、変数名を変数値と置換してから、タスクを呼び出します。

関連タスク

174 ページの『[コマンドおよびスクリプトを開始する MFT モニター・タスクの構成](#)』

リソース・モニターの関連タスクは、ファイル転送の実行に限定されません。また、実行可能プログラム、Ant スクリプト、または JCL ジョブなどのモニター・エージェントから他のコマンドを呼び出すようにモニターを構成することもできます。コマンドを呼び出すには、モニター・タスク定義 XML を編集して、引数およびプロパティーなど、対応するコマンド呼び出しパラメーターを指定した 1 つ以上のコマンド・エレメントを含めます。

関連資料

413 ページの『[fteCreateMonitor: MFT リソース・モニターの作成](#)』

fteCreateMonitor コマンドは、コマンド行から新規リソース・モニターを作成して開始します。

Managed File Transfer を使用してリソース (例えば、ディレクトリーの内容) をモニターして、トリガー条件が満たされたときに、指定したタスク (ファイル転送など) を開始することができます。

例: MFT リソースの構成

-mq パラメーターを **fteCreateMonitor** コマンドで使用するにより、リソース・モニターによってモニターされるリソースとして、IBM MQ キューを指定できます。

このタスクについて

この例では、モニターされるリソースは *MONITORED_QUEUE* というキューです。このキューは、モニター・エージェントのキュー・マネージャー *QM_NEPTUNE* に存在していなければなりません。キューがモニター対象になる条件は、メッセージの完全グループが存在することです。条件が満たされる場合に実行されるタスクは、ファイル *task.xml* 内に定義されます。

注: 個々のキューをモニターするために、複数のリソース・モニターを作成しないでください。作成した場合、予測不能な動作が発生します。

手順

次のコマンドを入力します。

```
fteCreateMonitor -ma AGENT_NEPTUNE -mn myMonitor -mm QM_NEPTUNE -mq MONITORED_QUEUE  
-mt task.xml -tr completeGroups -pi 5 -pu minutes
```

条件 *completeGroups* が真であるかどうかを調べるため、モニターは 5 分ごとにキューをチェックします。1 つ以上の完全グループがキューにある場合には、完全グループごとに一度ずつ、*task.xml* ファイルに定義されているタスクを、モニターが実行します。

変数置換を使用した MFT リソース・モニター・タスクのカスタマイズ

アクティブなリソース・モニターのトリガー条件が満たされると、定義されたタスクが呼び出されます。毎回同じ宛先エージェントまたは同じ宛先ファイル名を使用して転送またはコマンド・タスクを呼び出すことができますが、実行時にタスク定義を変更することもできます。これは、タスク定義 XML に変数名を挿入することで行います。トリガー条件が満たされているとモニターが判断し、タスク定義に変数名が含まれている場合は、変数名を変数値と置換してから、タスクを呼び出します。



重要: 変数名は大/小文字を区別しません。

置換に使用される変数は、正のトリガー条件でのみ使用可能です。match および *fileSize* トリガー条件によってのみ、変数は置換されます。不一致条件が使用され、タスク定義に置換変数名がある場合、タスクは呼び出されず、モニターは 110 とエラー・メッセージ *BFGDM0060E* の戻りコードを出します。

モニターされるリソースがキューの場合

モニターされるキューから読み取られる最初のメッセージにある任意の IBM MQ メッセージ・プロパティの値を、タスク XML 定義に置換できます。

ユーザー定義メッセージ・プロパティには、接頭部 `usr.` が付きますが、変数名にはこの接頭部を含めません。変数名は、中括弧 `{}` で囲んで、その前にドル記号 (`$`) 文字を付加する必要があります。

例えば、`${destFileName}` は、ソース・キューから読み取られる最初のメッセージの `usr.destFileName` メッセージ・プロパティの値に置き換えられます。詳しくは、679 ページの『MFT がソース・キューのメッセージから読み取る IBM MQ メッセージ・プロパティ』および 186 ページの『キューのモニターおよび変数置換の使用』を参照してください。

変数がメッセージ・プロパティとして定義されていない場合、モニターは BFGDM0060E エラーを報告し、戻りコード 110 を戻します (モニター・タスク変数置換が失敗しました)。これに加えて、エージェントは以下のエラー・メッセージをイベント・ログ (`outputN.log`) に書き込みます。

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

モニターに対して中程度のリソース・モニター・ロギングまたは詳細リソース・モニター・ロギングが使用可能になっている場合、モニターは以下のメッセージをエージェントのリソース・モニター・イベント・ログ (`resmoneventN.log`) に書き込みます。

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

リソース・モニターのロギングについて詳しくは、191 ページの『MFT リソース・モニターのロギング』を参照してください

デフォルトで用意されている置換変数を以下の表にまとめます。例えば、`${AGENTNAME}` は、リソース・モニター・エージェントの名前に置換されます。

変数	説明
AGENTNAME	リソース・モニター・エージェントの名前。
QUEUENAME	モニターされるキューの名前。
ENCODING	キューにある最初のメッセージ、またはグループにある最初のメッセージの文字エンコード。
MESSAGEID	キューにある最初のメッセージ、またはグループにある最初のメッセージの IBM MQ メッセージ ID。
GROUPID	グループの IBM MQ グループ ID、または単一のメッセージのみ検出された場合はメッセージ ID。この変数は完全なグループをモニターしている場合のみ、設定されます。
CurrentTimeStamp	モニターがトリガーした時の現地時間に基づいたタイム・スタンプ。タイム・スタンプ値はエージェントに固有です。
CurrentTimeStamp UTC	モニターがトリガーした時の UTC タイム・ゾーンでのタイム・スタンプ。タイム・スタンプ値はエージェントに固有です。

モニターされるリソースがディレクトリーの場合

次の表に、タスク XML 定義で置換できる一連の変数名を示します。

表 19. 置換可能な変数	
変数	説明
FilePath	トリガー・ファイルの完全なパス名。
FileName	トリガーのファイル名の部分。
LastModifiedTime	トリガー・ファイルの最終変更時刻。エージェントを実行しているタイム・ゾーンの現地時間が ISO 8601 の時間形式で表示されます。
LastModifiedDate	トリガー・ファイルの最終変更日。エージェントを実行しているタイム・ゾーンの現地日付が ISO 8601 の日付形式で表示されます。
LastModifiedTimeUTC	トリガー・ファイルの最終変更時刻。この時刻は、UTC タイム・ゾーンに変換された現地時間として表され、ISO 8601 時間として書式設定されます。
LastModifiedDateUTC	トリガー・ファイルの最終変更日。この日付は、UTC タイム・ゾーンに変換された現地日付として表され、ISO 8601 日付として書式設定されます。
AgentName	リソース・モニター・エージェントの名前。
CurrentTimeStamp	モニターがトリガーした時の現地時間に基づいたタイム・スタンプ。タイム・スタンプ値はエージェントに固有です。
CurrentTimeStampUTC	モニターがトリガーした時の UTC タイム・ゾーンでの時刻に基づいたタイム・スタンプ。タイム・スタンプ値はエージェントに固有です。

モニター対象リソースがトリガー・ファイルの場合

次の表に、リソース・モニターがトリガー・ファイルの内容を使用して転送する必要のあるファイルを判別するときに置き換えることができる変数名のセットを示します。

表 20. トリガー・ファイルを使用する際に置換できる変数	
変数	説明
contentSource	ソース・ファイルの完全パス名。
contentDestination	宛先ファイルの完全パス名。

変数名の前には、ドル記号 (\$) 文字が必要です。また、中括弧で囲まれていなければなりません。例えば、`${FilePath}` は、一致するトリガー・ファイルの完全修飾ファイル・パスに置き換えられます。

変数名に適用してさらに細分化できる特殊キーワードが 2 つあります。次のとおりです。

トークン

置換するトークン索引 (左側は 1 から始まり、右側は -1 から始まる)

分離文字

変数値をトークン化する単一文字。デフォルトは、UNIX プラットフォームではスラッシュ文字 (/)、Windows プラットフォームでは円記号文字 (¥) ですが、区切り文字には、変数値に使用できる任意の有効な文字を使用できます。

変数名で separator キーワードが指定された場合には、変数値は、その分離文字でトークンに分割されます。

token キーワードに割り当てた値は、変数名を置き換えるために使用するトークンを選択する索引として使用されます。トークン索引は、変数内の最初の文字に対して相対的なもので、1 から始まります。token キーワードが指定されていない場合は、変数全体が挿入されます。

メッセージ XML のエージェント名に置換される値はすべて、大/小文字を区別せずに扱われます。Managed File Transfer Agent 名はすべて大文字です。Paris という値がメッセージ XML のエージェント属性に置換された場合には、この値はエージェント PARIS への参照として解釈されます。

関連概念

183 ページの『例: リソース・モニター定義の変数置換』

XML と IBM MQ Explorer を使用したリソース・モニター定義の変数置換の例。

関連資料

330 ページの『変数置換によって複数のファイルが1つのファイル名に送られる場合の対応策』

Managed File Transfer で、ディレクトリーをモニターして複数のファイルをソースから宛先の場所に転送する時に、`${FileName}` の変数置換を使用する場合は、変数置換の結果をテストする必要があります。

変数置換をテストしなければならないのは、変数置換の結果として、予期しない組み合わせのファイル転送コマンドが呼び出されることがあるからです。

例: リソース・モニター定義の変数置換

XML と IBM MQ Explorer を使用したリソース・モニター定義の変数置換の例。

変数置換の仕組みを示す例

一致するトリガー・ファイルへのファイル・パスが、Windows プラットフォームでは `c:\MONITOR\REPORTS\Paris\Report2009.doc`、UNIX プラットフォームでは `/MONITOR/REPORTS/Paris/Report2009.doc` であると仮定すると、変数は以下の表に示すように置換されます。

変数の指定	変数置換後
<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 : 資料 UNIX : 資料
<code>\${FilePath{token=3}}</code>	Windows : レポート UNIX : パリ

負のトークン索引を指定して、変数の最後の文字を基準にした相対指定でトークンを選択することもできます (以下の表を参照)。表の例では、同じ変数値 `c:\MONITOR\REPORTS\Paris\Report2009.doc` (Windows の場合) と `/MONITOR/REPORTS/Paris/Report2009.doc` (UNIX プラットフォームの場合) を使用しています。

変数の指定	変数置換後
<code>\${FilePath}</code>	Windows : <code>c:\MONITOR\REPORTS\Paris\Report2009.doc</code> UNIX : <code>/MONITOR/REPORTS/Paris/Report2009.doc</code>

表 22. 負のトークン索引の使用例 (続き)

変数の指定	変数置換後
<code>\${FilePath{token=-2}{separator=.}}</code>	Windows : c:\MONITOR\REPORTS\Paris\Report2009 UNIX : /MONITOR/REPORTS/Paris/Report2009
<code>\${FilePath{token=-2}{separator=\\}}</code>	Windows : パリ UNIX : パリ
<code>\${FilePath{token=-4}}</code>	Windows : モニター UNIX : モニター

置換に使用される変数は、以下の肯定的なトリガー条件および `noSizeChange` オプションに対してのみ使用可能です。これは、肯定トリガー条件規則の例外です。

- 一致
- `fileSize`
- `noSizeChange`

不一致条件が使用され、タスク定義に置換変数名がある場合、タスクは呼び出されず、モニターは 110 とエラー・メッセージ `BFGDM0060E` の戻りコードを出します。

XML の使用例

以下のタスク定義 XML の例では、転送のソース・エージェントとしてモニター・エージェント名 (Paris) を使用し、転送の宛先エージェント名としてファイル・パスの `penultimate` ディレクトリー名 (Report2009) を使用し、転送後のファイル名を、トリガー・ファイルのルートに拡張子 `.rpt` を付けた名前に変更しています。

```
<?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>
```

この結果、タスク XML は以下のように変換されます。

```
<?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"/>
  </managedTransfer>
</request>
```

```

<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> エレメントの agent 属性にある変数 `#{FilePath{token=-2}}` は、値「Paris」で置き換えられます。この値は大/小文字を区別せずに扱われ、エージェント PARIS への参照として解釈されます。

IBM MQ Explorer の使用例

IBM MQ Explorer でリソース・モニターを作成する時に、モニター・プロパティとトリガー条件を指定すると、転送項目をモニターに追加するためのオプションが表示されます。次の例は、「転送項目の追加パネル」で `#{FilePath}` と `#{FileName}` 変数を使用して、リソースモニタの一致から生じる転送をカスタマイズする方法を示しています。

例 1

トリガー条件に適合した時にソース・ファイルを別の場所に転送するために、`#{FilePath}` 変数を使用します。

- ソース・ファイル名を `#{FilePath}` に設定します。
- 宛先の「タイプ」のドロップダウン・メニューから、「ディレクトリー」を選択します。
- 宛先のファイル名を、ソース・ファイルの転送先のロケーションに設定します。例えば、`C:\MFT\out\`にすることができます。

例 2

ソース・ファイルを別の場所に転送し、ファイル拡張子を変更するために、`#{FilePath}` 変数と一緒に `#{FileName}` 変数を使用します。

以下の例では、ソース・ファイルのファイル・パスが `C:\MONITOR\REPORTS\Paris\Report2009.doc` に等しいと想定されています。

- ソース・ファイル名を `#{FilePath}` に設定します。
- 宛先ファイル名をソース・ファイルの転送先の場所に設定し、その後に `#{FileName{token=1}{separator=.}}`、さらにその後にファイルの新しい拡張子を指定します。例えば、ソースファイル名 `C:\MFT\out\Report2009.rpt` と等しい `C:\MFT\out\#{FileName{token=1}{separator=.}}.rpt`、とすることができます。

例 3

ソース・ファイルのファイル・パスの一部を使用して転送の宛先を指定するために、トークンや区切り文字と一緒に `#{FilePath}` 変数を使用します。

以下の例では、ソース・ファイルのファイル・パスが `C:\MONITOR\REPORTS\Paris\Report2009.doc` に等しいことを前提としています。

ソース・ファイル・パスの一部を使用してファイルの宛先を指定できます。

`C:\MONITOR\REPORTS\Paris\Report2009.doc` のファイル・パスの例を使用して、ファイルがソース・ファイルの場所（この例では Paris）に応じてフォルダーに転送される場合は、以下のことを行うことができます。

- ソース・ファイル名を `#{FilePath}` に設定します。

- 宛先ファイル名をそれぞれの場所に対応するフォルダーが置かれる宛先に設定し、ファイル・パスの宛先の部分とファイル名を追加します。例えば、ソースファイル名
C:\MFT\out\Paris\Report2009.doc と等しい C:\MFT\out\\${FilePath}{token=-2}{separator=\\}\\${FileName}, とすることができます。

関連概念

180 ページの『変数置換を使用した MFT リソース・モニター・タスクのカスタマイズ』

アクティブなリソース・モニターのトリガー条件が満たされると、定義されたタスクが呼び出されます。毎回同じ宛先エージェントまたは同じ宛先ファイル名を使用して転送またはコマンド・タスクを呼び出すことができますが、実行時にタスク定義を変更することもできます。これは、タスク定義 XML に変数名を挿入することで行います。トリガー条件が満たされているとモニターが判断し、タスク定義に変数名が含まれている場合は、変数名を変数値と置換してから、タスクを呼び出します。

関連資料

330 ページの『変数置換によって複数のファイルが 1 つのファイル名に送られる場合の対応策』

Managed File Transfer で、ディレクトリーをモニターして複数のファイルをソースから宛先の場所に転送する時に、\${FileName} の変数置換を使用する場合は、変数置換の結果をテストする必要があります。変数置換をテストしなければならないのは、変数置換の結果として、予期しない組み合わせのファイル転送コマンドが呼び出されることがあるからです。

キューのモニターおよび変数置換の使用

fteCreateMonitor コマンドを使用して、キューをモニターし、モニターしたキューからファイルにメッセージを転送できます。モニターされるキューから読み取られる最初のメッセージにある任意の IBM MQ メッセージ・プロパティの値をタスク XML 定義に置換して、転送動作の定義に使用できます。

このタスクについて

この例では、ソース・エージェントは AGENT_VENUS という名前であり、QM_VENUS に接続します。AGENT_VENUS がモニターするキューは START_QUEUE という名前であり、QM_VENUS にあります。エージェントは、キューを 30 分おきにポーリングします。

メッセージの完全に揃ったグループがキューに書き込まれると、モニター・タスクは、いくつかの宛先エージェントの 1 つのファイルにメッセージのグループを送信します。この宛先エージェントは、すべてキュー・マネージャー QM_MARS に接続しています。メッセージのグループが転送されるファイルの名前は、グループの最初のメッセージの IBM MQ メッセージ・プロパティ `usr.fileName` で定義します。メッセージのグループが送信されるエージェントの名前は、グループの最初のメッセージの IBM MQ メッセージ・プロパティ `usr.toAgent` で定義します。`usr.toAgent` ヘッダーが未設定の場合は、宛先エージェント用に使用されるデフォルト値は、AGENT_MAGENTA です。

`useGroups="true"` を指定する場合、`groupId="${GROUPID}"` を指定しないと、転送ではキュー内の最初のメッセージのみが取得されます。そのため、例えば変数置換を使用して `fileName` を生成した場合、`a.txt` の内容が正しくなくなる可能性があります。これは、`fileName` はモニターによって生成されますが、転送では、実際には `fileName` というファイルを生成するメッセージではなく、別のメッセージを取得するためです。

手順

1. モニター起動時にモニターが実行するタスクを定義するタスク XML を作成します。

```
<?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>

```

IBM MQ メッセージ・ヘッダーの値で置き換えられる変数は、**太字**で強調表示しています。このタスク XML は、ファイル /home/USER1/task.xml に保存されます。

- キュー START_QUEUE をモニターするリソース・モニターを作成します。
以下のコマンドを実行依頼します。

```

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

```

- ユーザーまたはプログラムは、メッセージのグループをキュー START_QUEUE に書き込みます。
このグループの最初のメッセージは、次の IBM MQ メッセージ・プロパティを設定しています。

```

usr.fileName=larmer
usr.toAgent=AGENT_VIOLET

```

- 完全に揃ったグループが書き込まれると、モニターが起動されます。エージェントは、IBM MQ メッセージ・プロパティをタスク XML に置換します。
この結果、タスク XML は以下のように変換されます。

```

<?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>

```

タスクの結果

タスク XML によって定義された転送が実行されます。AGENT_VENUS によって START_QUEUE から読み取られた、完全に揃ったメッセージ・グループが、AGENT_VIOLET を実行中のシステムの /reports/larmer.rpt というファイルに書き込まれます。

次のタスク

各メッセージの別個のファイルへの転送

キューをモニターして、すべてのメッセージが別個のファイルに転送されるようにする場合には、このトピックで前述した方法と同様の方法を使用することができます。

1. **fteCreateMonitor** コマンドに **-tr completeGroups** パラメーターを指定して、前述のようにモニターを作成します。
2. タスク XML で、次のように指定します。

```
<queue useGroups="true" groupId="${GROUPID}">START_QUEUE</queue>
```

ただし、メッセージをソース・キューに入れる場合は、それらのメッセージを IBM MQ グループには入れないでください。IBM MQ メッセージ・プロパティを各メッセージに追加します。例えば、メッセージごとに固有のファイル名の値を持つ `usr.filename` プロパティを指定します。こうすることで効果的に、Managed File Transfer Agent がソース・キュー内の各メッセージを異なるグループとして扱います。

メッセージからファイルへの転送の再試行動作をモニターする

リソース・モニターにより起動されたメッセージからファイルへの転送が失敗し、モニターを起動したメッセージ・グループがキューに残っている場合、その転送は後続のポーリング間隔で再発信されます。転送が再発信される回数は、モニター・エージェントの **monitorGroupRetryLimit** プロパティにより制限されます。

メッセージからファイルへの転送が起動された回数は、グループの最初のメッセージの MQMD バックアウト・カウントで判別されます。

メッセージからファイルへの転送が新たに起動されるたびに、転送タスクに対して新しい転送 ID が生成されます。

エージェントを再起動した場合、モニターは、転送が起動された回数が **monitorGroupRetryLimit** の値を超えていた場合でも、再度転送を起動します。この転送の試行により、起動された転送回数が **monitorGroupRetryLimit** の値を超えてしまった場合、エージェントはイベント・ログにエラーを書き込みます。

1 つのメッセージはあたかも 1 つのグループであったように処理され、メッセージがキューに残っており、転送が起動された回数が **monitorGroupRetryLimit** の値未満である間は、ポーリング間隔ごとに転送が起動されます。

monitorGroupRetryLimit プロパティの設定

monitorGroupRetryLimit プロパティの値は、メッセージ・グループがまだキューに存在している場合、モニターがメッセージからファイルへの転送を再度起動する最大回数です。このプロパティのデフォルト値は 10 です。このプロパティの値は、任意の正整数値または -1 に設定できます。このプロパティに値 -1 が指定された場合、モニターは、起動条件が満たされなくなるまで何度でも転送を再度起動します。

モニター・エージェントで **monitorGroupRetryLimit** プロパティを設定するには、次の手順を実行します。

1. **fteStopAgent** コマンドを使用してモニター・エージェントを停止します。
2. モニター・エージェント `agent.properties` ファイルを編集して、
`monitorGroupRetryLimit=number_of_retries` という行を組み込みます。
`agent.properties` ファイルは、`MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/monitoring_agent_name` ディレクトリーにあります。
3. **fteStartAgent** コマンドを使用してモニター・エージェントを開始します。

関連タスク

[180 ページの『例: MFT リソースの構成』](#)

-mq パラメーターを **fteCreateMonitor** コマンドで使用することにより、リソース・モニターによってモニターされるリソースとして、IBM MQ キューを指定できます。

関連資料

[550 ページの『MFT agent.properties ファイル』](#)

各 Managed File Transfer Agent には、独自のプロパティ・ファイル `agent.properties` があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。`agent.properties` ファイルには、エージェントの動作を変更するプロパティを含めることもできます。

トリガー・ファイルの使用

リソース・モニター内のトリガー・ファイルの内容を使用して、単一の転送要求で転送するファイルのセットを定義することができます。一致するトリガー・ファイルが検出されるたびに、ソース・ファイル・パスに関して (オプションで宛先ファイル・パスに関して) その内容が解析されます。次いで、それらのファイル・パスを使用して、ユーザーが指定するタスク転送 XML ファイル内のファイル項目が定義され、それが単一の転送要求としてエージェントに実行依頼されます。リソース・モニターの定義により、トリガー内容が使用可能かどうかが決まります。

-tc (トリガー内容) パラメーターを指定することによって、モニターを作成するときに、ファイル内容によるトリガー発行を使用可能にできます。この **-tc** パラメーターは、ファイル・トリガー・オプションの `match` および `noSizeChange` にのみ適用されます。モニターの作成の詳細については、[413 ページの『fteCreateMonitor: MFT リソース・モニターの作成』](#)を参照してください。

トリガー・コンテンツ・ファイルを使用する場合、各行のデフォルトの形式は次のいずれかです。

- 単一のソース・ファイル・パス、または
- コンマで区切られたソース・ファイル・パスと宛先ファイル・パス

ここで、空白文字はファイル・パスの一部として処理されます。**fteCreateMonitor** コマンドで **-tcr** パラメーターと **-tcc** パラメーターを指定することにより、デフォルトの行形式を変更することができます。詳しくは、[190 ページの『詳細オプション』](#)を参照してください。

トリガー・ファイルが解析された後、ファイル・パスのリストが生成され、ユーザーが指定した転送タスク XML に適用されます。すべてのモニターの場合と同様に、転送タスク XML の形式は、単一の項目またはファイルが定義された **fteCreateTransfer** コマンドによって生成される、完全な転送タスク XML です。単一項目では、ソースおよび宛先ファイルのパスで置き換えるものとして、置換変数 `${contentSource}` と、オプションで `${contentDestination}` を使用する必要があります。モニターは転送タスク XML を拡張して、トリガー・ファイル内の行ごとにファイル項目 (ファイル・パス) が含まれるようにします。

-tc パラメーターはトリガー・ファイルごとに 1 つの転送要求を暗黙指定するので、**-bs** パラメーターと共にファイル内容によるトリガー発行を使用することはできません。

例

以下の例では、`trig` で終わるファイル上でトリガーするモニターを定義し、そのファイル内のファイル・パスを読み取ります。

```
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 コマンドは、ソース・ファイル・パスが `${sourceContent}` である単一ファイルに対して、`task.xml` というファイルを作成します。以下に例を示します。

```
<item checksumMethod="MD5" mode="binary">
  <source disposition="leave" recursive="false">
    <file>${contentSource}</file>
  </source>
</item>
```

fteCreateMonitor コマンドは、`/home/trigdir` ディレクトリーにあって `trig` で終わるファイルをスキャンし、その内容を使用して、そのトリガー・ファイル内のすべてのパスに対し、`task.xml` に基づく

単一転送要求を作成します。トリガー・ファイルの必須の形式は、各行に1つのファイル・パス(ソースのみ)で、コンマ分離文字はありません。以下に例を示します。

```
/home/file/first.txt
/home/file/second.txt
/home/different/third.txt
:
```

すべてのファイルは、ファイル・パスではなくファイル名だけで /file/destdir ディレクトリーに送達されます。つまり、/home/file/first.txt は /file/destdir/first.txt に送達されます。

あるいは、**fteCreateTransfer** コマンドの **-dd /file/destdir** パラメーターを **-df \$ {contentDestination}** に変更し、トリガー・ファイルの内容の形式を *source file path,destination file path* とすれば、同じ宛先エージェントに対して異なる宛先パスを定義できます。以下に例を示します。

```
/home/file/first.txt,/home/other/sixth.txt
```

これによって、宛先の場所は /home/other/sixth.txt になります。

置換変数はトークン化することができます。例えば、 **\${contentDestination{token=-1}}** を使用して、ファイル名の部分を、提供されているパスから分離することができます。したがって、**fteCreateTransfer** の宛先が **-df /file/destdir/\${contentDestination{token=-1}}** と定義されている場合、/home/file/first.txt の新しい宛先は /file/destdir/sixth.txt となります。

詳細オプション

-tcr regex パラメーターを使用することにより、トリガー・ファイルの内容のデフォルト行形式を変更することができます。必要な行形式に一致する正規表現を提供し、1つか2つのキャプチャー・グループを提供します。最初のキャプチャー・グループはソースで、2番目のオプションのキャプチャー・グループは宛先です。以下に例を示します。

- ソースと宛先のパスをハイフンで分離します。

```
((?:[^-]+)-((?:[^-]+))
```

この例では、分離文字は3つの場所で定義されており、ハイフン(-)の3つの出現個所のすべては任意の文字に変更することができます。特殊文字は、必ずすべてエスケープしてください。

- ソースと宛先のパスを、コンマと後続のスペースで分離します。番号記号(#)で示されるコメントは無視されます。

```
((?:[^\, ]+),((?:[^\, ]+)) *(?:#.*)+
```

ファイル・パスに番号記号(#)を含めることはできません。通常、エントリーは /home/source/from.txt,/home/destination/to.txt # some comment のようになります。

-tcr パラメーターを使用する場合には、式でエラーを検出できるよう、また正しくトリガー・ファイルを解析できるよう、正規表現が適切に設計され、テストされていることを確認してください。

-tcc destSrc パラメーターを使用することにより、キャプチャーの順序を逆にすることができます。このパラメーターを指定する場合、最初のキャプチャー・グループは宛先ファイル・パス、2番目のグループはソース・ファイル・パスになります。

エラーの処理方法

空のトリガー・ファイル

トリガー・ファイルが空である場合、結果としてファイル転送が行われません。つまり、モニターは転送要求を作成しますが、ファイル項目は何も指定されません。

エラーのあるトリガー・ファイル

トリガー・ファイル内の何らかのエントリーが、期待される形式に照らして解析に失敗した場合、転送要求は生成されません。モニター・エラー・ログがパブリッシュされ、イベント・ログにもエラーが記録されます。トリガー・ファイルは処理済みとしてマークを付けられ、そのファイルが更新されるまでは、モニターはファイルを再度処理しようとはしません。

一致しない転送タスク XML

転送タスク XML はトリガー・ファイルに一致している必要があります。つまり、転送タスク XML に `sourceContent` と `destinationContent` の両方が入っている場合、そのモニター用のすべてのトリガー・ファイルにも、ソースと宛先のファイル・パスが入っている必要があります。逆の場合も同様です。最初のケースで、トリガー・ファイルにソース・ファイル・パスしか指定されていない場合、モニターは `destinationContent` の置換失敗を報告します。

例

以下の例は、トリガー・ファイルの内容がソース・ファイル・パスだけである、基本的な内容トリガーです。

```
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 パラメーターは、スペース文字で分離された任意の文字のシーケンスのキャプチャー・グループを 2 つ定義します。**-tcc destSrc** パラメーターおよびオプションは、キャプチャー・グループがまず宛先として、それからソースとして処理されることを示します。

```
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 MFT リソース・モニターのロギング

IBM MQ 9.0.3 Managed File Transfer より、リソース・モニターに関する診断情報を取得するための方法が改良されています。

このタスクについて

IBM MQ 9.0.3 以降、リソース・モニターでロギングを使用できます。**fteSetAgentLogLevel** コマンドまたは `agent.properties` ファイルを使用して、リソース・モニターのロギングを制御できます。

情報の取得に既存のトレース・ポイントが引き続き使用されることに注意してください。

リソース・モニター・ログは、`resmoneventN.log` という名前のファイルに書き込まれます。ここで、*N* は数値を表します。例えば、`resmonevent0.log` です。



重要: 1つのエージェントのリソース・モニターはすべて同じログ・ファイルに書き込まれます。

`resmoneventN.log` ファイルの出力例については、315 ページの『[MFT ディレクトリー・リソース・モニターがファイルをトリガーしない場合の対応策](#)』を参照してください。

次の表に、リソース・モニターがログ・ファイルに書き込むイベントのタイプをリストします。3 番目の列は、各イベントの取り込みに必要なログ・レベルを示しており、最低レベルが `INFO` で、最高レベルが `VERBOSE` です。

高いログ・レベルを設定すると、それより低いレベルのイベントも書き込まれることに注意してください。例えば、ログ・レベルを `MODERATE` に設定すると、`INFO` レベルのイベントも書き込まれますが、`VERBOSE` レベルのイベントは書き込まれません。

Number	イベント	ログ・レベル	説明
1	モニターの作成	INFO	リソース・モニターが作成されました。
2	モニターの削除	INFO	リソース・モニターが削除されました。
3	モニターの停止	INFO	リソース・モニターが停止されました。
4	モニターの開始	INFO	リソース・モニターが開始されました。
5	モニターによるポーリングの開始	INFO	リソース・モニターが新しいポーリング周期を開始しました。
6	モニターによるポーリングの終了	INFO	リソース・モニターのポーリング周期が終了しました。
7	パターンの一致	VERBOSE	トリガー・モニター・ディレクトリー上のファイル、または指定されたパターンに一致するキュー内のメッセージが見つかりました。
8	パターンの不一致	VERBOSE	トリガー・モニター・ディレクトリー上の一致しないファイル、または指定されたパターンに一致しないキュー内のメッセージが見つかりました。
9	転送の実行依頼	INFO	リソース・モニターにより転送が開始されました。
10	ディレクトリーが深すぎる	VERBOSE	リソース・モニターによってモニターされるディレクトリーに、リソース・モニター構成で指定された数より多くのポーリング対象サブディレクトリーが含まれています。
11	ファイルのロック	MODERATE	リソース・モニターがモニターするトリガー・ファイルが、別のプロセスによってロックされています。
12	ファイル・サイズが小さい	MODERATE	トリガー・ファイルが、リソース・モニター構成で指定されているサイズより小さいサイズです。
13	ファイル・サイズが固定されていない	MODERATE	トリガー・ファイルが、リソース・モニター構成で予想されるより頻繁に変更されています。
14	ポーリングが多すぎる	MODERATE	リソース・モニターによるサイズが固定されていないトリガー・ファイルのポーリング回数が多すぎます。
15	一致する項目	INFO	リソース・モニターによってポーリングされるディレクトリーで見つかったトリガー・ファイルの総数。
16	転送中の項目	INFO	転送要求内の項目の総数。
17	FDC または例外	MODERATE	リソース・モニターが例外を生成しました。
18	転送要求	INFO	リソース・モニターにより転送要求が実行依頼されました。
19	モニター開始の失敗	MODERATE	リソース・モニターの開始に失敗しました。

手順

- **fteSetAgentLogLevel** を使用してリソース・モニターのログのオン/オフを切り替える場合は、[fteSetAgentLogLevel](#) を参照して、**logMonitor** パラメーターの説明や各種オプションの使用例を確認してください。
- `agent.properties` ファイルを使用してリソース・モニターのログを制御する場合は、550 ページの『MFT agent.properties ファイル』を参照して、以下のログ・アクティビティーを実行するための追加プロパティーの説明を確認してください。
 - ログをオンまたはオフにする
 - 各ログ・ファイルのサイズを制限する
 - リソース・モニターで生成可能なログの数を制限する

例

次のメッセージ例は、キュー・マネージャー MFTDEMO 上のエージェント HA2 に verbose レベルのログを設定します。

```
<?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>
```

関連資料

[fteSetAgentLogLevel コマンド](#)

fteSetAgentLogLevel コマンドを使用して、Managed File Transfer プロトコル・ブリッジ・エージェントと FTP/SFTP/FTPS ファイル・サーバーの間のファイル転送に関する診断情報のログを有効または無効にします。

550 ページの『MFT agent.properties ファイル』

各 Managed File Transfer Agent には、独自のプロパティー・ファイル `agent.properties` があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。`agent.properties` ファイルには、エージェントの動作を変更するプロパティーを含めることもできます。

MFT リソース・モニターのバックアップとリストア

リソース・モニターをバックアップしておけば、後で使用することができます。そのためには、リソース・モニターの定義を XML ファイルにエクスポートします。エクスポートした定義をインポートすれば、バックアップから新しいリソース・モニターを作成できます。

このタスクについて

以前に定義したリソース・モニターをバックアップしておけば、後からその定義を再利用することができます。例えば、別のインフラストラクチャーでリソース・モニターを再作成する場合や、キュー・マネージャーの問題が原因でリソース・モニターの再作成が必要になる場合などが考えられます。

1つのリソース・マネージャー定義をバックアップするには、**fteCreateMonitor** コマンドまたは **fteListMonitors** コマンドで **-ox** パラメーターを使用します。どちらの場合も、リソース・マネージャー定義を XML ファイルにエクスポートすることでバックアップを作成します。**fteCreateMonitor** コマンドの **-ix** パラメーターを使用してその定義を XML ファイルからインポートすれば、新しいリソース・マネージャーを作成できます。

-ox パラメーターを使用する場合は、一度に1つのリソース・モニター定義しかバックアップできません。

V 9.0.5 IBM MQ 9.0.5 以降、**-od** パラメーターが **ftelistmonitors** コマンドに追加されました。このパラメーターを指定すれば、複数のリソース・モニター定義を指定のディレクトリーに一括してエクスポートすることで、バックアップを一度に作成できます。各リソース・モニター定義は、*agent name.monitor name.xml* の形式で名前が指定された別個の XML ファイルに保存されます。

V 9.0.5 バックアップするリソース・モニターが多数存在する場合は、**-od** パラメーターが非常に便利です。リソース定義ごとに **ftelistmonitors -ox** コマンドを別々に実行したり、リソース・モニターごとに別々のスクリプトで **ftelistmonitors -ox** コマンドを実行したりする代わりに、**ftelistmonitors -od** コマンドを1回実行するだけです。

手順

- 1つのリソース・モニター定義を XML ファイルにエクスポートしてバックアップを作成するには、以下のいずれかのコマンドを使用します。

– **-ox** パラメーターを指定した **ftecreatemonitor** コマンド。

– **-ox** パラメーターを指定した **ftelistmonitors** コマンド。

-ox パラメーターを使用する場合は、以下の例のように **-ma** パラメーターと **-mn** パラメーターも指定する必要があります。

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

- **V 9.0.5** 複数のリソース・モニター定義を、指定されたディレクトリー内の XML ファイルにエクスポートすることによってバックアップするには、以下の例に示すように、**-od** パラメーターを指定して **ftelistmonitors** コマンドを使用します。

```
ftelistmonitors -od /usr/mft/resmonbackup
```

複数のリソース・モニターを一括してバックアップする場合は、有効なターゲット・ディレクトリーを指定する必要があります。ターゲット・パスを指定しないと、以下の例のようなエラー・メッセージが表示されます。

BFGCL0762E: 出力ディレクトリーが指定されていません。 有効なパスを指定して、コマンドを再実行してください。

-od パラメーターを **-ox** パラメーターと組み合わせることはできません。組み合わせると、以下のようなエラー・メッセージが表示されます。

BFGCL0761E: '-od' パラメーターと '-ox' パラメーターを両方一緒に指定することは無効です。

バックアップに含めるリソース・モニターのセットを定義できます。例えば、以下の例のように **-ma** パラメーターでエージェントの名前を指定すれば、そのエージェントのすべてのリソース・モニターをバックアップできます。

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

エージェント名とモニター名のいずれかまたは両方と突き合わせるために使用するパターンの定義時には、アスタリスク文字 (*) を組み込んだワイルドカード・マッチングも使用できます。以下の例では、指定のパターンに合致する名前のエージェントにある、指定のパターンに合致する名前のすべてのリソース・モニターをバックアップします。

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

コマンドの実行中に、以下のような進行状況レポート・メッセージが表示されます。

合計 *number* 個の一致するリソース・モニター定義が見つかりました。

number 個中 *index* 個のリソース・モニター定義がファイル・システムに保存されました。

詳細オプションを使用した場合でも合計実行数は表示されますが、以下の部分の代わりに、

number 個中 *index* 個のリソース・モニター定義がファイル・システムに保存されました

保存されるリソース・モニター定義の名前がこのコマンドで表示されます。例えば、以下のようになります。

BFGCL0762I: エージェント'XFERAGENT'のモニター'FILEMON'の定義が FILEMON.XFERAGENT.XML としてファイル・システムに保存されました。

V9.0.5

特定のエージェントの1つのリソース・モニターを、指定されたディレクトリー内のXMLファイルにエクスポートすることによってバックアップするには、**-od** パラメーターを指定して

fteListMonitors コマンドを使用します。

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

-od パラメーターを使用して1つのリソース・モニターをバックアップすると、**-ox** パラメーターを使用した場合と似たような結果になりますが、出力ファイル名が *agent name.monitor name.xml* という形式になります。

- バックアップからリソース・モニター定義をリストアする場合は、以下の例のように **-ix** パラメーターを付けて **fteCreateMonitor** コマンドを使用します。

```
fteCreateMonitor -ix file name
```

-od パラメーターの使用法のその他の例については、[476 ページの『fteListMonitors: MFT リソース・モニターのリスト』](#)を参照してください。

関連資料

[413 ページの『fteCreateMonitor: MFT リソース・モニターの作成』](#)

fteCreateMonitor コマンドは、コマンド行から新規リソース・モニターを作成して開始します。

Managed File Transfer を使用してリソース (例えば、ディレクトリーの内容) をモニターして、トリガー条件が満たされたときに、指定したタスク (ファイル転送など) を開始することができます。

[476 ページの『fteListMonitors: MFT リソース・モニターのリスト』](#)

コマンド行から **fteListMonitors** コマンドを使用すると、Managed File Transfer ネットワークの既存のリソース・モニターをすべてリストできます。

ファイル転送テンプレートの処理

ファイル転送テンプレートを使用すると、繰り返しの転送または複雑な転送を行うための共通のファイル転送設定を保管できます。転送テンプレートは **fteCreateTemplate** コマンドを使用してコマンド行から作成します。また、IBM MQ Explorer で、「ファイル転送管理のテンプレート新規作成」ウィザードを使用して転送テンプレートを作成することも、ファイル転送の作成時に「転送設定をテンプレートとして保存する」チェック・ボックスを選択してテンプレートを保存することもできます。「転送テンプレート」ウィンドウには、Managed File Transfer ネットワーク内に作成した転送テンプレートがすべて表示されます。

このタスクについて

コマンド行から転送テンプレートを作成するには、**fteCreateTemplate** コマンドを使用します。その後、コマンド行で作成した転送テンプレートを実行依頼するには、IBM MQ Explorer の「実行依頼」をクリックします。

IBM MQ Explorer で転送テンプレートを表示するには、以下のステップを実行します。

手順

- 「ナビゲーター」ビューで「ファイル転送管理」を展開します。「ファイル転送管理 - メイン」が「コンテンツ」ビューに表示されます。
- すべての調整キュー・マネージャーが「ナビゲーター」ビューにリストされます。スケジュール済みの転送に使用した調整キュー・マネージャーの名前を展開します。接続先の調整キュー・マネージャーを変更する場合は、「ナビゲーター」ビューで使用する調整キュー・マネージャーの名前を右クリックして、「接続」をクリックします。

3. 「転送テンプレート」をクリックします。「転送テンプレート」ウィンドウが「コンテンツ」ビューに表示されます。
4. 「転送テンプレート」ウィンドウには、ファイル転送に関する以下の詳細がリストされます。
 - a) 「名前」。ファイル転送テンプレートの名前。
 - b) 「ソース」。ソース・システムからファイルを転送するために使用するエージェントの名前。
 - c) 「ソース・ファイル」。ホスト・システムにおける、転送するファイルの名前。
このフィールドを表示するには、転送テンプレート情報を展開する必要があります。
 - d) 「宛先」。宛先システムでファイルを受け取るために使用するエージェントの名前。
 - e) 「宛先ファイル」。宛先システムに転送された後のファイルの名前。
このフィールドを表示するには、転送テンプレート情報を展開する必要があります。
 - f) 「スケジュール済みの開始時刻 (選択したタイム・ゾーン)」。ファイル転送を開始するようスケジュールされた、管理者が使用するタイム・ゾーンでの時刻と日付。表示されるタイム・ゾーンを変更するには、**ウィンドウ > 設定 > IBM MQ Explorer > Managed File Transfer** をクリックし、**タイム・ゾーン**: リストから別のタイム・ゾーンを選択します。「OK」をクリックします。
 - g) 「トリガー・イベント」。ファイル転送を起動して開始させるイベントのタイプ。タイプは次のいずれかの値になります。存在、存在しない、または超過。

タスクの結果

「転送テンプレート」ウィンドウに表示されている内容を最新表示するには、「コンテンツ」ビューのツールバーにある「リフレッシュ」ボタン  をクリックします。

転送テンプレートを実行依頼し、テンプレートで定義されている転送を開始するには、テンプレート名を右クリックし、「実行依頼」をクリックします。

転送テンプレートを変更するには、テンプレート名を右クリックして「編集」をクリックします。元のテンプレートに含まれているすべてのファイルが転送グループの一部として表示されます(それらのファイルが元のテンプレートでグループの一部として組み込まれていない場合でも、そのような動作になります)。テンプレートからファイルを削除する場合は、グループからそのファイル指定を選択し、「選択した項目を削除」をクリックする必要があります。テンプレートに新しいファイル指定を追加する場合は、テンプレート・パネルにあるフィールドを使用して、「グループに追加」ボタンをクリックします。編集を行うと、編集済みテンプレートに新しい名前を付けるように求められます。

転送テンプレートからファイル転送を作成するには、テンプレート名を右クリックして「新規の転送として編集」をクリックします。

転送テンプレートの複製コピーを作成するには、テンプレート名を右クリックして「複製」をクリックします。重複転送テンプレートは、元のテンプレートと同じ名前でも自動的に保存され、「(コピー)」に追加されます。

転送テンプレートを削除するには、テンプレート名を右クリックして「削除」をクリックします。

関連タスク

[197 ページの『IBM MQ Explorer を使用したファイル転送テンプレートの作成』](#)

ファイル転送テンプレートを IBM MQ Explorer またはコマンド行から作成することができます。その後そのテンプレートを使用してそのテンプレート詳細を使用する新規ファイル転送を作成したり、そのテンプレートを送信してファイル転送を開始したりできます。

関連資料

[421 ページの『fteCreateTemplate: 新規ファイル転送テンプレートの作成』](#)

fteCreateTemplate コマンドは、将来の使用のために保持できるファイル転送テンプレートを作成します。必須パラメーターは **-tn (template_name)** パラメーターのみです。その他のパラメーターはすべてオプションです。ただし、ソース・ファイル仕様を指定する場合には、宛先ファイルも提供する必要があります。同様に、宛先ファイルを指定する場合には、ソース・ファイル仕様も指定する必要があります。

[480 ページの『fteListTemplates』](#)

fteListTemplates コマンドを使用して、調整キュー・マネージャーで使用できる Managed File Transfer の転送テンプレートをリスト表示します。

470 ページの『[fteDeleteTemplates](#)』

fteDeleteTemplates コマンドを使用して、調整キュー・マネージャーから既存の Managed File Transfer テンプレートを削除します。

IBM MQ Explorer を使用したファイル転送テンプレートの作成

ファイル転送テンプレートを IBM MQ Explorer またはコマンド行から作成することができます。その後そのテンプレートを使用してそのテンプレート詳細を使用する新規ファイル転送を作成したり、そのテンプレートを送信してファイル転送を開始したりできます。

このタスクについて

ファイル転送テンプレートをコマンド行から作成するには、[fteCreateTemplate](#) コマンドを使用します。

IBM MQ Explorer の「ファイル転送管理のテンプレート新規作成」ウィザードを使用してファイル転送テンプレートを作成するには、以下のステップを実行します。

手順

1. 「ナビゲーター」ビューで、「ファイル転送管理」をクリックします。「ファイル転送管理 - メイン」が「コンテンツ」ビューに表示されます。
2. すべての調整キュー・マネージャーが「ナビゲーター」ビューに表示されます。スケジュール済みの転送に使用した調整キュー・マネージャーの名前を展開します。接続先の調整キュー・マネージャーを変更する場合は、「ナビゲーター」ビューで使用する調整キュー・マネージャーの名前を右クリックして、「接続」をクリックします。
3. 「転送テンプレート」を右クリックしてから「テンプレートの新規作成」をクリックして、「ファイル転送管理のテンプレート新規作成」ウィザードを開始します。
4. ウィザード・パネルの指示に従います。各パネルにはコンテキスト・ヘルプがあります。Windows 上でコンテキスト・ヘルプにアクセスするには、F1 キーを押します。Linux 上では、Ctrl+F1 キーまたは Shift+F1 キーを押します。

転送に関する必要な全詳細を含むテンプレートを作成してある場合、「転送の要約」ページで「転送設定をテンプレートとして保存する」チェック・ボックスがまだ選択されていない場合には、このチェック・ボックスを必ず選択します。また、「名前」フィールドにテンプレートの名前を入力します。転送に関する必要な全詳細がまだ含まれていないテンプレートを作成している場合、「転送設定をテンプレートとして保存する」チェック・ボックスに自動的にチェック・マークが付けられます。

関連タスク

195 ページの『[ファイル転送テンプレートの処理](#)』

ファイル転送テンプレートを使用すると、繰り返しの転送または複雑な転送を行うための共通のファイル転送設定を保管できます。転送テンプレートは **fteCreateTemplate** コマンドを使用してコマンド行から作成します。また、IBM MQ Explorer で、「ファイル転送管理のテンプレート新規作成」ウィザードを使用して転送テンプレートを作成することも、ファイル転送の作成時に「転送設定をテンプレートとして保存する」チェック・ボックスを選択してテンプレートを保存することもできます。「転送テンプレート」ウィンドウには、Managed File Transfer ネットワーク内に作成した転送テンプレートがすべて表示されます。

関連資料

421 ページの『[fteCreateTemplate: 新規ファイル転送テンプレートの作成](#)』

fteCreateTemplate コマンドは、将来の使用のために保持できるファイル転送テンプレートを作成します。必須パラメーターは **-tn** (*template_name*) パラメーターのみです。その他のパラメーターはすべてオプションです。ただし、ソース・ファイル仕様を指定する場合には、宛先ファイルも提供する必要があります。同様に、宛先ファイルを指定する場合には、ソース・ファイル仕様も指定する必要があります。

480 ページの『[fteListTemplates](#)』

fteListTemplates コマンドを使用して、調整キュー・マネージャーで使用できる Managed File Transfer の転送テンプレートをリスト表示します。

470 ページの『[fteDeleteTemplates](#)』

fteDeleteTemplates コマンドを使用して、調整キュー・マネージャーから既存の Managed File Transfer テンプレートを削除します。

ファイルからメッセージへのデータ転送

Managed File Transfer のファイルからメッセージへの転送機能を使用すれば、1つのファイルにあるデータを IBM MQ のキューにある1つまたは複数のメッセージに転送できます。

ファイルからメッセージへの転送およびメッセージからファイルへの転送を実行するには、転送のソース・エージェントと宛先エージェントの両方のバージョンが、IBM WebSphere MQ 7.5 以降であるか、IBM WebSphere MQ File Transfer Edition 7.0.3 以降である必要があります。メッセージからファイルへの転送に関しては、206 ページの『[メッセージからファイルへのデータ転送](#)』を参照してください。

ファイルからメッセージへの転送の宛先エージェントは、プロトコル・ブリッジ・エージェントまたは Connect:Direct ブリッジ・エージェントであることはできません。

ファイル・データを IBM MQ のメッセージ・データに転送できます。IBM MQ のメッセージは、各種アプリケーションで読み取ったり使用したりできます。ファイルからメッセージへの転送では、以下のタイプの転送がサポートされています。

- 1つのファイルから1つのメッセージへ。メッセージには、IBM MQ グループ ID が設定されていません。
- 1つのファイルから複数のメッセージへ (ファイルを指定の長さのメッセージに分割します)。すべてのメッセージには、同じ IBM MQ グループ ID が割り当てられます。
- 1つのファイルから複数のメッセージへ (テキスト・ファイルを Java 正規表現の区切り文字で分割します)。すべてのメッセージには、同じ IBM MQ グループ ID が割り当てられます。
- 1つのファイルから複数のメッセージへ (バイナリー・ファイルを 16 進数の区切り文字で分割します)。すべてのメッセージには、同じ IBM MQ グループ ID が割り当てられます。

区切り文字として一連のバイトを使用してバイナリー・ファイルを分割するには、**fteCreateTransfer** コマンドで **-sqdb** パラメーターを指定します。詳細については、『[-sqdb パラメーター](#)』を参照してください。

デフォルトでは、ファイルからメッセージへの転送で作成されるメッセージは、永続メッセージになります。そのメッセージを非永続メッセージに設定したり、永続性の値を宛先キューで定義したりすることも可能です。

ファイルを複数のメッセージに分割するように指定すると、同じファイルから作成されるすべてのメッセージには、同じ IBM MQ グループ ID が割り当てられます。ファイルを複数のメッセージに分割するように指定しない場合は、1つのメッセージのみがファイルから作成され、このメッセージには、IBM MQ グループ ID が設定されません。

ファイルを大きいメッセージに転送する場合、または多数の小さいメッセージに転送する場合は、IBM MQ または Managed File Transfer の一部のプロパティに変更が必要になる場合があります。これについては、322 ページの『[メッセージ・サイズに関連する MQ 属性および MFT プロパティを設定する際のガイド](#)』を参照してください。

注: 宛先キューがクラスター・キューであるか、クラスター・キューの別名である場合には、エージェント・プロパティ `enableClusterQueueInputOutput` が `true` に設定されていなければ、キューへのファイルの転送時にエラー・メッセージを受け取ります。詳しくは、318 ページの『[宛先キューがクラスター・キューであるか、クラスター・キューの別名である場合の対処法](#)』を参照してください。

ファイルからメッセージへの転送を実行するためのエージェントの構成

エージェントは、デフォルトで、ファイルからメッセージへの転送またはメッセージからファイルへの転送を実行できません。この機能を有効にするには、エージェント・プロパティ `enableQueueInputOutput` を `true` に設定する必要があります。IBM MQ・クラスター・キューへの書き込みを有効にするには、エージェント・プロパティ `enableClusterQueueInputOutput` も `true` に設定する必要があります。

このタスクについて

enableQueueInputOutput プロパティが true に設定されていない宛先エージェントに対して、ファイルからメッセージへの転送を実行しようとする、その転送は失敗します。調整キュー・マネージャーにパブリッシュされる転送ログ・メッセージには、以下のメッセージが組み込まれます。

```
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.
```

エージェントがキューへの書き込みと読み取りを行えるようにするには、以下のステップを実行します。

手順

1. **fteStopAgent** コマンドを使用して宛先エージェントを停止します。
2. `agent.properties` ファイルを編集して、`enableQueueInputOutput=true` という行を組み込みます。
`agent.properties` ファイルは、`MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/destination_agent_name` ディレクトリーにあります。
3. オプション: `agent.properties` ファイルを編集して、`enableClusterQueueInputOutput=true` という行を組み込みます。`agent.properties` ファイルは、`MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/destination_agent_name` ディレクトリーにあります。
4. **fteStartAgent** コマンドを使用して宛先エージェントを開始します。

例: 1つのファイルから1つのメッセージへの転送

fteCreateTransfer コマンドで `-dq` パラメーターを使用することにより、ファイル転送の宛先にするキューを指定できます。ソース・ファイルは、宛先キューで設定されている最大メッセージ長より小さいサイズでなければなりません。宛先キューは、宛先エージェントが接続するキュー・マネージャーと同じキュー・マネージャーにある必要はありませんが、これらの2つのキュー・マネージャー同士が通信できなければなりません。

このタスクについて

ソース・ファイルは、`/tmp/single_record.txt` という名前で、ソース・エージェント `AGENT_NEPTUNE` と同じシステムにあります。ソース・エージェント `AGENT_NEPTUNE` はキュー・マネージャー `QM_NEPTUNE` を使用します。宛先エージェントは `AGENT_VENUS` で、このエージェントはキュー・マネージャー `QM_VENUS` に接続します。宛先キュー `RECEIVING_QUEUE` は、キュー・マネージャー `QM_MERCURY` にあります。`QM_MERCURY` は、キュー・マネージャー `QM_VENUS` と同じ IBM MQ ネットワークにあり、そのキュー・マネージャーからのアクセスが可能です。

手順

次のコマンドを入力します。

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -dm QM_VENUS  
-dq RECEIVING_QUEUE@QM_MERCURY /tmp/single_record.txt
```

宛先エージェントが使用するキュー・マネージャーとは異なるキュー・マネージャーに宛先キューがある場合は、次の形式で `-dq` パラメーターの値を指定する必要があります。

`queue_name@queue_manager_name` この値に `@queue_manager_name` を指定しない場合、宛先エージェントは宛先キューが宛先エージェント・キュー・マネージャーにあると想定します。例外は、`enableClusterQueueInputOutput` エージェント・プロパティが true に設定されている場合です。この場合、宛先エージェントは標準の IBM MQ 解決手順を使用して、キューが置かれている場所を判別します。

ソース・エージェント `AGENT_NEPTUNE` はファイル `/tmp/single_record.txt` からデータを読み込み、このデータを宛先エージェント `AGENT_VENUS` に転送します。宛先エージェント `AGENT_VENUS` はその

データを RECEIVING_QUEUE@QM_MERCURY キューにある永続メッセージに送信します。メッセージには、IBM MQ グループ ID が設定されていません。

例: 1つのファイルを長さによって複数のメッセージに分割する操作

fteCreateTransfer コマンドの **-qs** パラメーターを使用することにより、1つのファイルを複数の IBM MQ メッセージに分割できます。ファイルを固定長の各セクションに分割し、各セクションをそれぞれのメッセージに書き込みます。

このタスクについて

ソース・ファイルは、/tmp/source.file という名前で、サイズは 36 KB です。ソース・ファイルは、ソース・エージェント AGENT_NEPTUNE と同じシステムにあります。ソース・エージェント AGENT_NEPTUNE はキュー・マネージャー QM_NEPTUNE に接続します。宛先エージェントは AGENT_MERCURY で、このエージェントはキュー・マネージャー QM_MERCURY に接続します。宛先キュー RECEIVING_QUEUE もキュー・マネージャー QM_MERCURY にあります。この転送では、ソース・ファイルを 1 KB のサイズのいくつかのセクションに分割し、各セクションを RECEIVING_QUEUE のメッセージに書き込みます。

手順

次のコマンドを入力します。

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY  
-dq RECEIVING_QUEUE -qs 1K /tmp/source.file
```

ソース・エージェント AGENT_NEPTUNE はファイル /tmp/source.file からデータを読み込み、このデータを宛先エージェント AGENT_MERCURY に転送します。宛先エージェント AGENT_MERCURY はそのデータを RECEIVING_QUEUE@QM_MERCURY キューの 36 個の 1 KB 永続メッセージに書き込みます。メッセージは、すべて同じ IBM MQ グループ ID を持ち、グループの最後のメッセージは IBM MQ LAST_MSG_IN_GROUP フラグ・セットを持ちます。

例: 正規表現区切り文字を使用してテキスト・ファイルを複数のメッセージに分割する操作

特定の Java 正規表現にマッチングするそれぞれの箇所で 1つのテキスト・ファイルを分割して複数のメッセージに転送します。そのために、**fteCreateTransfer** コマンドの **-dqdt** パラメーターを使用します。

このタスクについて

ファイルを可変長のセクションに分割し、各セクションをそれぞれのメッセージに書き込みます。特定の正規表現にマッチングするテキストの地点でテキスト・ファイルを分割します。ソース・ファイルは、/tmp/names.text という名前で、以下の内容が含まれています。

```
Jenny Jones,John Smith,Jane Brown
```

ファイルを分割する箇所を指定する正規表現は、コンマ文字 (,) です。

ソース・ファイルは、キュー・マネージャー QM_NEPTUNE に接続しているソース・エージェント AGENT_NEPTUNE と同じシステムにあります。宛先キュー RECEIVING_QUEUE は、キュー・マネージャー QM_MERCURY にあります。QM_MERCURY は、宛先エージェント AGENT_MERCURY が使用するキュー・マネージャーでもあります。この転送では、ソース・ファイルをいくつかのセクションに分割し、各セクションを RECEIVING_QUEUE のメッセージに書き込みます。

手順

次のコマンドを入力します。

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY
-dq RECEIVING_QUEUE -t text -dqdp postfix -dqdt "," /tmp/names.text
```

ソース・エージェント AGENT_NEPTUNE はファイル /tmp/names.text からデータを読み取り、このデータを宛先エージェント AGENT_MERCURY に転送します。宛先エージェント AGENT_MERCURY はデータをキュー RECEIVING_QUEUE の 3 つの永続メッセージに書き込みます。メッセージは、すべて同じ IBM MQ グループ ID を持ち、グループの最後のメッセージは IBM MQ LAST_MSG_IN_GROUP フラグ・セットを持ちます。

メッセージ内のデータは次のとおりです。

- 1 番目のメッセージ:

```
Jenny Jones
```

- 2 番目のメッセージ:

```
John Smith
```

- 3 番目のメッセージ:

```
Jane Brown
```

例: 正規表現区切り文字を使用してテキスト・ファイルを分割し、その区切り文字をメッセージに組み込む操作

1 つのテキスト・ファイルを、所定の Java 正規表現にマッチングするそれぞれの箇所で分割し、その正規表現の一致を結果に含めて、複数のメッセージに転送します。そのために、**fteCreateTransfer** コマンドの **-dqdt** パラメーターと **-qi** パラメーターを使用します。

このタスクについて

1 つのテキスト・ファイルを 1 つのキューにある複数のメッセージに転送します。ファイルを可変長のセクションに分割し、各セクションをそれぞれのメッセージに書き込みます。特定の正規表現にマッチングするテキストの地点でテキスト・ファイルを分割します。ソース・ファイルは、/tmp/customers.text という名前で、以下の内容が含まれています。

```
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
```

ファイルを分割する場所を指定する正規表現は、Customer\snumber:\s\d+ で、"Customer number: " の後に任意の桁数の数字が続くテキストにマッチします。コマンド行で指定する正規表現は、コマンド・シェルによって評価されないようにするために、二重引用符で囲む必要があります。正規表現は Java 正規表現として評価されます。詳細については [659 ページの『MFT が使用する正規表現』](#) を参照してください。

デフォルトでは、正規表現にマッチング可能な文字の数は、5 個に設定されています。この例で使用する正規表現にマッチングするのは、5 文字より長いストリングです。5 文字より長いマッチング項目を許可するには、エージェント・プロパティ・ファイルを編集して、**maxDelimiterMatchLength** プロパティを組み込みます。

デフォルトでは、正規表現にマッチングするテキストは、メッセージに組み込まれません。この例のように、正規表現にマッチングするテキストをメッセージに組み込むには、**-qi** パラメーターを使用します。ソース・ファイルは、キュー・マネージャー QM_NEPTUNE に接続しているソース・エージェント AGENT_NEPTUNE と同じシステムにあります。宛先キュー RECEIVING_QUEUE は、キュー・マネージャー QM_MERCURY にあります。QM_MERCURY は、宛先エージェント AGENT_MERCURY が使用するキュー・マネージャーでもあります。この転送では、ソース・ファイルをいくつかのセクションに分割し、各セクションを RECEIVING_QUEUE のメッセージに書き込みます。

手順

1. 次のコマンドを使用して、宛先エージェントを停止します。

```
fteStopAgent AGENT_MERCURY
```

2. AGENT_MERCURY のエージェント・プロパティ・ファイルに以下の行を追加します。

```
maxDelimiterMatchLength=25
```

注: **maxDelimiterMatchLength** の値を大きくすると、パフォーマンスが低下する可能性があります。

3. 次のコマンドを使用して、宛先エージェントを開始します。

```
fteStartAgent AGENT_MERCURY
```

4. 次のコマンドを入力します。

```
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
```

ソース・エージェント AGENT_NEPTUNE はファイル /tmp/customers.text からデータを読み込み、このデータを宛先エージェント AGENT_MERCURY に転送します。宛先エージェント AGENT_MERCURY はそのデータをキュー RECEIVING_QUEUE の 3 つの永続メッセージに書き込みます。メッセージは、すべて同じ IBM MQ グループ ID を持ち、グループの最後のメッセージは IBM MQ LAST_MSG_IN_GROUP フラグ・セットを持ちます。

メッセージ内のデータは次のとおりです。

- 1 番目のメッセージ:

```
Customer name: John Smith  
Customer contact details: john@example.net  
Customer number: 314
```

- 2 番目のメッセージ:

```
Customer name: Jane Brown  
Customer contact details: jane@example.com  
Customer number: 42
```

- 3 番目のメッセージ:

```
Customer name: James Jones  
Customer contact details: jjones@example.net  
Customer number: 26
```

例: ファイルからメッセージへの転送に関する IBM MQ メッセージ・プロパティの設定

fteCreateTransfer コマンドの **-qmp** パラメーターを使用して、転送によって宛先キューに書き込まれる最初のメッセージに IBM MQ メッセージ・プロパティを設定するかどうかを指定できます。IBM MQ メッセージ・プロパティを使用すれば、アプリケーションで IBM MQ メッセージ記述子 (MQMD) または MQRFH2 ヘッダーにアクセスしなくても、処理対象のメッセージを選択したり、メッセージに関する情報を取得したりすることが可能になります。

このタスクについて

fteCreateTransfer コマンドに **-qmp true** パラメーターを組み込みます。この例では、コマンドを実行依頼するユーザーの MQMD ユーザー ID は **larmer** です。

手順

次のコマンドを入力します。

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq MY_QUEUE@MyQM -qmp true
-t text /tmp/source_file.txt
```

宛先エージェント **AGENT_SATURN** からキュー・マネージャー **MyQM** のキュー **MY_QUEUE** に書き込まれる最初のメッセージの IBM MQ メッセージ・プロパティは、以下の値に設定されます。

```
usr.WMQFTETransferId=414cbaedefa234889d999a8ed09782395ea213ebbc9377cd
usr.WMQFTETransferMode=text
usr.WMQFTESourceAgent=AGENT_JUPITER
usr.WMQFTEDestinationAgent=AGENT_SATURN
usr.WMQFTEFileName=source_file.txt
usr.WMQFTEFileSize=1024
usr.WMQFTEFileLastModified=1273740879040
usr.WMQFTEFileIndex=0
usr.WMQFTEMQMDUser=larmer
```

例: ファイルからメッセージへの転送に関するユーザー定義プロパティの設定

ユーザー定義のメタデータが、転送で宛先キューに書き込まれる最初のメッセージで、IBM MQ メッセージ・プロパティとして設定されます。IBM MQ メッセージ・プロパティを使用すれば、アプリケーションで IBM MQ メッセージ記述子 (MQMD) または MQRFH2 ヘッダーにアクセスしなくても、処理対象のメッセージを選択したり、メッセージに関する情報を取得したりすることが可能になります。

このタスクについて

パラメーター **-qmp true** および **-md account=123456** を **fteCreateTransfer** コマンドに組み込んで、RFH2 ヘッダーの **usr.account** プロパティを **123456** に設定します。

手順

次のコマンドを入力します。

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq MY_QUEUE@MyQM
-qmp true -md account=123456 /tmp/source_file.txt
```

IBM MQ メッセージ・プロパティの標準セットに加えて、ユーザー定義のプロパティが、最初のメッセージのメッセージ・ヘッダーに設定されます。その最初のメッセージは、宛先エージェント

AGENT_SATURN により、キュー・マネージャー MyQM 上のキュー MY_QUEUE に書き込まれるものです。ヘッダーは次の値に設定されます。

```
usr.account=123456
```

ユーザー定義のメタデータの名前の先頭には、接頭部 `usr` が追加されます。

例: ファイルからメッセージへの転送のためのユーザー定義メッセージ・プロパティの追加

メッセージからファイルへの管理対象転送に Managed File Transfer を使用する場合には、結果のメッセージにユーザー定義のメッセージ・プロパティを含めることができます。

このタスクについて

カスタム・メッセージ・プロパティを定義するために、以下のいずれかの方式を使用することができます。

- 転送要求に `-md` パラメーターを指定します。詳しくは、203 ページの『例: ファイルからメッセージへの転送に関するユーザー定義プロパティの設定』を参照してください。
- Ant タスクを使用します。 `fte:filecopy` または `fte:filemove` のいずれかを使用できます。以下の例は `fte:filecopy` タスクです。

```
<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>
```

- リソース・モニターと変数置換を使用します。以下の例は、転送タスク XML を示しています。

```
<?xml version="1.0" encoding="UTF-8"?>
<monitor:monitor
xmlns:monitor="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="5.00"
xsi:schemaLocation="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition ./Monitor.xsd">
  <name>METADATA</name>
  <pollInterval units="minutes">5</pollInterval>
  <batch maxSize="5"/>
  <agent>AGENT1</agent>
  <resources>
    <directory recursionLevel="0">e:\temp</directory>
  </resources>
  <triggerMatch>
    <conditions>
      <allOf>
        <condition>
```

```

        <fileMatch>
          <pattern>*.txt</pattern>
        </fileMatch>
      </condition>
    </allOf>
  </conditions>
</triggerMatch>
<tasks>
  <task>
    <name/>
    <transfer>
      <request version="5.00"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
        <managedTransfer>
          <originator>
            <hostName>mqjason.raleigh.ibm.com.</hostName>
            <userID>administrator</userID>
          </originator>
          <sourceAgent QMgr="AGENTQM" agent="AGENT1"/>
          <destinationAgent QMgr="AGENTQM" agent="AGENT2"/>
          <transferSet priority="0">
            <metaDataSet>
              <metaData key="FileName">${FileName}</metaData>
            </metaDataSet>
            <item checksumMethod="MD5" mode="text">
              <source disposition="delete" recursive="false">
                <file>${FilePath}</file>
              </source>
              <destination type="queue">
                <queue persistent="true"
setMqProps="true">TEST.QUEUE@AGENTQM</queue>
              </destination>
            </item>
          </transferSet>
        </job>
        <name>Metadata_example</name>
      </job>
    </managedTransfer>
  </request>
</transfer>
</task>
</tasks>
<originator>
  <hostName>mqjason.raleigh.ibm.com.</hostName>
  <userID>administrator</userID>
</originator>
</monitor:monitor>

```

関連タスク

203 ページの『例: ファイルからメッセージへの転送に関する IBM MQ メッセージ・プロパティの設定』[fteCreateTransfer](#) コマンドの **-qmp** パラメーターを使用して、転送によって宛先キューに書き込まれる最初のメッセージに IBM MQ メッセージ・プロパティを設定するかどうかを指定できます。IBM MQ メッセージ・プロパティを使用すれば、アプリケーションで IBM MQ メッセージ記述子 (MQMD) または MQRFH2 ヘッダーにアクセスしなくても、処理対象のメッセージを選択したり、メッセージに関する情報を取得したりすることが可能になります。

関連資料

853 ページの『[fte:filecopy の Ant タスク](#)』

fte:filecopy タスクは、Managed File Transfer エージェント間でファイルをコピーします。ファイルはソース・エージェントから削除されません。

856 ページの『[fte:filemove の Ant タスク](#)』

fte:filemove タスクは、Managed File Transfer エージェント間でファイルを移動します。ファイルがソース・エージェントから宛先エージェントに正常に転送されると、そのファイルはソース・エージェントから削除されます。

ファイルからメッセージへの転送の失敗

ファイルからメッセージへの転送で、エージェントがファイル・データを宛先キューに書き込み始めた後に障害が発生すると、エージェントは、メッセージをコンシュームするアプリケーションに障害の発生を通知するためのメッセージをキューに書き込みます。

障害が発生した場合、以下のようなメッセージが宛先キューに書き込まれます。

- 内容はブランクです
- エージェントが宛先キューに書き込んだ直前のメッセージと同じ IBM MQ グループ ID が付きます
- IBM MQ の LAST_MSG_IN_GROUP フラグが設定されます
- メッセージ・プロパティが有効になっている場合は、追加の IBM MQ メッセージ・プロパティが組み込まれています。詳しくは、[678 ページの『障害プロパティ』](#)のトピックを参照してください。

例

以下のコマンドを実行して転送を要求します。

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq RECEIVING_QUEUE  
-qmp true -qs 1K /tmp/source1.txt
```

source1.txt ファイルは、48 KB です。この転送では、このファイルを 1 KB のメッセージに分割し、それらのメッセージを宛先キュー RECEIVING_QUEUE に書き込みます。

転送の進行中、エージェントが 16 個のメッセージを RECEIVING_QUEUE に書き込んだ後に、ソース・エージェントで障害が発生します。

エージェントは、ブランクのメッセージを RECEIVING_QUEUE に書き込みます。ブランクのメッセージでは、メッセージ・プロパティの標準セットに加えて、以下のメッセージ・プロパティが設定されます。

```
usr.WMQFTEResultCode = 40  
usr.WMQFTESupplement = BFGTR0036I: The transfer failed to complete successfully.
```

メッセージからファイルへのデータ転送

Managed File Transfer のメッセージからファイルへの転送機能を使用すれば、IBM MQ の 1 つのキューにある 1 つ以上のメッセージのデータを、1 つのファイル、1 つのデータ・セット (z/OS の場合)、または 1 つのユーザー・ファイル・スペースに転送できます。IBM MQ メッセージを作成または処理するアプリケーションがあれば、Managed File Transfer のメッセージからファイルへの転送機能を使用して、Managed File Transfer ネットワーク内の任意のシステムにあるファイルにメッセージを転送することができます。

ファイルからメッセージへの転送に関しては、[198 ページの『ファイルからメッセージへのデータ転送』](#)を参照してください。



重要: メッセージからファイルへの転送のソース・エージェントは、プロトコル・ブリッジ・エージェントまたは Connect:Direct ブリッジ・エージェントであることはできません。

IBM MQ のメッセージ・データをファイルに転送できます。メッセージからファイルへの転送では、以下のタイプの転送がサポートされています。

- 1 つのメッセージから 1 つのファイルへ
- 複数のメッセージから 1 つのファイルへ
- IBM MQ グループ ID が同じ複数のメッセージから 1 つのファイルへ
- 複数のメッセージから 1 つのファイルへ (各メッセージのデータの間にあるテキスト区切り文字またはバイナリー区切り文字をファイルに書き込みます)

ファイルを大きいメッセージから転送する場合、または多数の小さいメッセージから転送する場合は、IBM MQ または Managed File Transfer の一部のプロパティに変更が必要になる場合があります。詳しくは、[322 ページの『メッセージ・サイズに関連する MQ 属性および MFT プロパティを設定する際のガイダンス』](#)を参照してください。

V9.0.1 IBM MQ 9.0.1 以降、メッセージからファイルへの転送では、ソース・エージェントは、IBM MQ 9.0.1 より前のバージョンの IBM MQ では破壊的な GET とは異なり、ソース・キューからメッセージを参照します。すべてのメッセージ (メッセージのグループ化が使用されている場合はグループ内のすべてのメッセージ) が参照され、データが宛先ファイルに書き込まれた後に、メッセージはソース・キューから

削除されます。これにより、転送が失敗したり、キャンセルされたりした場合にメッセージがソース・キューに残ることができます。この変更のため、メッセージからファイルへの転送を実行するには、GET 権限と BROWSE 権限が必要になります。

V 9.0.0.2 IBM MQ 9.0.0 Fix Pack 2 以降、Managed File Transfer が更新され、転送要求 XML ペイロード内の転送 ID とグループ ID 属性の値の比較検査 (以前に [APAR IT18213](#) で削除されました) が復元されました。比較した 2 つの ID が等しい場合、ソース・エージェントはその ID を、メッセージからファイルへの転送のための入力キューに対して行われる 1 回目の MQGET の試行で、メッセージ ID のマッチ・オプション (グループ ID のマッチ・オプションと対照) として使用します。

メッセージからファイルへの転送を実行するためのエージェントの構成

デフォルトでは、エージェントがメッセージからファイルへの転送またはファイルからメッセージへの転送を実行することはできません。この機能を有効にするには、エージェント・プロパティ `enableQueueInputOutput` を `true` に設定する必要があります。

このタスクについて

`enableQueueInputOutput` プロパティが `true` に設定されていないソース・エージェントから、メッセージからファイルへの転送を実行しようとする、その転送は失敗します。調整キュー・マネージャーにパブリッシュされる転送ログ・メッセージには、以下のメッセージが組み込まれます。

```
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.
```

エージェントがキューへの書き込みと読み取りを行えるようにするには、以下のステップを実行します。

手順

1. **fteStopAgent** コマンドを使用してソース・エージェントを停止します。
2. `agent.properties` ファイルを編集して、`enableQueueInputOutput=true` という行を組み込みます。
`agent.properties` ファイルは、`MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/source_agent_name` ディレクトリにあります。
3. **fteStartAgent** コマンドを使用してソース・エージェントを開始します。

例: 1 つのキューから 1 つのファイルへの転送

fteCreateTransfer コマンドの `-sq` パラメーターを使用することにより、IBM MQ キューをファイル転送のソースとして指定できます。

このタスクについて

`START_QUEUE` キューにある 3 個のメッセージにソース・データが格納されています。このキューは、ソース・エージェントのキュー・マネージャー `QM_NEPTUNE` に存在していなければなりません。

手順

次のコマンドを入力します。

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

`START_QUEUE` キューにあるメッセージのデータが、`AGENT_VENUS` を実行しているシステムの `/out/three_to_one.txt` ファイルに書き込まれます。

このタスクについて

この例では、START_QUEUE キューに 3 個のメッセージがあるとします。このキューは、ソース・エージェントのキュー・マネージャー QM_NEPTUNE にあります。各メッセージのデータの後に挿入するバイナリー区切り文字は、16 進数バイトのコンマ区切りリストとして記述する必要があります (例: x34,xE7,xAE)。

手順

次のコマンドを入力します。

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -df /out/binary.file
                 -sqdp postfix -sqdb x34,xE7,xAE -sq START_QUEUE
```

ソース・エージェント AGENT_NEPTUNE によって、START_QUEUE にある 3 個のメッセージそれぞれのデータの後にバイナリー区切り文字が追加されます。このデータは、宛先ファイル /out/binary.file に書き込まれます。

キューのモニターおよび変数置換の使用

fteCreateMonitor コマンドを使用して、キューをモニターし、モニターしたキューからファイルにメッセージを転送できます。モニターされるキューから読み取られる最初のメッセージにある任意の IBM MQ メッセージ・プロパティの値をタスク XML 定義に置換して、転送動作の定義に使用できます。

このタスクについて

この例では、ソース・エージェントは AGENT_VENUS という名前であり、QM_VENUS に接続します。AGENT_VENUS がモニターするキューは START_QUEUE という名前であり、QM_VENUS にあります。エージェントは、キューを 30 分おきにポーリングします。

メッセージの完全に揃ったグループがキューに書き込まれると、モニター・タスクは、いくつかの宛先エージェントの 1 つのファイルにメッセージのグループを送信します。この宛先エージェントは、すべてキュー・マネージャー QM_MARS に接続しています。メッセージのグループが転送されるファイルの名前は、グループの最初のメッセージの IBM MQ メッセージ・プロパティ `usr.fileName` で定義します。メッセージのグループが送信されるエージェントの名前は、グループの最初のメッセージの IBM MQ メッセージ・プロパティ `usr.toAgent` で定義します。`usr.toAgent` ヘッダーが未設定の場合は、宛先エージェント用に使用されるデフォルト値は、AGENT_MAGENTA です。

`useGroups="true"` を指定する場合、`groupId="${GROUPID}"` を指定しないと、転送ではキュー内の最初のメッセージのみが取得されます。そのため、例えば変数置換を使用して `fileName` を生成した場合、`a.txt` の内容が正しくなくなる可能性があります。これは、`fileName` はモニターによって生成されますが、転送では、実際には `fileName` というファイルを生成するメッセージではなく、別のメッセージを取得するためです。

手順

1. モニター起動時にモニターが実行するタスクを定義するタスク XML を作成します。

```
<?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>

```

IBM MQ メッセージ・ヘッダーの値で置き換えられる変数は、**太字**で強調表示しています。このタスク XML は、ファイル /home/USER1/task.xml に保存されます。

- キュー START_QUEUE をモニターするリソース・モニターを作成します。
以下のコマンドを実行依頼します。

```

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

```

- ユーザーまたはプログラムは、メッセージのグループをキュー START_QUEUE に書き込みます。
このグループの最初のメッセージは、次の IBM MQ メッセージ・プロパティを設定しています。

```

usr.fileName=larmer
usr.toAgent=AGENT_VIOLET

```

- 完全に揃ったグループが書き込まれると、モニターが起動されます。エージェントは、IBM MQ メッセージ・プロパティをタスク XML に置換します。
この結果、タスク XML は以下のように変換されます。

```

<?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>

```

タスクの結果

タスク XML によって定義された転送が実行されます。AGENT_VENUS によって START_QUEUE から読み取られた、完全に揃ったメッセージ・グループが、AGENT_VIOLET を実行中のシステムの /reports/larmer.rpt というファイルに書き込まれます。

次のタスク

各メッセージの別個のファイルへの転送

キューをモニターして、すべてのメッセージが別個のファイルに転送されるようにする場合には、このトピックで前述した方法と同様の方法を使用することができます。

1. **fteCreateMonitor** コマンドに **-tr completeGroups** パラメーターを指定して、前述のようにモニターを作成します。
2. タスク XML で、次のように指定します。

```
<queue useGroups="true" groupId="${GROUPID}">START_QUEUE</queue>
```

ただし、メッセージをソース・キューに入れる場合は、それらのメッセージを IBM MQ グループには入れないでください。IBM MQ メッセージ・プロパティを各メッセージに追加します。例えば、メッセージごとに固有のファイル名の値を持つ `usr.filename` プロパティを指定します。こうすることで効果的に、Managed File Transfer Agent がソース・キュー内の各メッセージを異なるグループとして扱います。

例: IBM MQ メッセージ・プロパティを使用したメッセージからファイルへの転送の失敗

`usr.UserReturnCode` IBM MQ メッセージ・プロパティをゼロ以外の値に設定することによって、メッセージからファイルへの転送を失敗させることができます。さらに、`usr.UserSupplement` IBM MQ メッセージ・プロパティを設定することによって、失敗の理由に関する補足情報を指定することもできます。

このタスクについて

この例では、キュー `INPUT_QUEUE` とファイル `/home/user/output.file` との間で転送が進行中です。

ユーザーはメッセージを作成し、これをキュー `INPUT_QUEUE` の上に配置しています。ソース・エージェントはキュー `INPUT_QUEUE` からメッセージをコンシュームし、転送データを宛先エージェントに送信しています。宛先エージェントは、このデータをファイル `/home/user/output.file` に書き込んでいます。

メッセージをキュー `INPUT_QUEUE` に書き込んでいるユーザーは、進行中の転送を停止し、既に宛先ファイルに書き込まれたデータをすべて削除しようとしています。

手順

1. ユーザーは、次の IBM MQ メッセージ・プロパティを設定したメッセージをキュー `INPUT_QUEUE` に書き込みます。

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

2. ソース・エージェントは、IBM MQ メッセージ・プロパティを読み取り、キューからのメッセージの処理を停止します。宛先エージェントは、宛先ディレクトリーに書き込まれたファイル・データをすべて削除します。
3. ソース・エージェントは、転送の失敗を報告する転送ログ・メッセージを調整キュー・マネージャーに送信します。

このメッセージには、次の情報が含まれています。

```
<?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>

```

MFT エージェントのリスト

特定のキュー・マネージャーに登録された Managed File Transfer エージェントは、コマンド行または IBM MQ Explorer を使用してリストできます。

このタスクについて

コマンド行を使用してエージェントをリストするには、[fteListAgents コマンド](#)を参照してください。

IBM MQ Explorer を使用してエージェントをリストするには、「ナビゲーター」ビューで、調整キュー・マネージャー名の下に「エージェント」をクリックします。

エージェントが **fteListAgents** コマンドでリストされない場合、または IBM MQ Explorer に表示されない場合は、次のトピックの診断フローチャートを使用して、問題を見つけて修正してください。[エージェントが fteListAgents コマンドでリストされない場合](#)。

MFT エージェントの停止

Managed File Transfer エージェントはコマンド行から停止できます。エージェントを停止するときには、停止する前にエージェントを静置させて、エージェントが現行のファイル転送を完了するようにします。さらに、コマンド行で **-i** パラメーターを指定して、エージェントをただちに停止することもできます。エージェントが停止してしまうと、再始動するまでそのエージェントを使用してファイルを転送することはできません。

始める前に

キュー・マネージャーと関連付けられたエージェントの名前を確認する場合は、IBM MQ Explorer またはコマンド行を使用してエージェントをリストできます。これについては、[fteListAgents コマンド](#)を参照してください。

このタスクについて

コマンド行からエージェントを停止する場合は、[fteStopAgent](#) を参照してください。

エージェントを Windows サービスとして実行するように構成した場合、**fteStopAgent** コマンドを実行すると、Windows サービスが停止します。または、Windows の「サービス」ツールを使用してサービスを停止することによって、エージェントを停止できます。詳しくは、[155 ページの『Windows サービスとしての MFT エージェントの開始』](#)のトピックを参照してください。

z/OS 上の MFT エージェントの停止

JCL からの開始タスクとして z/OS 上で Managed File Transfer Agent を実行している場合、エージェントは、**fteStopAgent** コマンドに加えて、z/OS オペレーター・コマンド **MODIFY** および **STOP** を受け入れます。

開始タスクは、特定のユーザー ID の下で実行され、ユーザーのログオフに影響されないため、使用されません。

注: **V9.0.0.1** → **V9.0.2** 通常、開始済みタスクはログオン特権がない可能性がある管理ユーザーの下で実行されるため、エージェントを実行しているユーザーとして z/OS システムにログオンすることはできません。そのエージェントに対して、**fteStartAgent**、**fteStopAgent**、**fteSetAgentTraceLevel** の各コマンド、および **-d** パラメーターが指定されている **fteShowAgentDetails** コマンドを実行することはできません。

IBM MQ 9.0.2 および IBM MQ 9.0.0 Fix Pack 1 以降、z/OS 上の Managed File Transfer エージェントでエージェント・プロパティー **adminGroup** を使用できるようになりました。例えば、MFTADMIN というセキュリティ・マネージャー・グループを定義し、開始済みタスクのユーザー ID と管理者 TSO ID をこのグループに追加できます。エージェント・プロパティー・ファイルを編集し、**adminGroup** プロパティーの値をこのセキュリティ・マネージャー・グループの名前に設定します。

```
adminGroup=MFTADMIN
```

これにより、このグループのメンバーは、開始済みタスクとして実行されているエージェントに対して、**fteStartAgent**、**fteStopAgent**、**fteSetAgentTraceLevel** の各コマンド、および **-d** パラメーターが指定されている **fteShowAgentDetails** コマンドを実行できます。

詳しくは、**adminGroup** プロパティー (550 ページの『MFT agent.properties ファイル』) を参照してください。

z/OS MODIFY コマンド (F) を使用したエージェント・シャットダウンの制御

MODIFY コマンドを使用すると、**fteStopAgent** コマンドの代替手段として、制御された方法でエージェントを停止できます。エージェントは現在進行中の転送は完了しますが、新規転送は開始しません。

以下に例を示します。

```
F job_name,APPL=STOP
```

ここで、*job_name* はエージェント・プロセスを実行しているジョブです。

z/OS STOP コマンド (P) を使用した即時エージェント・シャットダウン

STOP コマンドは、**-i** パラメーターを指定して **fteStopAgent** コマンドを使用した場合の即時停止に相当します。エージェントは、現在ファイルを転送中であっても、即時停止されます。

以下に例を示します。

```
P job_name
```

ここで、*job_name* はエージェント・プロセスを実行しているジョブです。

関連資料

159 ページの『z/OS での MFT エージェントの開始』

z/OS でエージェントを実行するには、UNIX System Services セッションから **fteStartAgent** コマンドを実行する方法に加えて、JCL から開始済みタスクとして開始する方法もあり、この場合には対話式セッションを必要としません。

550 ページの『MFT agent.properties ファイル』

各 Managed File Transfer Agent には、独自のプロパティー・ファイル **agent.properties** があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。**agent.properties** ファイルには、エージェントの動作を変更するプロパティーを含めることもできます。

プロトコル・ブリッジ

プロトコル・ブリッジを使用すれば、Managed File Transfer (MFT) ネットワークから、MFT ネットワークの外部(ローカル・ドメインとリモート・ロケーションの両方)にあるファイル・サーバーに格納されているファイルにアクセスできます。このファイル・サーバーでは、FTP、FTPS、またはSFTP ネットワーク・プロトコルを使用できます。それぞれのファイル・サーバーで少なくとも1つの専用エージェントが必要です。この専用エージェントは、プロトコル・ブリッジ・エージェントとして知られています。ブリッジ・エージェントは、複数のファイル・サーバーと相互作用できます。

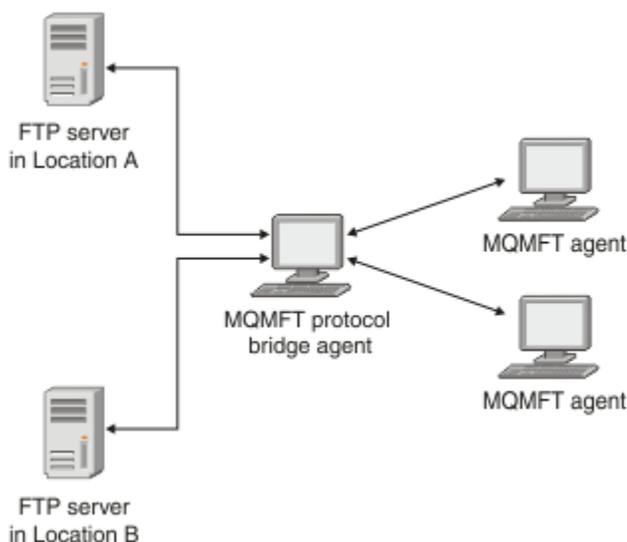
プロトコル・ブリッジは、Managed File Transfer のサービス・コンポーネントの一部として使用可能です。MFT を実行する単一のシステムに、さまざまなファイル・サーバーに接続する複数の専用エージェントを作成することができます。

プロトコル・ブリッジ・エージェントを使用して、複数のエンドポイントに同時にファイル転送を行うことができます。MFT には ProtocolBridgeProperties.xml と呼ばれるファイルがあるので、このファイルを編集して、ファイルの転送先にするさまざまなプロトコル・ファイル・サーバーを定義できます。**fteCreateBridgeAgent** コマンドは、ProtocolBridgeProperties.xml にデフォルトのプロトコル・ファイル・サーバーの詳細情報を追加します。このファイルについての説明は [577 ページの『プロトコル・ブリッジ・プロパティ・ファイルのフォーマット』](#)にあります。

プロトコル・ブリッジ・エージェントを使用して、以下のアクションを実行できます。

- MFT ネットワークからリモート・サーバーへのファイルのアップロード (FTP、FTPS、またはSFTP を使用)
- リモート・サーバーから MFT ネットワークへのファイルのダウンロード (FTP、FTPS、またはSFTP を使用)

注: プロトコル・ブリッジ・エージェントは、絶対ファイル・パスによってファイルへのアクセスを可能にする FTP、FTPS、またはSFTP サーバーのみをサポートできます。転送要求に相対ファイル・パスが指定されると、プロトコル・ブリッジ・エージェントは、プロトコル・サーバーへのログインに使用されたホーム・ディレクトリーが基準になっていると見なし、相対パスを絶対ファイル・パスに変換しようとします。現行ディレクトリーに基づいたファイルへのアクセスのみが可能なプロトコル・サーバーは、プロトコル・ブリッジ・エージェントではサポートされません。



この図は、異なるロケーションにある2つのFTPサーバーを示しています。FTPサーバーは、Managed File Transfer エージェントとファイルを交換するために使用されています。プロトコル・ブリッジ・エージェントは、FTPサーバーと、MFT ネットワークの残りの部分との間にあり、両方のFTPサーバーと通信するように構成されています。

プロトコル・ブリッジ・エージェントに加え、MFT ネットワークに別のエージェントがあることを確認します。プロトコル・ブリッジ・エージェントは、FTP、FTPS、またはSFTP サーバーに対してのみのブリッジであり、転送されたファイルをローカル・ディスクに書き込むことはありません。ファイルをFTP、

FTPS、または SFTP サーバーとの間で転送する場合は、プロトコル・ブリッジ・エージェントを (FTP、FTPS、または SFTP サーバーを代表する) ファイル転送の宛先またはソースとして使用し、別の標準エージェントを対応するソースまたは宛先として使用する必要があります。

プロトコル・ブリッジを使用してファイルを転送する場合、ブリッジは、転送するファイルが格納されているソースまたは宛先ディレクトリーを読み取るための権限を持っている必要があります。例えば、実行権限 (d--x--x--x) のみを持つディレクトリー /home/fte/bridge からファイルを転送すると、このディレクトリーから試行した転送は、以下のエラー・メッセージを表示して失敗します。

```
BFGBR0032E: Attempt to read filename from the protocol file server
has failed with server error 550. Failed to open file.
```

プロトコル・ブリッジ・エージェントの構成

プロトコル・ブリッジ・エージェントは、標準的な MFT エージェントに類似しています。

fteCreateBridgeAgent コマンドを使用してプロトコル・ブリッジ・エージェントを作成します。

ProtocolBridgeProperties.xml ファイルを使用してプロトコル・ブリッジ・エージェントを構成できます。このファイルについては、577 ページの『[プロトコル・ブリッジ・プロパティー・ファイルのフォーマット](#)』で説明しています。旧バージョンを使用している場合は、[拡張エージェント・プロパティー](#)に記載されている特定のプロトコル・ブリッジ・プロパティーを使用してエージェントを構成してください。すべてのバージョンで、222 ページの『[ファイル・サーバーの資格情報のマップ](#)』での説明に従って資格情報マッピングを構成することもできます。特定のプロトコル・ファイル・サーバー用にプロトコル・ブリッジ・エージェントを構成した後で、このエージェントをその他の目的で使用できません。

プロトコル・ブリッジのリカバリー

ファイル・サーバーが使用できないために、プロトコル・ブリッジ・エージェントがファイル・サーバーに接続できない場合、すべてのファイル転送要求は、ファイル・サーバーが使用可能になるまで、キューに入れます。エージェントが誤った資格情報を使用しているために、プロトコル・ブリッジ・エージェントがファイル・サーバーに接続できない場合、転送は失敗し、転送ログ・メッセージにこのエラーが反映されます。何らかの理由によってプロトコル・ブリッジ・エージェントが終了した場合、要求済みのファイル転送はすべて保持され、プロトコル・ブリッジが再始動すると、続きが処理されます。

ファイル転送中は、ファイルは通常、一時ファイルとして転送先に書き込まれ、転送が完了した時点でリネームされます。ただし、転送先が、書き込みを制限して構成されている (ユーザーはプロトコル・ファイル・サーバーにファイルをアップロードできるが、それらのアップロードされたファイルは決して変更できず、ユーザーが書き込めるのは事実上一度だけである) プロトコル・ファイル・サーバーの場合、転送されるファイルは転送先に直接書き込まれます。そのため、転送中に問題が発生した場合、一部しか書き込まれていないファイルが転送先プロトコル・ファイル・サーバーに残りますが、Managed File Transfer はこのようなファイルを削除も編集もできません。このシチュエーションでは、転送は失敗します。

ProtocolBridgeProperties.xml ファイルを使用したプロトコル・ファイル・サーバーのプロパティーの定義

ProtocolBridgeProperties.xml ファイルを使用して、ファイルの転送先および転送元の 1 つ以上のプロトコル・ファイル・サーバーのプロパティーを定義します。このファイルは、Managed File Transfer により提供され、エージェントの構成ディレクトリー内にあります。

このタスクについて

fteCreateBridgeAgent コマンドを使用すると、エージェント構成ディレクトリー `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` に ProtocolBridgeProperties.xml ファイルが作成されます。このコマンドの実行時にデフォルトが指定されている場合は、コマンドはこのファイル内にデフォルトのプロトコル・ファイル・サーバーの項目も作成します。

メッセージ BFGCL0392I で ProtocolBridgeProperties.xml ファイルの場所が示されます。

```
<?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\FTPSKeyStore.jks"
  trustStorePassword="password">

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

  </tns:ftpsServer>

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

    <tns:limits connectionTimeout="60"/>

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

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

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

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

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

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

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

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

```
-->
<tns:serverProperties xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeProperties
  ProtocolBridgeProperties.xsd">
  <!-- By default the location of the credentials file is in the home directory of the user
  that started the -->
  <!-- protocol bridge agent. If you wish to specify a different location use the
  credentialsFile element to -->
  <!-- do this. For
  example: -->
  <!-- <tns:credentialsFile path="/test/
  ProtocolBridgeCredentials.xml"/> -->
  <tns:defaultServer name="WINMVSCA.HURSLEY.IBM.COM"/>
  <tns:ftpServer name="WINMVSCA.HURSLEY.IBM.COM" host="WINMVSCA.HURSLEY.IBM.COM"
  platform="UNIX"
    timeZone="Europe/London" locale="en-GB" fileEncoding="US-ASCII"
    listFormat="unix" limitedWrite="false" />
  <!-- Define servers here -->
</tns:serverProperties>
```

このコマンドは、以下のメッセージ:BFGCL0532I を生成することがあります。

このエージェントが機能するためには、追加の資格情報ファイルを手動で作成する必要があります。デフォルトでは、このファイルは ProtocolBridgeCredentials.xml という名前で、ホームにあります。エージェントを開始するユーザーのディレクトリー。例えば、このユーザーがエージェントを開始したとします。場所は次のとおりです: \$HOME/ProtocolBridgeCredentials.xml

資格情報ファイルを使用する場合、次の点に注意してください。

1. 作成方法の詳細については、以下のテキストを参照してください。
2. 資格情報ファイルは、アクセス権が制限されたディレクトリー内になければなりません。例えば、他のユーザーの読み取りアクセスがないディレクトリーである必要があります。
3. エージェントを開始したユーザー ID の \$HOME 環境変数で資格情報ファイルのディレクトリーの場所を指定するか、ProtocolBridgeProperties.xml ファイルを編集して以下の場所でロケーションを指定します。

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

デフォルト以外のプロトコル・サーバーをさらに追加する場合は、このファイルを編集して、プロトコル・サーバーのプロパティを定義してください。この例では、追加のFTPサーバーを加えます。

注: プロトコル・ブリッジ・エージェントはファイル・ロックをサポートしていません。これは、Managed File Transfer がファイル・サーバーのファイル・ロック・メカニズムをサポートしていないためです。

手順

1. 以下の行を <tns:serverProperties>の子エレメントとしてファイルに挿入して、プロトコル・ファイル・サーバーを定義します。

```
<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. 次に、属性の値を変更します。

- name はプロトコル・ファイル・サーバーの名前です。
- host はプロトコル・ファイル・サーバーのホスト名または IP アドレスです。
- port はプロトコル・ファイル・サーバーのポート番号です。
- platform はプロトコル・ファイル・サーバーが実行されるプラットフォームです。
- timeZone はプロトコル・ファイル・サーバーを実行する時間帯です。
- locale はプロトコル・ファイル・サーバーで使用される言語です。
- fileEncoding はプロトコル・ファイル・サーバーの文字エンコードです。
- listFormat はプロトコル・ファイル・サーバーから戻されるファイルのリスト形式です。
- limitedWrite はファイル・サーバーに書き込みを行う際にデフォルト・モードに従うかどうかを判別します。デフォルト・モードでは、一時ファイルを作成し、転送が完了した後にそのファイルをリネームします。書き込み専用として構成されたファイル・サーバーの場合、ファイルは、最終的な名前をそのまま使用して作成されます。このプロパティの値は、true または false のいずれかになります。limitedWrite 属性と doNotUseTempOutputFile エージェント・プロパティは、プロトコル・ブリッジ・エージェントの場合に一緒に使用します。一時ファイルを使用する場合は、doNotUseTempOutputFile の値を設定せず、limitedWrite の値を false に設定する必要があります。これ以外の組み合わせで設定を行うと、一時ファイルは使用されません。
- maxListFileNames はプロトコル・ファイル・サーバー上のディレクトリーでファイル名をスキャンする際に収集される名前の最大数です。
- maxListDirectoryLevels はプロトコル・ファイル・サーバー上のディレクトリーでファイル名をスキャンする際に繰り返されるディレクトリー・レベルの最大数です。

これらの属性のデフォルト値や、これらの属性が必須かオプションかなど、これらの属性に関する詳細情報については、[577 ページの『プロトコル・ブリッジ・プロパティ・ファイルのフォーマット』](#)を参照してください。

関連資料

[577 ページの『プロトコル・ブリッジ・プロパティ・ファイルのフォーマット』](#)

エージェント構成ディレクトリー内の ProtocolBridgeProperties.xml ファイルは、プロトコル・ファイル・サーバーのプロパティを定義します。

[659 ページの『MFT が使用する正規表現』](#)

Managed File Transfer では、さまざまな状況で正規表現を使用します。例えば、ユーザー ID を Connect:Direct のセキュリティー資格情報と突き合わせたり、正規表現との一致項目があるたびに新しいメッセージを作成することによって 1 つのファイルを複数のメッセージに分割したりするときに、正規表現を使用します。Managed File Transfer が使用する正規表現の構文は、java.util.regex API でサポートされる構文です。この正規表現構文は Perl 言語で使用される正規表現構文と似ていますが、同一ではありません。

ProtocolBridgePropertiesExit2: プロトコル・ファイル・サーバー・プロパティの検索

プロトコル・ファイル・サーバーが多数ある場合は、`com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit2` インターフェースを実装して、転送で参照されているプロトコル・ファイル・サーバー・プロパティを検索できます。`ProtocolBridgeProperties.xml` ファイルを保持するよりはむしろ、このインターフェースを実装することができます。Managed File Transfer には、プロトコル・ファイル・サーバー・プロパティを検索するサンプルのユーザー出口が用意されています。

プロトコル・ブリッジ・プロパティを検索するユーザー出口の構成

このタスクについて

プロトコル・ブリッジ・プロパティを検索するユーザー出口は、インターフェース `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit2` を実装する必要があります。詳細内容は [を参照してください](#)。

他のユーザー出口と同じように、複数のプロトコル・サーバー・プロパティ出口をまとめてチェーニングできます。出口は、エージェント・プロパティ・ファイルで `protocolBridgePropertiesExitClasses` プロパティを使用して指定された順序で呼び出されます。`initialize` メソッドはすべて個別に値を返します。1つ以上のメソッドが値 `false` を返す場合は、エージェントは開始しません。エージェントのイベント・ログにエラーが報告されます。

すべての出口の `getProtocolServerProperties` メソッドについては、1つの全体的な結果のみが返されます。メソッドがプロパティ・オブジェクトを結果コードとして返す場合、この値は返された結果となり、後続の出口の `getProtocolServerProperties` メソッドは呼び出されません。メソッドがヌル値を結果コードとして返す場合は、次の出口の `getProtocolServerProperties` メソッドが呼び出されます。後続の出口がない場合は、ヌルの結果が返されます。全体的な結果コードがヌルである場合は、プロトコル・ブリッジ・エージェントによる検索が失敗したとみなされます。

`ProtocolBridgePropertiesExit2.java` インターフェースの使用が推奨されますが、`ProtocolBridgePropertiesExit.java` インターフェースについては、『[221 ページの『ProtocolBridgePropertiesExit: プロトコル・ファイル・サーバー・プロパティの検索』](#)』を参照してください。

出口を実行するには、以下のステップを実行します。

手順

1. プロトコル・サーバー・プロパティ・ユーザー出口をコンパイルします。
2. コンパイルした出口とそのパッケージ構造が含まれる Java アーカイブ (JAR) ファイルを作成します。
3. 出口クラスが含まれる JAR ファイルをプロトコル・ブリッジ・エージェントの `exits` ディレクトリーに置きます。このディレクトリーは、`MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` ディレクトリーにあります。
4. プロパティ `protocolBridgePropertiesExitClasses` を含めるように、プロトコル・ブリッジ・エージェントのプロパティ・ファイルを編集します。このプロパティの値には、プロトコル・ブリッジ・サーバー・プロパティ・ユーザー出口を実装するクラスのコンマ区切りのリストを指定します。出口クラスは、このリストで指定された順序で呼び出されます。詳細内容は [を参照してください](#)。
5. オプションで、`protocolBridgePropertiesConfiguration` プロパティを指定できます。このプロパティに指定した値は、`protocolBridgePropertiesExitClasses` によって指定された出口クラスの `initialize()` メソッドにストリングとして渡されます。詳細内容は [を参照してください](#)。

サンプル・ユーザー出口の使用

このタスクについて

プロトコル・ブリッジ・プロパティを探索するユーザー出口のサンプルが `MQ_INSTALLATION_PATH/mqft/samples/protocolBridge` ディレクトリーに用意されています (267 ページの『[プロトコル・ブリッジ・プロパティ・ユーザー出口のサンプル](#)』のトピックを参照)。

`SamplePropertiesExit2.java` 出口は、プロトコル・サーバーのプロパティを含むプロパティ・ファイルを読み取ります。プロパティ・ファイル内の各項目の形式は、次のとおりです。

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

プロパティ・ファイルの場所は、プロトコル・ブリッジ・エージェント・プロパティである `protocolBridgePropertiesConfiguration` から取得されます。

サンプル・ユーザー出口を実行するには、以下のステップを実行します。

手順

1. `SamplePropertiesExit2.java` ファイルをコンパイルします。
2. コンパイルした出口とそのパッケージ構造が含まれる JAR ファイルを作成します。
3. JAR ファイルを `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/exits` ディレクトリーに置きます。
4. `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` ファイルを編集して、次の行を含めます。

```
protocolBridgePropertiesExitClasses=SamplePropertiesExit2
```

5. プロトコル・ブリッジ・プロパティのファイル (例えば `protocol_bridge_properties.properties`) を、ディレクトリー `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent` に作成します。このファイルを編集して、次の形式の項目を含めます。

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

6. `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/agent.properties` ファイルを編集して、次の行を含めます。

```
protocolBridgePropertiesConfiguration=MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/protocol_bridge_properties.properties
```

`protocol_bridge_properties.properties` ファイルへの絶対パスを使用する必要があります。

7. `fteStartAgent` コマンドを使用してプロトコル・ブリッジ・エージェントを開始します。

関連概念

[214 ページの『プロトコル・ブリッジ』](#)

プロトコル・ブリッジを使用すれば、Managed File Transfer (MFT) ネットワークから、MFT ネットワークの外部 (ローカル・ドメインとリモート・ロケーションの両方) にあるファイル・サーバーに格納されているファイルにアクセスできます。このファイル・サーバーでは、FTP、FTPS、または SFTP ネットワーク・プロトコルを使用できます。それぞれのファイル・サーバーで少なくとも 1 つの専用エージェントが必要です。この専用エージェントは、プロトコル・ブリッジ・エージェントとして知られています。ブリッジ・エージェントは、複数のファイル・サーバーと相互作用できます。

関連資料

[907 ページの『ProtocolBridgePropertiesExit2.java インターフェース』](#)

267 ページの『[プロトコル・ブリッジ・プロパティ・ユーザー出口のサンプル](#)』

550 ページの『[MFT agent.properties ファイル](#)』

各 Managed File Transfer Agent には、独自のプロパティ・ファイル `agent.properties` があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。`agent.properties` ファイルには、エージェントの動作を変更するプロパティを含めることもできます。

393 ページの『[fteCreateBridgeAgent \(MFT プロトコル・ブリッジ・エージェントの作成および構成\)](#)』

`fteCreateBridgeAgent` コマンドは、Managed File Transfer プロトコル・ブリッジ・エージェントとその関連構成を作成します。プロトコル・ブリッジ・エージェントは、ファイルを送受信するファイル・サーバーごとに作成します。

ProtocolBridgePropertiesExit: プロトコル・ファイル・サーバー・プロパティの検索

プロトコル・ファイル・サーバーが多数ある場合は、`com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` インターフェースを実装して、転送で参照されているプロトコル・ファイル・サーバー・プロパティを検索できます。`ProtocolBridgeProperties.xml` ファイルを保持するよりはむしろ、このインターフェースを実装することができます。`ProtocolBridgePropertiesExit2.java` インターフェースの使用が推奨されますが、`ProtocolBridgePropertiesExit.java` インターフェースもサポートされています。IBM WebSphere MQ File Transfer Edition から `ProtocolBridgePropertiesExit.java` インターフェースを既に実装していた場合は、IBM WebSphere MQ 7.5 以降でそれを使用できます。`ProtocolBridgePropertiesExit2.java` 内の新しい `getCredentialLocation` メソッドは、`ProtocolBridgeCredentials.xml` ファイルのデフォルトの場所 (つまり、ホーム・ディレクトリー) を使用します。

プロトコル・ブリッジ・プロパティを検索するユーザー出口の構成

プロトコル・ブリッジ・プロパティを検索するユーザー出口は、インターフェース `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` を実装する必要があります。詳細内容は [を参照してください](#)。

他のユーザー出口と同じように、複数のプロトコル・サーバー・プロパティ出口をまとめてチェーニングできます。出口は、エージェント・プロパティ・ファイルで `protocolBridgePropertiesExitClasses` プロパティを使用して指定された順序で呼び出されます。`initialize` メソッドはすべて個別に値を返します。1つ以上のメソッドが値 `false` を返す場合は、エージェントは開始しません。エージェントのイベント・ログにエラーが報告されます。

すべての出口の `getProtocolServerProperties` メソッドについては、1つの全体的な結果のみが返されます。メソッドがプロパティ・オブジェクトを結果コードとして返す場合、この値は返された結果となり、後続の出口の `getProtocolServerProperties` メソッドは呼び出されません。メソッドがヌル値を結果コードとして返す場合は、次の出口の `getProtocolServerProperties` メソッドが呼び出されます。後続の出口がない場合は、ヌルの結果が返されます。全体的な結果コードがヌルである場合は、プロトコル・ブリッジ・エージェントによる検索が失敗したとみなされます。

出口を実行するには、以下のステップを実行します。

1. プロトコル・サーバー・プロパティ・ユーザー出口をコンパイルします。
2. コンパイルした出口とそのパッケージ構造が含まれる Java アーカイブ (JAR) ファイルを作成します。
3. 出口クラスが含まれる JAR ファイルをプロトコル・ブリッジ・エージェントの `exits` ディレクトリーに置きます。このディレクトリーは、`MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` ディレクトリーにあります。
4. プロパティ `protocolBridgePropertiesExitClasses` を含めるように、プロトコル・ブリッジ・エージェントのプロパティ・ファイルを編集します。このプロパティの値には、プロトコル・ブリッジ・サーバー・プロパティ・ユーザー出口を実装するクラスのコンマ区切りのリストを指定します。出口クラスは、このリストで指定された順序で呼び出されます。詳細内容は [を参照してください](#)。
5. オプションで、`protocolBridgePropertiesConfiguration` プロパティを指定できます。このプロパティに指定した値は、`protocolBridgePropertiesExitClasses` によって指定された出口クラスの `initialize()` メソッドにストリングとして渡されます。詳細内容は [を参照してください](#)。

ProtocolBridgePropertiesExit.java インターフェース

```
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);
}
}
```

ファイル・サーバーの資格情報のマップ

プロトコル・ブリッジ・エージェントのデフォルトの資格情報マッピング機能を使用するか、独自のユーザー出口を作成して、Managed File Transfer にあるユーザー資格情報をファイル・サーバーのユーザー資格情報にマップします。Managed File Transfer には、ユーザー資格情報マッピングを実行するサンプルのユーザー出口が用意されています。

ProtocolBridgeCredentials.xml ファイルを使用してファイル・サーバーの資格情報をマップする

プロトコル・ブリッジ・エージェントのデフォルトの資格情報マッピング機能を使用して、Managed File Transfer のユーザー資格情報をファイル・サーバーのユーザー資格情報にマップします。Managed File Transfer で提供される XML ファイルを編集して、ユーザーの資格情報を組み込むことができます。

このタスクについて

ProtocolBridgeCredentials.xml ファイルは、ユーザーが手動で作成する必要があります。このファイルのデフォルトの場所は、プロトコル・ブリッジ・エージェントを始動したユーザーのホーム・ディレクトリです。しかし、エージェントがアクセス可能なファイル・システム上の任意の場所に、このファイルを保管することができます。別の場所を指定するには、<credentialsFile> エレメントを ProtocolBridgeProperties.xml ファイルに追加します。例:

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

プロトコル・ブリッジ・エージェントを使用する前に、このファイルを編集してホスト、ユーザー、および資格情報を含めることによって、資格情報マッピングをセットアップします。詳細およびサンプルについては、『574 ページの『プロトコル・ブリッジの資格情報ファイルのフォーマット』』を参照してください。

 z/OS を使用している z/OS プラットフォーム、IBM WebSphere MQ 7.5 またはそれ以前のプラットフォームで ProtocolBridgeCredentials.xml ファイルを作成した場合、ファイルを編集する前に、ファイルタグを設定する必要があります。次のコマンドを実行して、ファイルが ASCII コンテンツを持つものとしてマークを付けます。

```
chtag -t -c IS08859-1 ProtocolBridgeCredentials.xml
```

注:  z/OS では、プロトコル・ブリッジ資格情報ファイルをデータ・セット上に保管することができます。なお、.xml ファイルの名前はユーザーが指定できます。

手順

1. <tns:server name="server name"> の行を編集して、name 属性の値を ProtocolBridgeProperties.xml ファイル内のサーバー名に変更します。

IBM WebSphere MQ File Transfer Edition 7.0.4 用に作成されたプロトコル・ブリッジ・エージェントには、ProtocolBridgeProperties.xml ファイル (または関連するユーザー出口) が含まれていないため、IBM WebSphere MQ File Transfer Edition 7.0.4 Fix Pack 1 以降のサーバー名には自動的にサーバーのホスト名が割り当てられます。したがって、更新された ProtocolBridgeCredentials.xml ファイルを <server> エントリで使用すると、サーバーのホスト名に対応する名前が一致します。

ワイルドカードや正規表現を含むサーバー名を使用したことを指定するには、パターン属性を使用します。例:

```
<tns:server name="serverA*" pattern="wildcard">
```

2. <tns:server> の子エレメントとして、ユーザー ID および資格情報をファイルに挿入します。ファイルには、以下の 1 つ以上のエレメントを挿入できます。

- プロトコル・ファイル・サーバーが FTP、FTPS または SFTP サーバーである場合は、パスワードを使用して、転送を要求しているユーザーを認証できます。次の行をファイルに挿入します。

```
<tns:user name="FTE User ID"  
  serverUserId="Server User ID"  
  serverPassword="Server Password">  
</tns:user>
```

次に、属性の値を変更します。

- name は、MFT 転送要求に関連付けられた MQMD ユーザー ID と一致する Java 正規表現です。
- serverUserId は、プロトコル・ファイル・サーバーにログイン・ユーザー ID として渡される値です。serverUserId 属性が指定されていない場合は、MFT 転送要求に関連付けられた MQMD ユーザー ID が代わりに使用されます。
- serverPassword は、serverUserId に関連付けられたパスワードです。

name 属性には Java 正規表現を含めることができます。資格情報マッパーは、MFT 転送要求の MQMD ユーザー ID を、この正規表現と突き合わせようとします。プロトコル・ブリッジ・エージェントは、エレメントがファイルに存在している順序で、<tns:user> エレメントの name 属性内の正規表現と MQMD ユーザー ID を突き合わせようとします。一致が検出されると、プロトコル・ブリッジ・エージェントはその他の一致を検索しません。一致が検出されると、対応する serverUserId 値と serverPassword 値が、ログイン・ユーザー ID とパスワードとしてプロトコル・ファイル・サーバーに渡されます。MQMD ユーザー ID の突き合わせでは大/小文字が区別されます。

- プロトコル・ファイル・サーバーが SFTP サーバーである場合は、転送を要求しているユーザーの認証に公開鍵と秘密鍵を使用できます。次の行をファイルに挿入して、属性の値を変更します。<tns:user> エレメントには、1 つ以上の <tns:privateKey> エレメントを含めることができます。

```
<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 は、MFT 転送要求に関連付けられた MQMD ユーザー ID と一致する Java 正規表現です。
- serverUserId は、プロトコル・ファイル・サーバーにログイン・ユーザー ID として渡される値です。serverUserId 属性が指定されていない場合は、MFT 転送要求に関連付けられた MQMD ユーザー ID が代わりに使用されます。
- hostKey は、ログオン時にサーバーから返されることが予期される鍵です。
- key は、serverUserId の秘密鍵です。
- keyPassword は、公開鍵を生成するための鍵のパスワードです。
- associationName は、トレースとロギングの目的で識別するために使用される値です。

name 属性には Java 正規表現を含めることができます。資格情報マッパーは、MFT 転送要求の MQMD ユーザー ID を、この正規表現と突き合わせようとします。プロトコル・ブリッジ・エージェントは、エレメントがファイルに存在している順序で、<tns:user> エレメントの name 属性内の正規表現と MQMD ユーザー ID を突き合わせようとします。一致が検出されると、プロトコル・ブリッジ・エージェントはその他の一致を検索しません。一致が検出されると、対応する serverUserId 値と key 値が、プロトコル・ファイル・サーバーで MFT ユーザーを認証するために使用されます。MQMD ユーザー ID の突き合わせでは大/小文字が区別されます。

プロトコル・ブリッジ・エージェントでの秘密鍵の使用に関する詳細は、[227 ページの『例: UNIX SFTP サーバーで秘密鍵の資格情報を使用するようにプロトコル・ブリッジ・エージェントを構成する方法』](#)を参照してください。

注:

転送要求がコマンド・キューに書き込まれると、ソース・エージェントのコマンド・キューが z/OS システムまたは IBM i システム上にある場合、MQMD ユーザー ID は大文字に変換されることがあります。この結果、同じ発信元ユーザーの MQMD ユーザー ID であっても、転送要求で指定されたソース・エージェントに応じて、元の大/小文字の形式かまたは変換された大文字の形式で資格情報出口に着信することになります。デフォルトの資格情報マッピング出口は、提供された MQMD ユーザー ID に照らして大/小文字を区別する突き合わせを実行します。このことは、マッピング・ファイルにおいて考慮する必要があります。

出口クラスを使用したファイル・サーバーの資格情報のマップ

プロトコル・ブリッジ・エージェントのデフォルトの資格情報マッピング機能を使用しない場合は、独自のユーザー出口を作成して、Managed File Transfer のユーザー資格情報をファイル・サーバーのユーザー資格情報にマップできます。Managed File Transfer には、ユーザー資格情報マッピングを実行するサンプルのユーザー出口が用意されています。資格情報マッピング・ユーザー出口を構成すると、デフォルトの資格情報マッピング機能の代わりになります。

プロトコル・ブリッジ資格情報ユーザー出口の構成

このタスクについて

マッピング・プロトコル・ブリッジ資格情報のユーザー出口は、以下のいずれかのインターフェースを実装する必要があります。

- `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit`。プロトコル・ブリッジ・エージェントが1つのデフォルト・プロトコル・ファイル・サーバーとの間でファイルを転送できるようにします。
- `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit2`。複数のエンドポイントとの間でファイルを転送できるようにします。

`com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit2` インターフェースには、`com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit` と同じ機能が含まれ、加えて拡張機能が含まれています。詳細については、905 ページの『[ProtocolBridgeCredentialExit.java インターフェース](#)』および 907 ページの『[ProtocolBridgeCredentialExit2.java インターフェース](#)』を参照してください。

資格情報出口は、他のユーザー出口と同じような方法でまとめてチェーニングできます。出口は、エージェント・プロパティ・ファイルで `protocolBridgeCredentialConfiguration` プロパティを使用して指定された順序で呼び出されます。initialize メソッドはすべて個別に値を返します。1つ以上のメソッドが値 `false` を返す場合は、エージェントは開始しません。エージェントのイベント・ログにエラーが報告されます。

すべての出口の `mapMQUserId` メソッドについては、1つの全体的な結果のみが、以下のように返されます。

- メソッドが値 `USER_SUCCESSFULLY_MAPPED` または `USER_DENIED_ACCESS` を結果コードとして返す場合、この値は返された結果となり、後続の出口の `mapMQUserId` メソッドは呼び出されません。
- メソッドが値 `NO_MAPPING_FOUND` を結果コードとして返す場合は、次の出口の `mqMQUserId` メソッドが呼び出されます。
- 後続の出口がない場合は、結果 `NO_MAPPING_FOUND` が返されます。
- `USER_DENIED_ACCESS` または `NO_MAPPING_FOUND` の全体的な結果コードは、ブリッジ・エージェントによる転送障害であるとみなされます。

出口を実行するには、以下のステップを実行します。

手順

1. プロトコル・ブリッジ資格情報ユーザー出口をコンパイルします。
2. コンパイルした出口とそのパッケージ構造が含まれる Java アーカイブ (JAR) ファイルを作成します。
3. 出口クラスが含まれる JAR ファイルをブリッジ・エージェントの `exits` ディレクトリーに置きます。このディレクトリーは、`MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` ディレクトリーにあります。
4. プロパティ `protocolBridgeCredentialExitClasses` を含めるように、プロトコル・ブリッジ・エージェントのプロパティ・ファイルを編集します。このプロパティの値には、プロトコル・ブリッジ資格情報の出口ルーチンを実装するクラスのコンマ区切りのリストを指定します。出口クラスは、このリストで指定された順序で呼び出されます。詳細内容は [を参照してください](#)。
5. プロトコル・ブリッジ・エージェントのプロパティ・ファイルを編集して、以下を含めます。

```
exitClassPath=IBM MQ
installation_directory\mqft\config\configuration_queue_manager\agents\protocol_bridge_agent_name\exits\SampleCredentialExit.jar
```

エージェントの `agent.properties` ファイルは、`MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/bridge_agent_name` ディレクトリーにあります。

`agent.properties` ファイルを変更した場合は、エージェントを再始動してその変更が反映されるようにする必要があります。

- オプションで、`protocolBridgeCredentialConfiguration` プロパティを指定できます。このプロパティに指定した値は、`protocolBridgeCredentialExitClasses` によって指定された出口クラスの `initialize()` メソッドにストリング・オブジェクトとして渡されます。詳細については [550 ページの『MFT agent.properties ファイル』](#) を参照してください。
- fteStartAgent** コマンドを使用してプロトコル・ブリッジ・エージェントを開始します。

サンプル・ユーザー出口の使用

このタスクについて

プロトコル・ブリッジ資格情報出口のサンプルが `MQ_INSTALLATION_PATH/mqft/samples/protocolBridge` ディレクトリーと [『265 ページの『プロトコル・ブリッジ資格情報ユーザー出口のサンプル』』](#) のトピックに用意されています。このサンプルは、`com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit` インターフェースに基づいています。

`SampleCredentialExit.java` 出口は、転送要求に関連付けられた MQMD ユーザー ID をサーバー・ユーザー ID とサーバー・パスワードにマップするプロパティ・ファイルを読み取ります。プロパティ・ファイルの場所は、プロトコル・ブリッジ・エージェント・プロパティである `protocolBridgeCredentialConfiguration` から取得されます。

サンプル・ユーザー出口を実行するには、以下のステップを実行します。

手順

- `SampleCredentialExit.java` ファイルをコンパイルします。
- コンパイルした出口とそのパッケージ構造が含まれる JAR ファイルを作成します。
- JAR ファイルを `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/exits` ディレクトリーに置きます。
- `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` ファイルを編集して、次の行を含めます。

```
protocolBridgeCredentialExitClasses=SampleCredentialExit
```

- プロトコル・ブリッジ・エージェントのプロパティ・ファイルを編集して、以下を含めます。

```
exitClassPath=IBM MQ
installation_directory\mqft\config\configuration_queue_manager\agents\protocol_bridge_agent_name\exits\SampleCredentialExit.jar
```

エージェントの `agent.properties` ファイルは、`MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` ディレクトリーにあります。

`agent.properties` ファイルを変更した場合は、エージェントを再始動してその変更が反映されるようにする必要があります。

- 資格情報プロパティ・ファイル (credentials.properties) を `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent` ディレクトリーに作成し、次の形式で項目が含まれるように編集します。

```
mqUserId=serverUserId,serverPassword
```

- `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` ファイルを編集して、次の行を含めます。

```
protocolBridgeCredentialConfiguration=MQ_DATA_PATH/mqft/
config/coordination_queue_manager/agents/bridge_agent_name/credentials.properties
```

credentials.properties ファイルへの絶対パスを使用する必要があります。

- fteStartAgent** コマンドを使用してプロトコル・ブリッジ・エージェントを開始します。

例: UNIX SFTP サーバーで秘密鍵の資格情報を使用するようにプロトコル・ブリッジ・エージェントを構成する方法

この例では、ProtocolBridgeCredentials.xml ファイルを生成して構成する方法を例示します。この例は標準的な例であり、ご使用のプラットフォームに応じて詳細が異なることがありますが、原則は同じです。

このタスクについて

手順

- SFTP サーバーでの認証に使用する公開鍵と秘密鍵を生成します。

例えば、Linux ホスト・システムでは、'openssh' パッケージの一部として提供されているツール **ssh-keygen** を使用して、公開鍵と秘密鍵のペアを作成できます。

デフォルトでは、引数を指定せずに **ssh-keygen** コマンドを実行すると、2つの鍵ファイルの場所とパスフレーズの入力を求めるプロンプトが出されます。デフォルトの名前は以下のとおりです。

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



重要: 最新バージョンの OpenSSH (RHEL 8 で提供されているものなど) から **ssh-keygen** コマンドを使用している場合、使用される鍵フォーマットはプロトコル・ブリッジ・エージェントと互換性がなく、SFTP サーバーへの転送は失敗し、以下のメッセージが表示されます。

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

これらの新しいバージョンの OpenSSH と互換性のある秘密鍵を作成するには、**ssh-keygen** コマンドに以下の引数を使用して鍵フォーマットを指定します。

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

id_rsa 秘密鍵の内容には、以下の最初と最後の行が含まれます。

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

これは、プロトコル・ブリッジ・エージェントと互換性があります。

- id_rsa.pub ファイルの内容全体を SFTP サーバー上の SFTP ユーザーの `~/.ssh/authorized_keys` ファイルにコピーします。

SFTP サーバーが鍵認証を許可するように、このファイルおよび ~/.ssh ディレクトリーのファイル許可が適切に設定されていることを確認してください。通常、これらの権限は以下のとおりです

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

3. Managed File Transfer では、MD5 アルゴリズムを使用して生成されたホストの ssh 指紋が必要です。以下のいずれかのコマンドを実行して、SFTP サーバーのホストの ssh 指紋を取得します。

- Red Hat® Enterprise Linux バージョン 6.x 以前、および Linux Ubuntu 14.04 の場合、以下のコマンドを実行します。

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

- Red Hat Enterprise Linux バージョン 7.x、Linux Ubuntu 16.04、および SuSE Linux 12.4 以降では、ssh-keygen コマンドにより、SHA56 アルゴリズムを使用して ssh 指紋がデフォルトで生成されます。MD5 アルゴリズムを使用して ssh 指紋を生成するには、以下のコマンドを実行します。

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

コマンドの出力は、以下の例のようになります。

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

ProtocolBridgeCredentials.xml ファイル内でホスト・キーとして使用する出力の 16 進数部分のみを抽出します (ステップ 228 ページの『4』を参照)。したがって、この例では、64:39:f5:49:41:10:55:d2:0b:81:42:5c:87:62:9d:27 を抽出します。

4. プロトコル・ブリッジ・エージェント・システムで、ProtocolBridgeCredentials.xml ファイルを編集します。以下の例でイタリックで示されている値を独自の値に置換します。

```
<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>
```

ここで、

- Agent_name は、プロトコル・ブリッジ・エージェントの名前です。
- SFTP_host_name は、ProtocolBridgeProperties.xml ファイルに示されている SFTP サーバーの名前です。
- mq_User_ID は、転送要求に関連付けられた MQMD ユーザー ID です。
- SFTP_user_ID は、ステップ 2 で使用されている SFTP ユーザー ID です。これは、ログイン・ユーザー ID として SFTP 機能に渡される値です。
- ssh_host_finger は、ステップ 3 で収集した指紋です。
- name は、トレースとロギングを目的として使用するために指定できる名前です。
- pass_phrase は、ステップ 1 で ssh-keygen に指定したパスワードです。

- `id_rsa` ファイルの内容を完了します。は、ステップ 1 から生成された `id_rsa` ファイルの完全な内容です。接続エラーを回避するには、以下の両方の項目が含まれていることを確認してください。

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

<tns:privatekey> エレメントを複製することによって、鍵を追加できます。

5. プロトコル・ブリッジ・エージェントがまだ開始していない場合は、開始します。または、プロトコル・ブリッジ・エージェントが `ProtocolBridgeCredentials.xml` ファイルを定期的にポーリングして、変更を取得します。

FTPS サーバー用のプロトコル・ブリッジの構成

FTPS サーバーの構成は、FTP サーバーの構成と同様の方法で行います。つまり、サーバー用のブリッジ・エージェントを作成し、サーバー・プロパティを定義し、ユーザー資格情報をマップします。

このタスクについて

FTPS サーバーを構成するには、以下のステップを実行します。

手順

1. **fteCreateBridgeAgent** コマンドを使用して、FTPS サーバー用のプロトコル・ブリッジ・エージェントを作成します。FTP に適用できるパラメーターを FTPS にも適用できますが、それに加えて以下の 3 つの FTPS に固有な必須パラメーターがあります。

- a) **-bt** パラメーター。このパラメーターの値として FTPS を指定してください。
- b) トラストストア・ファイルの **-bts** パラメーター。コマンドはサーバー認証のみが必要だと想定するので、トラストストア・ファイルの場所を指定しなければなりません。

デフォルトで **fteCreateBridgeAgent** コマンドによって FTPS プロトコルの明示書式が構成されますが、暗黙書式はプロトコル・ブリッジ・プロパティ・ファイルを変更することによって構成できます。常にプロトコル・ブリッジはパッシブ・モードで FTPS サーバーに接続します。

fteCreateBridgeAgent コマンドについて詳しくは、[393 ページの『fteCreateBridgeAgent \(MFT プロトコル・ブリッジ・エージェントの作成および構成\)』](#)を参照してください。

トラストストア・ファイルの作成方法に関する説明が必要な場合は、IBM Developer 記事の [Configuring Secure Sockets Layer connectivity in IBM WebSphere MQ File Transfer Edition](#) を参照するか、Oracle 社の [keytool](#) の資料で keytool に関する情報を参照してください。

2. プロトコル・ブリッジ・プロパティ・ファイル `ProtocolBridgeProperties.xml` 内の <ftpsServer> エレメント内に FTPS サーバー・プロパティを定義します。詳しくは、[215 ページの『ProtocolBridgeProperties.xml ファイルを使用したプロトコル・ファイル・サーバーのプロパティの定義』](#)を参照してください。プロトコル・ブリッジ・プロパティ・ファイルを編集して、クライアント認証を使用可能にすることもできます。すべての構成オプションについては、[577 ページの『プロトコル・ブリッジ・プロパティ・ファイルのフォーマット』](#)を参照してください。
3. プロトコル・ブリッジ・エージェントのデフォルトの資格情報マッピング機能を使用するか、独自のユーザー出口を作成して、Managed File Transfer にあるユーザー資格情報を FTPS サーバーのユーザー資格情報にマップします。詳しくは、[222 ページの『ファイル・サーバーの資格情報のマップ』](#)を参照してください。
4. デフォルトでは、トラストストア・ファイルは JKS 形式になるように構成されます。この形式を変更する場合は、プロトコル・ブリッジ・プロパティ・ファイルを編集します。

例

プロトコル・ブリッジ・プロパティ・ファイル内の FTPS サーバーの項目の例を以下に示します。

```

<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="ftpserver.mycompany.com"/>

  <tns:ftpServer name="ftpserver.mycompany.com" host="ftpserver.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>

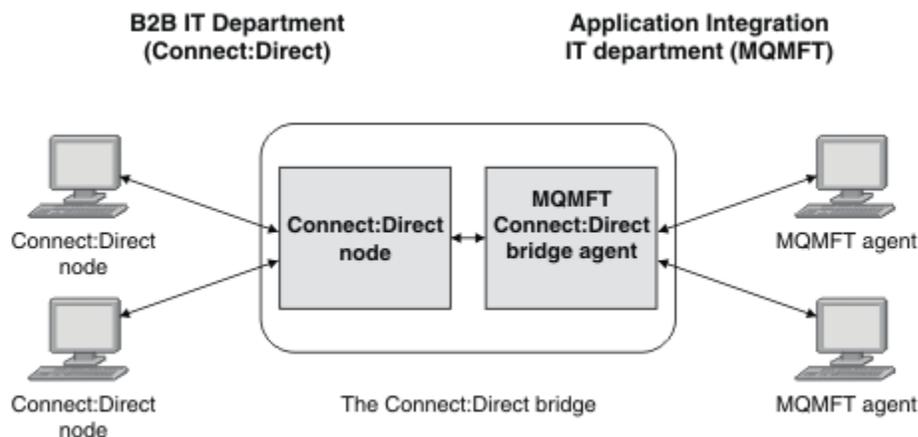
```

次のタスク

FTPS プロトコルのサポートされている部分と、サポートされていない部分に関する情報は、[665 ページの『プロトコル・ブリッジによる FTPS サーバーのサポート』](#)を参照してください。

Connect:Direct ブリッジ

既存の IBM Sterling Connect:Direct ネットワークとの相互間で、ファイルを転送することができます。Managed File Transfer のコンポーネントである Connect:Direct ブリッジを使用して、MFT と IBM Sterling Connect:Direct の間でファイルを転送します。



この図は、2つの部門、B2B IT 部門、およびアプリケーション統合 IT 部門の間の MFT Connect:Direct ブリッジを示しています。B2B IT 部門は、Connect:Direct を使用して、会社のビジネス・パートナーとの間でファイルを転送します。アプリケーション統合 IT 部門では、IBM MQ をメッセージング・インフラストラクチャーとして使用するため、最近ファイル転送ソリューションとして Managed File Transfer を選択しました。

MFT Connect:Direct ブリッジを使用することにより、B2B IT 部門の Connect:Direct ネットワークと、アプリケーション統合 IT 部門の MFT ネットワークの間で相互にファイルを転送できるようになりました。Connect:Direct ブリッジは、Managed File Transfer のコンポーネントであり、Connect:Direct ノードと通信する MFT エージェントが含まれています。MFT エージェントは Connect:Direct ノードとの転送用の専用エージェントで、Connect:Direct ブリッジ・エージェントと呼ばれます。

Connect:Direct ブリッジは Managed File Transfer の Service コンポーネントおよび Agent コンポーネントの一部として入手可能で、以下のタスクで使用できます。

1. Managed File Transfer コマンドを使用して、MFT エージェントから Connect:Direct ノードへの単一ファイルまたは複数ファイルの転送を開始します。
2. Managed File Transfer コマンドを使用して、Connect:Direct ノードから MFT エージェントへの単一ファイルまたは複数ファイルの転送を開始します。

3. Managed File Transfer コマンドを使用して、ユーザー定義 Connect:Direct プロセスを開始するファイル転送を開始します。

4. Connect:Direct プロセスを使用して、MFT のファイル転送要求を送信します。

Connect:Direct ブリッジでは、Connect:Direct ノードを転送元または転送先とするファイルの転送のみが可能です。Connect:Direct ブリッジは、Connect:Direct プロセスによって実行依頼された転送の一部としてのみ、ローカル・ファイル・システムとの間でファイルを転送できます。

z/OS Connect:Direct ブリッジを使用して、z/OS システム上の Connect:Direct ノード上にあるデータ・セットとの間で転送することができます。Managed File Transfer エージェントだけが関わっているデータ・セット転送と比較すると、動作にいくらかの違いがあります。詳細については、[634 ページの『Connect:Direct ノードとの間のデータ・セット転送』](#)を参照してください。

サポートされているオペレーティング・システム

Connect:Direct ブリッジは MFT Connect:Direct ブリッジ・エージェントと Connect:Direct ノードで構成されます。エージェントは、Windows と Linux for System x でサポートされます。ノードは、IBM Sterling Connect:Direct for Windows および IBM Sterling Connect:Direct for UNIX でサポートされるプラットフォームでサポートされます。Connect:Direct ブリッジ・エージェントを作成し、そのエージェントと通信できるように Connect:Direct ノードを構成する方法については、[143 ページの『Connect:Direct ブリッジの構成』](#)を参照してください。

Connect:Direct ブリッジは、Connect:Direct (Windows の場合) または Connect:Direct (UNIX

z/OS の場合)、あるいは Connect:Direct (z/OS Service インストールの場合) の一部として実行されている Connect:Direct ノードとの間でファイルを転送できます。サポートされている Connect:Direct のバージョンの詳細については、Web ページ『[IBM MQ System Requirements](#)』を参照してください。

Connect:Direct ブリッジを構成するエージェントとノードは、同じシステムに存在しているか、共用 NFS マウントなどによって同じファイル・システムにアクセスできる状態になっている必要があります。このファイル・システムは、Connect:Direct ブリッジがかかわっているファイル転送中に、**cdTmpDir** パラメーターで定義されているディレクトリーに一時的にファイルを保管するために使用されます。

Connect:Direct ブリッジ・エージェントと Connect:Direct ブリッジ・ノードでは、同じパス名を使用してこのディレクトリーを指定する必要があります。例えば、エージェントとノードが別個の Windows システムにある場合、共有ファイル・システムをマウントするためにそれらのシステムで同じドライブ名が使用されている必要があります。以下の構成を使用すると、エージェントとノードで同じパス名を使用できます。

- エージェントとノードが、Windows または Linux for System x のいずれかを実行する同じシステム上にある
- エージェントが Linux for System x 上にあり、ノードが UNIX 上にある
- エージェントが Windows システム上にあり、ノードがそれとは別の Windows システム上にある

以下の構成を使用すると、エージェントとノードで同じパス名を使用できません。

- エージェントが Linux for System x 上にあり、ノードが Windows 上にある
- エージェントが Windows 上にあり、ノードが UNIX 上にある

Connect:Direct ブリッジのインストールを計画する際には、これらの制約事項を考慮してください。

Connect:Direct ノードへのファイルの転送

Connect:Direct ブリッジを使用して、Managed File Transfer エージェントから Connect:Direct ノードへファイルを転送できます。Connect:Direct ブリッジ・エージェントを宛先エージェントとして指定し、宛先ファイルを `connect_direct_node_name:file_path` の形式で指定することにより、Connect:Direct ノードを転送の宛先として指定します。

始める前に

ファイルを転送する前に、Managed File Transfer のコンポーネントである Connect:Direct ブリッジを構成しておく必要があります。詳細については [143 ページの『Connect:Direct ブリッジの構成』](#) を参照してください。

このタスクについて

この例では、Connect:Direct ブリッジ・エージェントは CD_BRIDGE という名前です。ソース・エージェントは、FTE_AGENT という名前であり、WMQFTE のどのバージョンでもかまいません。宛先 Connect:Direct ノードは CD_NODE1 という名前です。転送するファイルは、FTE_AGENT があるシステム上のファイル・パス /home/helen/file.log にあります。ファイルは、CD_NODE1 が実行されているシステム上のファイル・パス /files/data.log に転送されます。

手順

1. **-df** (宛先ファイル) パラメーターに `connect_direct_node_name:file_path` 形式の値を使用し、**-da** (宛先エージェント) パラメーターの値として Connect:Direct ブリッジ・エージェントの名前を指定して、`fteCreateTransfer` コマンドを使用します。

注: `connect_direct_node_name` で指定される Connect:Direct ノードは、ファイルの転送先のノードであり、Connect:Direct ブリッジの一部として機能する Connect:Direct ノードではありません。

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
                 -df CD_NODE1:/files/data.log /home/helen/file.log
```

詳細については [437 ページの『fteCreateTransfer: 新規ファイル転送の開始』](#) を参照してください。

2. ソース・エージェント FTE_AGENT がファイルを Connect:Direct ブリッジ・エージェント CD_BRIDGE に転送します。ファイルは、Connect:Direct ブリッジ・エージェントが実行されているシステム上の、`cdTmpDir` エージェント・プロパティーで定義されている場所に一時的に保管されます。Connect:Direct ブリッジ・エージェントが Connect:Direct ノード CD_NODE1 にファイルを転送します。

関連概念

[230 ページの『Connect:Direct ブリッジ』](#)

既存の IBM Sterling Connect:Direct ネットワークとの相互間で、ファイルを転送することができます。Managed File Transfer のコンポーネントである Connect:Direct ブリッジを使用して、MFT と IBM Sterling Connect:Direct の間でファイルを転送します。

関連タスク

[232 ページの『Connect:Direct ノードからのファイルの転送』](#)

Connect:Direct ブリッジを使用して、Connect:Direct ノードから Managed File Transfer Agent へファイルを転送できます。Connect:Direct ブリッジ・エージェントをソース・エージェントとして指定し、ソース指定を `connect_direct_node_name:file_path` の形式で指定することにより、Connect:Direct ノードを転送のソースとして指定できます。

関連資料

[550 ページの『MFT agent.properties ファイル』](#)

各 Managed File Transfer Agent には、独自のプロパティー・ファイル `agent.properties` があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。agent.properties ファイルには、エージェントの動作を変更するプロパティーを含めることもできます。

Connect:Direct ノードからのファイルの転送

Connect:Direct ブリッジを使用して、Connect:Direct ノードから Managed File Transfer Agent へファイルを転送できます。Connect:Direct ブリッジ・エージェントをソース・エージェントとして指定し、ソース指定を `connect_direct_node_name:file_path` の形式で指定することにより、Connect:Direct ノードを転送のソースとして指定できます。

始める前に

ファイルを転送する前に、Managed File Transfer のコンポーネントである Connect:Direct ブリッジを構成しておく必要があります。 [143 ページの『Connect:Direct ブリッジの構成』](#)を参照してください。

このタスクについて

この例では、Connect:Direct ブリッジ・エージェントは CD_BRIDGE という名前です。宛先エージェントは、FTE_AGENT という名前であり、Managed File Transfer のどのバージョンでもかまいません。ソース Connect:Direct ノードは CD_NODE1 という名前です。転送するファイルは、CD_NODE1 があるシステム上のファイル・パス /home/brian/in.file にあります。ファイルは、FTE_AGENT が実行されているシステム上のファイル・パス /files/out.file に転送されます。

手順

ソース指定の値として `connect_direct_node_name:file_path` の形式を使用し、**-sa** パラメーターの値として Connect:Direct ブリッジ・エージェントの名前を指定して、**fteCreateTransfer** コマンドを使用します。

注: `connect_direct_node_name` で指定される Connect:Direct ノードは、ファイルの転送元のノードであり、Connect:Direct ブリッジの一部として機能する Connect:Direct ノードではありません。以下に例を示します。

```
fteCreateTransfer -sa CD_BRIDGE -da FTE_AGENT
                  -df /files/out.file CD_NODE1:/home/brian/in.file
```

詳しくは、[437 ページの『fteCreateTransfer: 新規ファイル転送の開始』](#)を参照してください。

タスクの結果

Connect:Direct ブリッジ・エージェント CD_BRIDGE が、Connect:Direct ノード CD_NODE1 からのファイルを要求します。Connect:Direct ノードが Connect:Direct ブリッジにファイルを送信します。Connect:Direct ノードからのファイルの転送中、Connect:Direct ブリッジは、cdTmpDir エージェント・プロパティーで定義されている場所に一時的にそのファイルを保管します。Connect:Direct ノードから Connect:Direct ブリッジへのファイル転送が完了すると、Connect:Direct ブリッジは、そのファイルを宛先エージェント FTE_AGENT に送信し、一時ロケーションからそのファイルを削除します。

関連概念

[230 ページの『Connect:Direct ブリッジ』](#)

既存の IBM Sterling Connect:Direct ネットワークとの相互間で、ファイルを転送することができます。Managed File Transfer のコンポーネントである Connect:Direct ブリッジを使用して、MFT と IBM Sterling Connect:Direct の間でファイルを転送します。

関連資料

[550 ページの『MFT agent.properties ファイル』](#)

各 Managed File Transfer Agent には、独自のプロパティー・ファイル `agent.properties` があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。agent.properties ファイルには、エージェントの動作を変更するプロパティーを含めることもできます。

z/OS 上の Connect:Direct ノードへのデータ・セットの転送

Windows または Linux のシステムにある Connect:Direct ブリッジを使用して、z/OS 上の Managed File Transfer エージェントから z/OS 上の Connect:Direct ノードにデータ・セットを転送できます。

始める前に

ファイルを転送する前に、Managed File Transfer のコンポーネントである Connect:Direct ブリッジを構成しておく必要があります。 [143 ページの『Connect:Direct ブリッジの構成』](#)を参照してください。

このタスクについて

この例では、**-df** パラメーターを使用して、転送の宛先を指定します。**-df** パラメーターは、転送のソース・エージェントが Managed File Transfer のどのバージョンの場合でも有効です。ソース・エージェントが IBM WebSphere MQ File Transfer Edition 7.0.4 以降の場合は、**-ds** パラメーターを使用することもできます。ソース・エージェントは、FTE_ZOS1 という名前であり、IBM WebSphere MQ File Transfer Edition 7.0.3 のエージェントです。Connect:Direct ブリッジ・エージェントは、CD_BRIDGE という名前であり、Linux システムにあります。宛先 Connect:Direct ノードは CD_ZOS2 という名前です。ソース・エージェントも宛先 Connect:Direct ノードも、z/OS システムにあります。転送するデータ・セットは、FTE_ZOS1 が配置されているシステムの //FTEUSER.SOURCE.LIB にあります。そのデータ・セットを、CD_ZOS2 が配置されているシステムのデータ・セット //CDUSER.DEST.LIB に転送します。

手順

1. **-df** パラメーターの値を `connect_direct_node_name:data_set_name;attributes` という形式で指定し、**-da** (宛先エージェント) パラメーターの値を Connect:Direct ブリッジ・エージェントの名前として指定して、`fteCreateTransfer` コマンドを使用します。

`connect_direct_node_name` で指定される Connect:Direct ノードは、データ・セットの転送先のノードであり、Connect:Direct ブリッジの一部として機能する Connect:Direct ノードではありません。

`data_set_name` で指定するデータ・セット名は、相対名ではなく絶対名でなければなりません。Connect:Direct では、データ・セット名の接頭部としてユーザー名が追加されません。

```
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'
```

詳細については 437 ページの『[fteCreateTransfer: 新規ファイル転送の開始](#)』を参照してください。

2. ソース・エージェント FTE_ZOS1 がデータ・セット内のデータを Connect:Direct ブリッジ・エージェント CD_BRIDGE に転送します。そのデータは、Connect:Direct ブリッジ・エージェントが稼働しているシステムでフラット・ファイルとして一時的に格納されます。格納場所は、`cdTmpDir` エージェント・プロパティーで定義されている場所になります。Connect:Direct ブリッジ・エージェントが Connect:Direct ノード CD_ZOS2 にデータを転送します。転送が完了すると、Connect:Direct ブリッジ・エージェントが稼働しているシステムからそのフラット・ファイルが削除されます。

関連概念

230 ページの『[Connect:Direct ブリッジ](#)』

既存の IBM Sterling Connect:Direct ネットワークとの相互間で、ファイルを転送することができます。Managed File Transfer のコンポーネントである Connect:Direct ブリッジを使用して、MFT と IBM Sterling Connect:Direct の間でファイルを転送します。

関連資料

634 ページの『[Connect:Direct ノードとの間のデータ・セット転送](#)』

Connect:Direct ブリッジを使用して、Managed File Transfer エージェントと IBM Sterling Connect:Direct ノードの間でデータ・セット転送を実行できます。転送のソースと宛先のいずれかまたは両方としてデータ・セットを指定できます。

637 ページの『[Connect:Direct プロセス・ステートメント・パラメーターと BPXWDYN キーのマッピング](#)』

ソースまたは宛先のいずれかが Connect:Direct ノードになっている状況でデータ・セットの転送要求を送信すると、指定したサポート対象の BPXWDYN キーが、Connect:Direct プロセスによって受け入れられる形式に変換されます。

Connect:Direct ノードへの複数ファイルの転送

Connect:Direct ブリッジを使用して、Managed File Transfer Agent から Connect:Direct ノードへ複数のファイルを転送できます。複数ファイル転送の宛先として Connect:Direct ノードを使用するには、Connect:Direct ブリッジ・エージェントを宛先エージェントとして指定し、`connect_direct_node_name:directory_path` の形式で宛先ディレクトリーを指定します。

始める前に

ファイルを転送する前に、Managed File Transfer のコンポーネントである Connect:Direct ブリッジを構成しておく必要があります。 [143 ページの『Connect:Direct ブリッジの構成』](#)を参照してください。

このタスクについて

この例では、ソース・エージェントは FTE_AGENT という名前です。 Connect:Direct ブリッジ・エージェントは CD_BRIDGE という名前です。宛先 Connect:Direct ノードは CD_NODE1 という名前です。転送するファイルは、FTE_AGENT があるシステム上の /home/jack/data.log、/logs/log1.txt および /results/latest です。これらのファイルが、CD_NODE1 が実行されているシステム上の /in/files ディレクトリーに転送されます。

手順

-dd (宛先ディレクトリー) パラメーターに `connect_direct_node_name:directory_path` 形式の値を使用して、`fteCreateTransfer` コマンドを使用します。 **-da** (宛先エージェント) パラメーターの値を、Connect:Direct ブリッジ・エージェントの名前として指定します。

注: `connect_direct_node_name` で指定される Connect:Direct ノードは、ファイルの転送先のノードであり、Connect:Direct ブリッジの一部として機能する Connect:Direct ノードではありません。

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
                  -dd CD_NODE1:/in/files /home/jack/data.log
                  /logs/log1.txt /results/latest
```

詳細については [437 ページの『fteCreateTransfer: 新規ファイル転送の開始』](#) を参照してください。

タスクの結果

ソース・エージェント FTE_AGENT が最初のファイルを Connect:Direct ブリッジ・エージェント CD_BRIDGE に転送します。 Connect:Direct ブリッジ・エージェントがこのファイルを、`cdTmpDir` プロパティーで定義された場所に一時的に保管します。ソース・エージェントから Connect:Direct ブリッジにファイルが完全に転送されると、Connect:Direct ブリッジ・エージェントがそのファイルを、`cdNode` エージェント・プロパティーで定義された Connect:Direct ノードに送信します。このノードがファイルを、宛先 Connect:Direct ノード CD_NODE1 に送信します。2つの Connect:Direct ノード間で転送が完了すると、Connect:Direct ブリッジ・エージェントが一時ロケーションからファイルを削除します。指定されたすべてのソース・ファイルごとに、このプロセスが繰り返されます。

関連概念

[230 ページの『Connect:Direct ブリッジ』](#)

既存の IBM Sterling Connect:Direct ネットワークとの相互間で、ファイルを転送することができます。Managed File Transfer のコンポーネントである Connect:Direct ブリッジを使用して、MFT と IBM Sterling Connect:Direct の間でファイルを転送します。

関連タスク

[231 ページの『Connect:Direct ノードへのファイルの転送』](#)

Connect:Direct ブリッジを使用して、Managed File Transfer エージェントから Connect:Direct ノードへファイルを転送できます。Connect:Direct ブリッジ・エージェントを宛先エージェントとして指定し、宛先ファイルを `connect_direct_node_name:file_path` の形式で指定することにより、Connect:Direct ノードを転送の宛先として指定します。

[237 ページの『ワイルドカードを使用した Connect:Direct への複数ファイルの転送』](#)

Managed File Transfer エージェントから Connect:Direct ノードへ複数のファイルを転送するには、Connect:Direct ブリッジを使用します。 **fteCreateTransfer** コマンドに指定するソース指定内では、ワイルドカード文字を使用できます。ワイルドカードを使用するすべての Managed File Transfer の転送と同様、ファイル・パスの最後の部分にのみワイルドカード文字を使用できます。例えば、`/abc/def*` は有効なファイル・パスですが、`/abc*/def` は無効です。

[232 ページの『Connect:Direct ノードからのファイルの転送』](#)

Connect:Direct ブリッジを使用して、Connect:Direct ノードから Managed File Transfer Agent へファイルを転送できます。Connect:Direct ブリッジ・エージェントをソース・エージェントとして指定し、ソース

指定を `connect_direct_node_name:file_path` の形式で指定することにより、Connect:Direct ノードを転送のソースとして指定できます。

236 ページの『Connect:Direct ノードからの複数ファイルの転送』

Connect:Direct ブリッジを使用して、Connect:Direct ノードから Managed File Transfer Agent へ、複数のファイルを転送できます。Connect:Direct ブリッジ・エージェントをソース・エージェントとして指定し、1つ以上のソース指定を `connect_direct_node_name:file_path` の形式で指定することにより、Connect:Direct ノードを複数ファイル転送のソースとして指定できます。

関連資料

550 ページの『MFT agent.properties ファイル』

各 Managed File Transfer Agent には、独自のプロパティ・ファイル `agent.properties` があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。`agent.properties` ファイルには、エージェントの動作を変更するプロパティを含めることもできます。

Connect:Direct ノードからの複数ファイルの転送

Connect:Direct ブリッジを使用して、Connect:Direct ノードから Managed File Transfer Agent へ、複数のファイルを転送できます。Connect:Direct ブリッジ・エージェントをソース・エージェントとして指定し、1つ以上のソース指定を `connect_direct_node_name:file_path` の形式で指定することにより、Connect:Direct ノードを複数ファイル転送のソースとして指定できます。

始める前に

ファイルを転送する前に、Managed File Transfer のコンポーネントである Connect:Direct ブリッジを構成しておくことが必要です。143 ページの『Connect:Direct ブリッジの構成』を参照してください。

このタスクについて

この例では、Connect:Direct ブリッジ・エージェントは `CD_BRIDGE` という名前です。宛先エージェントは `FTE_Z` という名前で、z/OS システム上で実行されています。ソース Connect:Direct ノードは `CD_NODE1` という名前です。転送するファイルは、`CD_NODE1` があるシステム上のファイル・パス `/in/file1`、`/in/file2`、および `/in/file3` にあります。これらのファイルは、`FTE_Z` が実行されているシステム上の区分データ・セット `//OBJECT.LIB` に転送されます。

手順

ソース指定の値として `connect_direct_node_name:file_path` の形式を使用し、`-sa` パラメーターの値として Connect:Direct ブリッジ・エージェントの名前を指定して、`fteCreateTransfer` コマンドを使用します。

注：`connect_direct_node_name` で指定される Connect:Direct ノードは、ファイルの転送元のノードであり、Connect:Direct ブリッジの一部として機能する Connect:Direct ノードではありません。

```
fteCreateTransfer -sa CD_BRIDGE -da FTE_Z
                  -dp '//OBJECT.LIB' CD_NODE1:/in/file1
                  CD_NODE1:/in/file2 CD_NODE1:/in/file3
```

詳細については 437 ページの『`fteCreateTransfer`: 新規ファイル転送の開始』を参照してください。

タスクの結果

Connect:Direct ブリッジ・エージェント `CD_BRIDGE` が、Connect:Direct ノード `CD_NODE1` からの最初のファイルを要求します。Connect:Direct ノードが Connect:Direct ブリッジにファイルを送信します。Connect:Direct ノードからのファイルの転送中、Connect:Direct ブリッジは、`cdTmpDir` エージェント・プロパティで定義されている場所に一時的にそのファイルを保管します。Connect:Direct ノードから Connect:Direct ブリッジへのファイル転送が完了すると、Connect:Direct ブリッジは、そのファイルを宛先エージェント `FTE_Z` に送信し、一時ロケーションからそのファイルを削除します。指定されたすべてのソース・ファイルごとに、このプロセスが繰り返されます。

関連概念

230 ページの『[Connect:Direct ブリッジ](#)』

既存の IBM Sterling Connect:Direct ネットワークとの相互間で、ファイルを転送することができます。Managed File Transfer のコンポーネントである Connect:Direct ブリッジを使用して、MFT と IBM Sterling Connect:Direct の間でファイルを転送します。

関連資料

550 ページの『[MFT agent.properties ファイル](#)』

各 Managed File Transfer Agent には、独自のプロパティ・ファイル `agent.properties` があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。`agent.properties` ファイルには、エージェントの動作を変更するプロパティを含めることもできます。

ワイルドカードを使用した Connect:Direct への複数ファイルの転送

Managed File Transfer エージェントから Connect:Direct ノードへ複数のファイルを転送するには、Connect:Direct ブリッジを使用します。`fteCreateTransfer` コマンドに指定するソース指定内では、ワイルドカード文字を使用できます。ワイルドカードを使用するすべての Managed File Transfer の転送と同様、ファイル・パスの最後の部分にのみワイルドカード文字を使用できます。例えば、`/abc/def*` は有効なファイル・パスですが、`/abc*/def` は無効です。

始める前に

ファイルを転送する前に、Managed File Transfer のコンポーネントである Connect:Direct ブリッジを構成しておく必要があります。詳細については [143 ページの『Connect:Direct ブリッジの構成』](#) を参照してください。

このタスクについて

この例では、ソース・エージェントは `FTE_AGENT` という名前で、Connect:Direct ブリッジ・エージェントは `CD_BRIDGE` という名前です。宛先 Connect:Direct ノードは `CD_NODE1` という名前です。転送するファイルは、`FTE_AGENT` があるシステム上の `/reports` ディレクトリーにあります。ファイル名が `report` で始まり、2 文字と接尾部 `.log` が続くファイルのみが転送されます。例えば、`/reports/report01.log` というファイルは転送されますが、`/reports/report1.log` というファイルは転送されません。ファイルは、`CD_NODE1` が実行されているシステム上の `/home/fred` ディレクトリーへ転送されます。

手順

1. `-dd` (宛先ディレクトリー) パラメーターに `connect_direct_node_name:directory_path` 形式の値を使用して、`fteCreateTransfer` コマンドを使用します。`-da` (宛先エージェント) パラメーターには、Connect:Direct ブリッジ・エージェントを指定します。

注: `connect_direct_node_name` で指定される Connect:Direct ノードは、ファイルの転送先のノードであり、Connect:Direct ブリッジの一部として機能する Connect:Direct ノードではありません。

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
                  -dd CD_NODE1:/home/fred "/reports/report??.log"
```

詳細については [437 ページの『fteCreateTransfer: 新規ファイル転送の開始』](#) を参照してください。

2. ソース・エージェント `FTE_AGENT` が、パターン `/reports/report??.log` に一致する最初のファイルを Connect:Direct ブリッジ・エージェント `CD_BRIDGE` に転送します。Connect:Direct ブリッジ・エージェントがこのファイルを、`cdTmpDir` プロパティで定義された場所に一時的に保管します。ソース・エージェントから Connect:Direct ブリッジにファイルが完全に転送されると、Connect:Direct ブリッジ・エージェントがそのファイルを、`cdNode` エージェント・プロパティで定義された Connect:Direct ノードに送信します。このノードがファイルを、宛先 Connect:Direct ノード `CD_NODE1` に送信します。2 つの Connect:Direct ノード間で転送が完了すると、Connect:Direct ブリッジ・エージェントが一時ロケーションからファイルを削除します。このプロセスは、ワイルドカード・パターン `/reports/report??.log` に一致するソース・ファイルごとに繰り返されます。

注: パターン `/reports/report??.log` に一致するファイルのリストは、ソース・エージェント `FTE_AGENT` があるシステムのエージェント・システムによって異なります。

- ソース・エージェントが Windows オペレーティング・システムを使用するシステム上にある場合、パターン・マッチングは大/小文字を区別しません。パターンは、`/reports` ディレクトリー内にある、`report` の後に 2 文字と接尾部 `.log` が続く形式のファイル名を持つすべてのファイルに一致します。文字が大文字でも小文字でも関係ありません。例えば、`Report99.Log` は一致します。
- ソース・エージェントが Linux または UNIX オペレーティング・システムを使用するシステム上にある場合、パターン・マッチングは大/小文字を区別します。パターンは、`/reports` ディレクトリー内にある、`report` の後に 2 文字と接尾部 `.log` が続く形式のファイル名を持つファイルのみに一致します。例えば、`reportAB.log` は一致しますが、`reportAB.LOG` および `Report99.Log` は一致しません。

Connect:Direct ノードを転送元および転送先とする転送のリカバリーおよび再始動

転送中に、Managed File Transfer が IBM Sterling Connect:Direct ノードに接続できなくなる場合があります。例えば、ノードが使用不可になる場合です。その場合、Managed File Transfer が転送のリカバリーを試行するか、転送が失敗してエラー・メッセージが生成されます。

Connect:Direct ノードが使用不可になる場合

Connect:Direct ノードが、ネットワーク障害や電源異常などが原因で使用不可になると、Managed File Transfer は以下の方法でファイル転送をリカバリーします。

- Managed File Transfer がこの転送要求の一部として Connect:Direct ノードに以前に正常に接続されていない場合、`cdMaxConnectionRetries` および `recoverableTransferRetryInterval` **properties** の値によって決定された時間だけ転送が再試行されます。これらのプロパティーは、Connect:Direct ブリッジ・エージェントの `agent.properties` ファイル内に指定されています。試行の失敗回数が `cdMaxConnectionRetries` **property** の値に達すると、転送が失敗し、エラー・メッセージが生成されます。デフォルトでは、転送は無限に試行され、試行の間隔は 60 秒です。
- この転送要求の一部として、Managed File Transfer がこれまでこの Connect:Direct ノードとの接続に成功している場合、`cdMaxPartialWorkConnectionRetries` プロパティーおよび `recoverableTransferRetryInterval` プロパティーの値によって決定される時間の間、転送が再試行されます。失敗した試行の回数が `cdMaxPartialWorkConnectionRetries` プロパティーの値に達すると転送が失敗し、エラー・メッセージが生成されます。デフォルトでは、転送は無限に試行され、試行の間隔は 60 秒です。
- 特定のタイプの Connect:Direct ノード障害 (例えば、強制的に停止されているノード) の場合、ノードがリカバリーすると、Connect:Direct プロセスは `Held Due to Error (HE)` 状況になります。Managed File Transfer はノードのリカバリー後、状況が `HE` で、このファイル転送に関連するすべての Connect:Direct プロセスを自動的に再開します。
- 転送が失敗すると、転送に関連したすべての一時ファイルが、Connect:Direct ブリッジをホストするシステムから削除されます。これらの一時ファイルの場所は、`cdTmpDir` プロパティーによって定義されています。
- Managed File Transfer から Connect:Direct への転送で、ソースの後処理として削除が指定されている場合、転送が失敗するとソース・ファイルは削除されません。

Connect:Direct ノードのユーザー資格情報が無効な場合

Managed File Transfer から Connect:Direct ノードへの接続で、ユーザーの資格情報がノードによって拒否されたために接続が失敗すると、転送が失敗し、エラー・メッセージが生成されます。このシチュエーションでは、Connect:Direct ノードに対して、正しいユーザー資格情報が提供されていることを確認します。詳しくは、[146 ページの『Connect:Direct の資格情報のマップ』](#)を参照してください。

Connect:Direct ブリッジ・エージェントが使用不可になる場合

Connect:Direct ブリッジ・エージェントが使用不可になると、すべての進行中のファイル転送は、標準の Managed File Transfer 転送と同様にリカバリーされます。詳しくは、[246 ページの『MFT のリカバリーと再始動』](#)を参照してください。

関連概念

[230 ページの『Connect:Direct ブリッジ』](#)

既存の IBM Sterling Connect:Direct ネットワークとの相互間で、ファイルを転送することができます。Managed File Transfer のコンポーネントである Connect:Direct ブリッジを使用して、MFT と IBM Sterling Connect:Direct の間でファイルを転送します。

[246 ページの『MFT のリカバリーと再始動』](#)

エージェントまたはキュー・マネージャーが何らかの理由 (例えば、電源やネットワークの障害など) で使用できない場合、Managed File Transfer は、以下のシナリオで示すようにリカバリーを行います。

関連タスク

[143 ページの『Connect:Direct ブリッジの構成』](#)

Managed File Transfer ネットワークと Connect:Direct ネットワークの間でファイルを転送するために、Connect:Direct ブリッジを構成します。Connect:Direct ブリッジのコンポーネントは、Connect:Direct ノードと、そのノードと通信するための専用の Managed File Transfer エージェントです。このエージェントのことを Connect:Direct ブリッジ・エージェントといいます。

関連資料

[550 ページの『MFT agent.properties ファイル』](#)

各 Managed File Transfer Agent には、独自のプロパティ・ファイル `agent.properties` があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。`agent.properties` ファイルには、エージェントの動作を変更するプロパティを含めることもできます。

ファイル転送要求からのユーザー定義 Connect:Direct プロセスの送信

ファイル転送の一部としてユーザー定義 Connect:Direct プロセスを呼び出す Connect:Direct ブリッジ・エージェントを経由する転送の転送要求を送信できます。

Connect:Direct ブリッジを経由する転送のファイル転送要求を送信すると、デフォルトでは、Connect:Direct ブリッジ・エージェントがリモート Connect:Direct ノードとの間でファイルを転送するための Connect:Direct プロセスを生成します。

一方、`ConnectDirectProcessDefinition.xml` ファイルを使用して、ユーザー定義 Connect:Direct プロセスを呼び出すように Connect:Direct ブリッジ・エージェントを構成することもできます。

ConnectDirectProcessDefinition.xml ファイル

`fteCreateCDAgent` コマンドを使用すると、エージェントの構成ディレクトリー `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name` にファイル

`ConnectDirectProcessDefinitions.xml` が作成されます。Connect:Direct ブリッジ・エージェントからユーザー定義 Connect:Direct プロセスを呼び出すには、まずこのファイルを編集してプロセス定義をセットアップする必要があります。

このファイルでは、転送の一部として呼び出す 1 つ以上の Connect:Direct プロセスの場所を組み込んだプロセス・セットを 1 つ以上定義します。それぞれのプロセス・セットには、いくつかの条件を組み込みます。転送がプロセス・セットのすべての条件を満たす場合、そのプロセス・セットを使用して、転送で呼び出す Connect:Direct プロセスが指定されます。詳細については [151 ページの『ConnectDirectProcessDefinition.xml ファイルを使用して、開始する Connect:Direct プロセスを指定する操作』](#)を参照してください。

組み込みシンボリック変数

Managed File Transfer が定義する組み込みシンボリック変数を使用して、値をユーザー定義 Connect:Direct プロセスに置換できます。Connect:Direct の命名規則に合わせて、Managed File Transfer

で使用するすべての組み込みシンボリック変数は、%FTE の後に 5 つの大文字英数字を付けた形式になっています。

Connect:Direct ノードから Connect:Direct ブリッジ・システムにファイルを転送するプロセスを作成する場合、Connect:Direct プロセスの TO FILE の値として組み込み変数 %FTETFILE を使用する必要があります。Connect:Direct ブリッジ・システムから Connect:Direct ノードにファイルを転送するプロセスを作成する場合、Connect:Direct プロセスの FROM FILE の値として組み込み変数 %FTEFFILE を使用する必要があります。これらの変数には、Connect:Direct ブリッジ・エージェントが Managed File Transfer ネットワークを転送先および転送元とする転送で使用する一時ファイル・パスが含まれます。

組み込みシンボリック変数の詳細については、Connect:Direct の製品資料を参照してください。

サンプル Connect:Direct プロセス

Managed File Transfer では、サンプル Connect:Direct プロセスが提供されています。これらのサンプルは、ディレクトリ `MQ_INSTALLATION_PATH/mqft/samples/ConnectDirectProcessTemplates` にあります。

ConnectDirectProcessDefinition.xml ファイルを使用して、開始する Connect:Direct プロセスを指定する操作

Managed File Transfer 転送の一部として開始する Connect:Direct プロセスを指定します。Managed File Transfer には、プロセス定義を指定するために編集できる XML ファイルが用意されています。

このタスクについて

fteCreateCDAgent コマンドを使用すると、エージェントの構成ディレクトリ `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name` にファイル `ConnectDirectProcessDefinitions.xml` が作成されます。Connect:Direct ブリッジ・エージェントからユーザー定義 Connect:Direct プロセスを呼び出すには、まずこのファイルを編集してプロセス定義をセットアップする必要があります。

Connect:Direct ブリッジを経由した転送の一部として呼び出すように指定するプロセスごとに、以下の手順を実行します。

手順

1. 転送の一部として Connect:Direct ブリッジ・エージェントから呼び出す Connect:Direct プロセスを定義し、プロセス・テンプレートをファイルに保存します。
2. テキスト・エディターで `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name/ConnectDirectProcessDefinitions.xml` ファイルを開きます。
3. `<processSet>` エlementを作成します。
4. `<processSet>` エlementの内部に、`<condition>` エlementを作成します。
5. `<condition>` エlement内で、ステップ 1 で定義した Connect:Direct プロセスを呼び出すために転送要求が一致する必要がある条件を定義するエlementを 1 つ以上作成します。これらのエlementは、`<match>` エlementまたは `<defined>` エlementのいずれかです。
 - `<match>` エlementを使用して、変数の値がパターンに一致する必要があることを指定します。以下の属性を使用して `<match>` エlementを作成します。
 - `variable` - 値を比較する変数の名前。この変数は、組み込みシンボルです。詳しくは、[660 ページの『ユーザー定義 Connect:Direct プロセスで使用する置換変数』](#)を参照してください。
 - `value` - 指定した変数の値と比較するパターン。
 - オプション: `pattern - value` 属性の値で使用するパターンのタイプ。このパターン・タイプは、`wildcard` または `regex` のいずれかになります。この属性は任意指定であり、デフォルトは `wildcard` です。

- `<defined>`エレメントを使用して、変数に値を定義する必要があることを指定します。以下の属性を使用して`<defined>`エレメントを作成します。
 - `variable` - 値が定義されていなければならない変数の名前。この変数は、組み込みシンボルです。詳しくは、[660 ページの『ユーザー定義 Connect:Direct プロセスで使用する置換変数』](#)を参照してください。

`<condition>`エレメント内に指定された条件は、論理 AND と結合されます。Connect:Direct ブリッジ・エージェントがこの `<processSet>`エレメントによって指定されたプロセスを呼び出すには、すべての条件を満たす必要があります。`<condition>`エレメントを指定しない場合、プロセス・セットはすべての転送に一致します。

6. `<processSet>` エレメントの内部に、`<process>`エレメントを作成します。

7. `<process>` エレメントの内部に、`<transfer>`エレメントを作成します。

`transfer` エレメントでは、Connect:Direct ブリッジ・エージェントが転送の一部として呼び出す Connect:Direct プロセスを指定します。以下の属性を使用して`<transfer>`エレメントを作成します。

- `process-` - ステップ 1 で定義した Connect:Direct プロセスの場所です。このファイルの場所は、絶対パスで指定されます。または、`MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name` ディレクトリーに対する相対パスで指定します。

タスクの結果

Connect:Direct ブリッジ・エージェントは、条件に合致する項目を検索するときに、ファイルの先頭から末尾に向かって検索します。最初に見つかった一致が使用されます。

関連タスク

[143 ページの『Connect:Direct ブリッジの構成』](#)

Managed File Transfer ネットワークと Connect:Direct ネットワークの間でファイルを転送するために、Connect:Direct ブリッジを構成します。Connect:Direct ブリッジのコンポーネントは、Connect:Direct ノードと、そのノードと通信するための専用の Managed File Transfer エージェントです。このエージェントのことを Connect:Direct ブリッジ・エージェントといいます。

関連資料

[590 ページの『Connect:Direct プロセスの定義ファイルのフォーマット』](#)

Connect:Direct ブリッジ・エージェントの構成ディレクトリーにある

`ConnectDirectProcessDefinitions.xml` ファイルでは、ファイル転送の一部として開始するユーザー定義 Connect:Direct プロセスを指定します。

[402 ページの『fteCreateCDAgent \(Connect:Direct ブリッジ・エージェントの作成\)』](#)

`fteCreateCDAgent` コマンドは、Managed File Transfer Agent、および Connect:Direct ブリッジのための関連する構成を作成します。

Managed File Transfer から呼び出す Connect:Direct プロセスでの組み込みシンボリック変数の使用

Managed File Transfer の転送からユーザー定義 Connect:Direct プロセスを呼び出すときに、プロセス定義の中で組み込みシンボリック変数を使用して、転送から Connect:Direct プロセスに情報を渡すことができます。

このタスクについて

この例では、組み込みシンボリック変数を使用して、Managed File Transfer 転送からユーザー定義 Connect:Direct プロセスに情報を渡します。Managed File Transfer で使用する組み込みシンボリック変数の詳細については、[660 ページの『ユーザー定義 Connect:Direct プロセスで使用する置換変数』](#)を参照してください。

この例では、Managed File Transfer Agent から Connect:Direct ブリッジ・ノードにファイルを転送します。転送の第 1 部分を Managed File Transfer が実行します。転送の第 2 部分をユーザー定義 Connect:Direct プロセスが実行します。

手順

1. 組み込みシンボリック変数を使用する Connect:Direct プロセスを作成します。

```
%FTEPNAME PROCESS
  SNODE=%FTESNODE
  PNODEID=(%FTEPUSER,%FTEPPASS)
  SNODEID=(%FTESUSER,%FTESPASS)

COPY001 COPY
  FROM (
    FILE=%FTEFFILE
    DISP=%FTEFDISP
  )
  TO (
    FILE=%FTETFILE
    DISP=%FTETDISP
  )
PEND
```

2. このプロセスを `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent/Example.cdp` という場所にあるテキスト・ファイルに保存します。
3. `ConnectDirectProcessDefinition.xml` ファイルを編集して、手順 1 で作成した Connect:Direct プロセスを呼び出すためのルールを組み込みます。

```
<?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>
```

この例では、ソースまたは宛先の Connect:Direct ノードが TOBERMORY になっている Connect:Direct ブリッジ・エージェントに転送要求が送信されると、`Example.cdp` Connect:Direct プロセスが呼び出されます。

4. 手順 3 で定義した `ConnectDirectProcessDefinition.xml` ファイルの条件を満たすファイル転送要求を送信します。

例:

```
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
```

この例では、宛先 Connect:Direct ノードが TOBERMORY になっています。このノードは、転送の 2 次ノードであり、`%FTESNODE` の値が TOBERMORY に設定されています。このコマンドは、`ConnectDirectProcessDefinition.xml` ファイルで設定されている条件に合致します。

5. Managed File Transfer が Connect:Direct ブリッジ・エージェントと同じシステムの一時的な場所にソース・ファイルを転送します。
6. Connect:Direct ブリッジ・エージェントが転送要求と構成情報に含まれている情報に基づいて組み込みシンボリック変数の値を設定します。

組み込みシンボリック変数は、以下の値に設定されます。

- `%FTEPNAME=process_name` - この値は、Connect:Direct ブリッジ・エージェントによって生成される 8 文字のプロセス名です。

- %FTESNODE=TOBERMORY - この値は、**fteCreateTransfer** コマンドの **-df** パラメーターから設定されます。
- %FTEPUSER=*primary_node_user* - この情報は、ConnectDirectCredentials.xml ファイルから取り込まれます。
- %FTEPPASS=*primary_node_user_password* - この情報は、ConnectDirectCredentials.xml ファイルから取り込まれます。
- %FTESUSER=*secondary_node_user* - この情報は、ConnectDirectCredentials.xml ファイルから取り込まれます。
- %FTESPASS=*secondary_node_user_password* - この情報は、ConnectDirectCredentials.xml ファイルから取り込まれます。
- %FTEFFILE=*temporary_location* - この値は、Connect:Direct ブリッジ・エージェントと同じシステムの一時的なファイル保管場所です。
- %FTEFDISP=leave - この値は、**fteCreateTransfer** コマンドの **-sd** パラメーターから設定されます。
- %FTETFILE=/home/bulgaria/destination.txt - この値は、**fteCreateTransfer** コマンドの **-df** パラメーターから設定されます。
- %FTETDISP=overwrite - この値は、**fteCreateTransfer** コマンドの **-de** パラメーターから設定されます。

7. Connect:Direct ブリッジ・ノードで Connect:Direct プロセスが開始されます。Connect:Direct が Connect:Direct ブリッジのシステムの一時的な場所から Connect:Direct ノード TOBERMORY が稼働しているシステムの宛先 /home/bulgaria/destination.txt にファイルを転送します。

関連概念

239 ページの『[ファイル転送要求からのユーザー定義 Connect:Direct プロセスの送信](#)』

ファイル転送の一部としてユーザー定義 Connect:Direct プロセスを呼び出す Connect:Direct ブリッジ・エージェントを経由する転送の転送要求を送信できます。

関連資料

660 ページの『[ユーザー定義 Connect:Direct プロセスで使用する置換変数](#)』

Managed File Transfer の固有の組み込みシンボリック変数を使用して、ユーザー定義 Connect:Direct プロセスで置換する値を定義できます。

Connect:Direct プロセスを使用して Managed File Transfer 転送要求を送信する操作

Connect:Direct プロセスから Connect:Direct ブリッジ・エージェントに転送要求を送信できます。

Managed File Transfer には、Connect:Direct プロセスの **RUN TASK** ステートメントから呼び出すことができるコマンドが用意されています。

Managed File Transfer には、Connect:Direct プロセスで使用できる以下のコマンドが用意されています。

ftetag

ftebxfer、**ftecxfer** の各コマンドの前のステップでこのコマンドを指定して、転送に関する必要な監査情報を作成します。このコマンドでは、転送のソース指定をパラメーターとして使用します。ソース指定の形式については、[437 ページの『fteCreateTransfer: 新規ファイル転送の開始』](#)を参照してください。

ftebxfer

転送要求の送信先のキュー・マネージャーが、コマンドを送信する Connect:Direct ノードと同じシステムに存在する場合は、このコマンドを指定して、ファイル転送要求を作成します。このコマンドでは、**fteCreateTransfer** コマンドと同じパラメーターを使用します。これらのパラメーターについて詳しくは、[437 ページの『fteCreateTransfer: 新規ファイル転送の開始』](#)を参照してください。このコマンドには、さらに追加のパラメーターがあります。

-qmgrname

必須。コマンドの送信先のキュー・マネージャーの名前。

ftecxfer

転送要求の送信先のキュー・マネージャーが、コマンドを送信する Connect:Direct ノードとは別のシステムに存在する場合は、このコマンドを指定して、ファイル転送要求を作成します。このコマンドでは、**ftcCreateTransfer** コマンドと同じパラメーターを使用します。パラメーターについて詳しくは、437 ページの『[ftcCreateTransfer: 新規ファイル転送の開始](#)』を参照してください。このコマンドには、さらに追加のパラメーターが 3 つあります。

-qmgrname

必須。コマンドの送信先のキュー・マネージャーの名前。

-connname

必須。コマンドの送信先のキュー・マネージャーのホストとポート。IBM MQ の CONNAME の形式で指定します。例えば、host.example.com(1337). などです。

-channelname

オプション。コマンドの送信先のキュー・マネージャーに接続するときに使用するチャンネルの名前。指定しない場合は、デフォルト値の SYSTEM.DEF.SVRCONN が使用されます。

関連タスク

244 ページの『[Connect:Direct Requester を使用して、Managed File Transfer を呼び出す Connect:Direct プロセスを作成して送信する操作](#)』

Connect:Direct 要求者は、Managed File Transfer を呼び出す Connect:Direct プロセスを作成および実行依頼するために使用できるグラフィカル・ユーザー・インターフェースです。

関連資料

663 ページの『[例: MFT コマンドを呼び出す Connect:Direct プロセス・ファイル](#)』

Managed File Transfer の **ftetag** コマンドと **ftecxfer** コマンドを呼び出す Connect:Direct プロセス・ファイルの例。

Connect:Direct Requester を使用して、Managed File Transfer を呼び出す Connect:Direct プロセスを作成して送信する操作

Connect:Direct 要求者は、Managed File Transfer を呼び出す Connect:Direct プロセスを作成および実行依頼するために使用できるグラフィカル・ユーザー・インターフェースです。

このタスクについて

このタスクでは、Managed File Transfer の **ftecxfer** コマンドまたは **ftebxfer** コマンドを呼び出す Connect:Direct プロセスの作成方法を示します。転送要求の送信先であるキュー・マネージャーが、コマンドを送信する Connect:Direct ノードとは異なるシステム上にある場合は、**ftecxfer** コマンドを使用します。転送要求の送信先であるキュー・マネージャーが、コマンドを送信する Connect:Direct ノードと同じシステム上にある場合は、**ftebxfer** コマンドを使用します。**ftecxfer** コマンドは、転送のソース・エージェントのエージェント・キュー・マネージャーに対するクライアント接続を確立します。**ftecxfer** コマンドを呼び出す前に、**ftetag** コマンドを呼び出して、ソースの指定情報を渡す必要があります。このようにすれば、Managed File Transfer から開始した転送の場合と同じ要領で、プロセスのログを記録して監査することが可能になります。

手順

1. Connect:Direct Requester を開始します。
2. パネルの「ノード」タブで、プロセスの 1 次ノードとして使用する Connect:Direct ノードを選択します。
3. 「ファイル」 > 「新規」 > 「プロセス」を選択します。「プロセス・プロパティー」ウィンドウが開きます。
4. 「名前:」フィールドにプロセスの名前を入力します。
5. 「Snode」 > 「名前:」リストから 2 次ノードを選択します。
6. 「Snode」 > 「オペレーティング・システム:」リストから 2 次ノードのオペレーティング・システムを選択します。
7. オプション: このウィンドウで必要な情報をさらに入力します。

8. 「OK」をクリックします。「プロセス・プロパティ」ウィンドウが閉じます。
9. Managed File Transfer の **ftetag** コマンドを実行するステートメントを作成します。
 - a) 「プロセス」ウィンドウで **End** ステートメントを右クリックします。
 - b) 「挿入」 > 「タスクの実行」を選択します。「タスク実行ステートメント」ウィンドウが開きます。
 - c) 「ラベル:」フィールドに Tag と入力します。
 - d) 「オプション・パラメーターまたはオプション・コマンド」フィールドに `pgm(MQ_INSTALLATION_PATH/bin/ftetag) args(source_specification)` と入力します。 `source_specification` のフォーマットについて詳しくは、[437 ページの『fteCreateTransfer: 新規ファイル転送の開始』](#)を参照してください。
 - e) 「OK」をクリックします。「タスク実行ステートメント」ウィンドウが閉じます。
10. Managed File Transfer の **ftecxfer** コマンドまたは **ftebxfer** コマンドを実行するステートメントを作成します。
 - a) 「プロセス」ウィンドウで **End** ステートメントを右クリックします。
 - b) 「挿入」 > 「タスクの実行」を選択します。「タスク実行ステートメント」ウィンドウが開きます。
 - c) 「ラベル:」フィールドに Transfer と入力します。
 - d) 選択するコマンドに応じて、「オプション・パラメーターまたはオプション・コマンド」フィールドに `pgm(MQ_INSTALLATION_PATH/bin/ftecxfer) args(parameters)` または `pgm(MQ_INSTALLATION_PATH/bin/ftebxfer) args(parameters)` と入力します。**ftecxfer** コマンドおよび **ftebxfer** コマンドで使用するパラメーターは、**fteCreateTransfer** コマンドで使用するパラメーターと同じですが、**ftecxfer** および **ftebxfer** に特定のパラメーターもいくつかあります。詳細については、[437 ページの『fteCreateTransfer: 新規ファイル転送の開始』](#) および [243 ページの『Connect:Direct プロセスを使用して Managed File Transfer 転送要求を送信する操作』](#)を参照してください。
 - e) 「OK」をクリックします。「タスク実行ステートメント」ウィンドウが閉じます。
11. オプション: 必要なステートメントをさらに作成します。
12. プロセスを送信します。
 - a) 「プロセス」ウィンドウで右クリックします。
 - b) 「実行依頼」を選択します。「Connect:Direct 接続」ウィンドウが開きます。
 - c) プロセスを実行するために使用するユーザー名とパスワードを入力します。
 - d) 「OK」をクリックします。

関連概念

[243 ページの『Connect:Direct プロセスを使用して Managed File Transfer 転送要求を送信する操作』](#)

Connect:Direct プロセスから Connect:Direct ブリッジ・エージェントに転送要求を送信できます。

Managed File Transfer には、Connect:Direct プロセスの **RUN TASK** ステートメントから呼び出すことができるコマンドが用意されています。

IBM Integration Bus からの MFT の操作

FTEOutput ノードと FTEInput ノードを使用すれば、IBM Integration Bus から Managed File Transfer を操作できます。

- FTEInput ノードを使用すると、Managed File Transfer を使用してネットワークでファイルを転送し、そのファイルを Integration Bus フローの一部として処理できます。
- FTEOutput ノードを使用すると、Integration Bus フローで出力されたファイルをネットワーク内の別の場所に転送できます。

ブローカー・エージェントとの間でファイルを転送するエージェントは、Managed File Transfer のどのレベルでも構いません。

詳しくは、[IBM Integration Bus 製品資料](#)を参照してください。

MFTのリカバリーと再始動

エージェントまたはキュー・マネージャーが何らかの理由(例えば、電源やネットワークの障害など)で使用できない場合、Managed File Transfer は、以下のシナリオで示すようにリカバリーを行います。

- 通常、ファイルの転送中に問題が発生すると、Managed File Transfer は、問題が修復された後にそのファイル転送をリカバリーおよび再開します。
- エージェントまたはキュー・マネージャーが使用できなくなっている間に、転送処理中のファイルが削除または変更されると、転送は失敗し、その失敗に関する詳細を示すメッセージが転送ログに記録されません。
- ファイル転送中にエージェント・プロセスが失敗しても、エージェントを再始動すると、転送が続行されます。
- エージェントがエージェント・キュー・マネージャーへの接続を失うと、エージェントはキュー・マネージャーへの再接続を試行する間、待機状態になります。エージェントがキュー・マネージャーに正常に再接続すると、現在の転送を続行します。
- エージェントが何らかの理由で停止した場合、エージェントに関連付けられているリソース・モニターはすべてポーリングを停止します。エージェントがリカバリーすると、モニターも再始動されて、リソースのポーリングも再開します。
- ソースのファイル属性指定が `delete` に設定されたファイル転送の場合、ソース・エージェントから宛先エージェントにすべてのデータが送信された後にリカバリーが発生すると、ソース・ファイルは削除の前にアンロックされます。このアンロックの影響で、ソース・ファイルが削除される前にファイルが変更される可能性があります。したがって、ソース・ファイルの削除は安全ではないと見なされるため、次の警告が表示されます。

```
BFGTR0075W: The source file has not been deleted because it is possible that the source file was modified after the source file was transferred.
```

この場合は、ソース・ファイルの内容が変更されていないことを確認してから、手動でソース・ファイルを削除してください。

転送の状況については、IBM MQ Explorer で確認することができます。いずれかの転送が `Stalled` として表示される場合、停止状況はエージェントの問題、または転送に関与する 2 つのエージェントの間の問題のいずれかを示すため、修正アクションを実行する必要がある場合があります。

関連概念

[246 ページの『ファイル転送のリカバリーのタイムアウト・オプション』](#)

停止したファイル転送のリカバリーをソース・エージェントが試行し続ける時間を秒単位で設定できます。転送が成功しないままエージェントが再試行間隔のタイムアウトに達した場合、その転送は失敗します。

V 9.0.1 ファイル転送のリカバリーのタイムアウト・オプション

停止したファイル転送のリカバリーをソース・エージェントが試行し続ける時間を秒単位で設定できます。転送が成功しないままエージェントが再試行間隔のタイムアウトに達した場合、その転送は失敗します。

IBM MQ 9.0.1 より前は、Managed File Transfer ソース・エージェントのデフォルトの動作では、停止した転送のリカバリーを転送が成功するまで試行し続けていました。この新しいパラメーターはオプションであるため、設定は必須ではありません。設定しなければ、デフォルトの動作に従った転送が行われます。

IBM MQ 9.0.1 以降、以下のいずれかのオプションを指定できます。

-1

エージェントは、停止した転送のリカバリーを、転送が成功するまで試行し続けます。このオプションを使用すると、このプロパティを設定しない場合のエージェントのデフォルトの動作と同じになります。

0

エージェントは、リカバリーに入るとすぐにファイル転送を停止します。

>0

エージェントは、指定された正整数値で設定された時間 (秒単位) だけ、停止した転送のリカバリーを試行し続けます。値 21600 は、エージェントがリカバリーに入ってから 6 時間にわたって転送のリカバリーを試行し続けることを示しています。このパラメーターの最大値は 999999999 です。

あるソース・エージェントのすべての転送に対するリカバリー再試行タイムアウトの設定

あるソース・エージェントのすべての転送に適用されるリカバリー・タイムアウトを設定するには、パラメーターと値のペアを `agent.properties` ファイルに追加します。

次の例では、転送リカバリー・タイムアウト値を `-1` に設定しているため、エージェントは停止した転送のリカバリーを転送が正常に完了するまで試行し続けます。

```
transferRecoveryTimeout=-1
```

次のように転送リカバリー・タイムアウト値を `0` に設定した場合、エージェントは、リカバリーに入るとすぐに転送に失敗のマークを付けます。

```
transferRecoveryTimeout=0
```

次のように転送リカバリー・タイムアウト値を `21600` に設定した場合、エージェントは、停止した転送を 6 時間再試行し続けた後に、転送に失敗のマークを付けます。

```
transferRecoveryTimeout=21600
```

個々の転送に対するリカバリー再試行タイムアウトの設定またはオーバーライド

次を行うときに、個々の転送に対してリカバリー再試行タイムアウト・パラメーターを設定することができます。

- Managed File Transfer コマンド行インターフェースまたは IBM MQ Explorer で `fteCreateTransfer` コマンドを使用して転送を作成する
- Managed File Transfer コマンド・ライン・インターフェースまたは IBM MQ Explorer で [421 ページの『fteCreateTemplate: 新規ファイル転送テンプレートの作成』](#) コマンドを使用して転送テンプレートを作成する
- Managed File Transfer コマンド行インターフェースで `fteCreateMonitor` を使用して、または IBM MQ Explorer を使用して、モニターを作成するとき。
- `fte:filecopy` Ant タスクまたは `fte:filemove` Ant タスクを使用して、ファイルをコピーまたは移動するとき。

転送の `-rt` 値を `-1` に設定するとデフォルトの動作と同じになり、転送が成功するかユーザーによって手動で取り消されるまでリカバリーが実行されます。次に例を示します。

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

新しく転送を作成するとき転送のリカバリー・タイムアウト・パラメーターを使用する方法については、[fteCreateTransfer コマンド](#)を参照してください。

`-rt` パラメーター値を `0` に設定することは、このテンプレートを使用して開始された転送が停止した場合に、転送がただちに失敗し、リカバリーはまったく試行されないということを示します。次に例を示します。

```
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
```

新しく転送テンプレートを作成するとき転送のリカバリー・タイムアウト・パラメーターを使用する方法については、[fteCreateTemplate コマンド](#)を参照してください。

IBM MQ Explorer を使用して、転送のリカバリー・タイムアウト・パラメーターと値を設定できます。IBM MQ Explorer を使用して転送を構成する方法については、160 ページの『新規ファイル転送の開始』および 197 ページの『IBM MQ Explorer を使用したファイル転送テンプレートの作成』を参照してください。

Ant タスクを使用してファイルを移動またはコピーする場合は、**fte:filecopy** または **fte:filemove** で **transferRecoveryTimeout** のオプションと値を指定してリカバリー・タイムアウトを設定することもできます。次に例を示します。

```
<fte:filecopy cmdqmq="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>
```

このファイル・コピー・タスクが開始され、転送がリカバリーに入るとすぐに転送は停止します。リカバリーは試行されません。**fte:filecopy** または **fte:filemove** で **transferRecoveryTimeout** オプションを設定すると、agent.properties ファイルに設定された値はオーバーライドされます。

fte:filecopy または **fte:filemove** で **transferRecoveryTimeout** 値を設定しない場合は、agent.properties ファイルの **transferRecoveryTimeout** パラメーター値が使用されます。詳しくは、853 ページの『fte:filecopy の Ant タスク』および 856 ページの『fte:filemove の Ant タスク』を参照してください。

リカバリー・タイムアウトの優先順位の扱い

転送作成コマンド、テンプレート・コマンド、またはモニター・コマンドのコマンド行インターフェース引数 (IBM MQ Explorer ウィザードでのオプションの設定を含む) で指定された転送リカバリー・タイムアウト値、または **fte:filespec** ネスト・エレメントで指定された転送リカバリー・タイムアウト値は、ソース・エージェントの agent.properties ファイルで **transferRecoveryTimeout** パラメーターに指定された値よりも優先されます。次のコマンドを例に挙げます。

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

このコマンドでは、**-rt** パラメーターと値のペアを指定せずに開始しているため、ソース・エージェント AGENT1 は agent.properties ファイルの **transferRecoveryTimeout** 値をチェックし、リカバリー・タイムアウト動作を決定します。

agent.properties ファイルに **transferRecoveryTimeout** が設定されていない場合、または **-1** に設定されている場合、エージェントはデフォルトの動作に従って転送が成功するまで転送のリカバリーを試行します。

Managed File Transfer コマンド行インターフェースで、例えば **fteCreateTransfer** コマンドにリカバリー・タイムアウト・オプション **-rt** を指定した場合、その値は agent.properties ファイルの値よりも優先され、転送の設定として使用されます。

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

リカバリー・タイムアウト・カウンターの扱い

転送がリカバリー状態に入ると、リカバリー・タイムアウト・カウンターが開始されます。転送状況がリカバリーに変わったことと、その状況が変わったソース・エージェントのクロック時刻を示す転送ログ・メッセージが、Log/agent_name/transfer_ID というトピック・ストリングで SYSTEM.FTE トピックにパブリッシュされます。設定された再試行間隔内で転送が再開されて、リカバリー・タイムアウトに達しなかった場合 (カウンター <= リカバリー・タイムアウト)、転送がリカバリーに入ったときに再び開始できるようにカウンターは 0 にリセットされます。

リカバリー・タイムアウトとして設定された最大値にカウンターが達すると (カウンター == リカバリー・タイムアウト)、転送のリカバリーは停止し、ソース・エージェントはその転送を失敗として報告します。

転送がリカバリー・タイムアウトに達したために発生するこの種の転送失敗は、新しいメッセージ・コード RECOVERY TIMEOUT (69) で示されます。転送が失敗したことを示す別の転送ログ・メッセージが、Log/agent_name/transfer_ID というトピック・ストリングで SYSTEM.FTE トピックにパブリッシュされます。この転送ログ・メッセージには、新しいメッセージ、新しい戻りコード、およびソース・エージェントのイベント・ログが含まれます。リカバリー中に以下のいずれかのイベントが発生すると、ソース・エージェントのイベント・ログがメッセージで更新されます。

- -1 よりも大きい値がリカバリー・タイムアウト・パラメーターに設定されると、Managed File Transfer はリカバリーに入ります。エージェントのイベント・ログが更新されて、**TransferId** のリカバリー・タイマーの開始と、ソース・エージェントがリカバリー・タイムアウト処理を開始するまで待機する時間が示されます。
- リカバリー中の Managed File Transfer が再開されると、ソース・エージェントのイベント・ログが新しいメッセージで更新されて、リカバリー状態だった **TransferId** が再開されたことが示されます。
- リカバリー中の Managed File Transfer がタイムアウトになると、ソース・エージェントのイベント・ログが更新されて、リカバリー中にリカバリー・タイムアウトのために失敗した **TransferId** が示されます。

これらのログ・メッセージから、ユーザー(サブスクライバーおよびロガー)は、転送リカバリー・タイムアウトのために失敗した転送を特定できます。

リカバリー・タイムアウトのカウンターは、常にソース・エージェント側にあります。ただし、ソース・エージェントからの情報を宛先エージェントがタイムリーに受信できなかった場合、宛先エージェントは転送のリカバリーを開始するように求める要求をソース・エージェントに送信できます。リカバリー・タイムアウト・オプションが設定された転送の場合、ソース・エージェントは宛先エージェントからこの要求を受信すると、リカバリー・タイムアウト・カウンターの開始します。

リカバリー・タイムアウト・オプションを使用しない転送、失敗した転送、および部分的に完了した転送については、引き続き手動処理が必要になります。

複数のファイルに対して単一の転送要求が発行される転送セットの場合、正常に完了したファイルが複数あっても、部分的にしか完了しなかったファイルが 1 つあると、その転送には失敗のマークが付けられます。予期されるとおりには完了しなかったからです。部分的に完了したファイルを転送中にソース・エージェントがタイムアウトになった可能性があります。

宛先エージェントとファイル・サーバーが作動可能でファイル転送を受け入れる状態にあることを確認してください。

セット全体の転送要求を再発行する必要があります。ただし、最初に転送を試行したときのファイルの一部が宛先に残っているために起きる問題を避けるために、「既存の場合は上書き」オプションを指定して新しい要求を発行してください。これにより、宛先にファイルを再度書き込む前に、前回の転送試行で生じた不完全なファイル・セットを、新しい転送の一環としてクリーンアップします。

トレースとメッセージ

診断の目的でトレース・ポイントが組み込まれています。リカバリー・タイムアウト値、再試行間隔の開始、再開期間の開始とカウンターのリセット、および転送がタイムアウトになって失敗したのかどうか、ログに記録されます。問題や予期しない動作が発生した場合、IBM サポートから要求されたら、トラブルシューティングに役立つように、ソース・エージェントの出力ログとトレース・ファイルを収集して提供してください。

ユーザーにはメッセージで、転送がリカバリーに入った (BFGTR0081I)、リカバリーがタイムアウトで強制終了した (BFGSS0081E)、リカバリー状態になった後に再開された (BFGTR0082I) ことが通知されます。

BFGTR0001 - BFGTR9999

BFGTR0081I

BFGTR0082I

BFGSS0001 - BFGSS9999

BFGSS0081E

MFT 用のアプリケーションの開発

MFT で実行するプログラムの指定

Managed File Transfer Agent が実行されているシステムで、プログラムを実行することができます。ファイル転送要求の一部として、転送の開始前または終了後のいずれかにプログラムを実行するように指定することができます。また、管理対象呼び出し要求を実行依頼することで、ファイル転送要求に含まれないプログラムを開始することも可能です。

以下の 5 つのシナリオで、プログラムの実行を指定することができます。

- 転送要求の一部として、転送の開始前にソース・エージェントで実行する
- 転送要求の一部として、転送の開始前に宛先エージェントで実行する
- 転送要求の一部として、転送の完了後にソース・エージェントで実行する
- 転送要求の一部として、転送の完了後に宛先エージェントで実行する
- 転送要求の一部としてではなく実行する。プログラム実行の要求を、エージェントに実行依頼することができます。このシナリオは、管理対象呼び出しと呼ばれる場合があります。

ユーザー出口とプログラム呼び出しは、以下の順序で呼び出されます。

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

注:

1. **宛先の転送終了出口** は、転送が正常に完了したか、部分的に正常に完了した場合にのみ実行されます。
2. **postDestinationCall** は、転送が正常に完了したか、または部分的に正常に完了した場合にのみ実行されます。
3. **SourceTransferEndExits** は、成功した転送、部分的に成功した転送、または失敗した転送に対して実行されます。
4. **postSourceCall** は、以下の場合にのみ呼び出されます。
 - 転送は取り消されませんでした。
 - 成功した結果または部分的に成功した結果があります。
 - 宛先転送後プログラムは正常に実行されました。

実行するプログラムを指定する方法は複数存在します。以下のオプションがあります。

Apache Ant タスクを使用する

`fte:filecopy`、`fte:filemove`、および `fte:call` のいずれか 1 つの Ant タスクを使用してプログラムを開始します。Ant タスクを使用する場合、ネストされたエレメント `fte:presrc`、`fte:predst`、`fte:postdst`、`fte:postsrc`、および `fte:command` を使用することにより、5 つのシナリオのどれにおいてもプログラムを指定することができます。詳細については [869 ページの『プログラム呼び出しのネスト・エレメント』](#) を参照してください。

ファイル転送要求メッセージを編集する

転送要求によって生成された XML を編集することができます。この方法を使用する場合、**preSourceCall**、**postSourceCall**、**preDestinationCall**、**postDestinationCall**、および **managedCall** の各エレメントを XML ファイルに追加することによって、5 つのシナリオのどれにおいてもプログラムを実行できます。その後、例えば `fteCreateTransfer -td` パラメーターを付けるな

どして、この変更した XML ファイルを新規ファイル転送要求の転送定義として使用します。詳細については [796 ページの『MFT エージェント呼び出し要求メッセージの例』](#) を参照してください。

fteCreateTransfer コマンドを使用する

fteCreateTransfer コマンドを使用して、開始するプログラムを指定することができます。このコマンドを使用して、最初の 4 つのシナリオで転送要求の一部としてプログラムを実行するように指定することはできますが、管理対象呼び出しを開始することはできません。使用するパラメーターについては、[437 ページの『fteCreateTransfer: 新規ファイル転送の開始』](#) を参照してください。このコマンドの使用例については、[844 ページの『fteCreateTransfer を使用してプログラムを開始する例』](#) を参照してください。

関連資料

[365 ページの『commandPath MFT プロパティ』](#)

commandPath プロパティを使用して、Managed File Transfer がコマンドを実行できる場所を指定します。指定された **commandPath** にある任意のコマンドは、エージェントにコマンドを送信できるリモート・クライアント・システムから事実上呼び出すことができるため、このプロパティを設定するときは十分な注意を払ってください。

管理対象呼び出し

Managed File Transfer (MFT) エージェントは、通常、ファイルまたはメッセージの転送に使用されます。これらは管理対象転送と呼ばれます。エージェントは、ファイルやメッセージを転送することなく、コマンド、スクリプト、または JCL を実行するためにも使用できます。この機能は、管理対象呼び出しと呼ばれます。

管理対象呼び出し要求は、以下のいくつかの方法でエージェントに実行依頼できます。

- **fte: call Ant** タスクを使用する。
- コマンドまたはスクリプトを実行するタスク XML を使用してリソース・モニターを構成する。詳しくは、[コマンドおよびスクリプトを開始するためのモニター・タスクの構成](#) を参照してください。
- エージェントのコマンド・キューに XML メッセージを直接書き込みます。Managed Call XML スキーマについて詳しくは、[ファイル転送要求メッセージ・フォーマット](#) を参照してください。

管理対象呼び出しの場合、エージェント・プロパティ **commandPath** に、実行されるコマンドまたはスクリプトを含むディレクトリーを指定する必要があります。

管理対象呼び出しでは、エージェントの **commandPath** で指定されていないディレクトリーにあるコマンドやスクリプトを実行することはできません。これは、エージェントが悪意のあるコードを実行しないようにするためです。

さらに、エージェントの権限検査を有効にして、許可されたユーザーのみが管理対象呼び出し要求を実行依頼できるようにすることもできます。これについて詳しくは、[MFT エージェント・アクションに対するユーザー権限の制限](#) を参照してください。

管理対象呼び出しの一部として呼び出されるコマンド、スクリプト、または JCL は、エージェントによってモニターされる外部プロセスとして実行されます。プロセスが終了すると、管理対象呼び出しが完了し、プロセスからの戻りコードが、**fte: Ant call Ant** タスクを呼び出したエージェントまたはスクリプトのいずれかで使用可能になります。

管理対象呼び出しが **fte: call Ant** タスクによって開始された場合、Ant スクリプトは戻りコードの値を検査して、管理対象呼び出しが成功したかどうかを判別できます。

その他のすべてのタイプの管理対象呼び出しでは、管理対象呼び出しが正常に完了したことを示すために使用する戻りコード値を指定できます。エージェントは、外部プロセスの終了時に、プロセスからの戻りコードをこれらの戻りコードと比較します。

注: 管理対象呼び出しは外部プロセスとして実行されるため、いったん開始すると取り消すことはできません。

管理対象コールとソース転送スロット

エージェントには、拡張エージェント・プロパティ:転送制限で説明されているように、エージェント・プロパティ **maxSourceTransfers** で指定された数のソース転送スロットが含まれています。

管理対象呼び出しまたは管理対象転送が実行されるたびに、ソース転送スロットが占有されます。このスロットは、管理対象通話または管理対象転送が完了すると解放されます。

エージェントが新規の管理対象呼び出し要求または管理対象転送要求のいずれかを受信したときに、すべてのソース転送スロットが使用中である場合、スロットが使用可能になるまで、要求はエージェントによってキューに入られます。

管理対象呼び出しが管理対象転送を開始する場合 (例えば、管理対象呼び出しが Ant スクリプトを実行し、その Ant スクリプトが fte: filecopy タスクまたは fte: filemove タスクを使用してファイルを転送する場合)、以下の 2 つのソース転送スロットが必要です。

- 1 つは管理対象転送用
- 1 つは管理対象呼び出し用

この状況では、管理対象転送が完了するまでに時間がかかるか、リカバリー状態になると、管理対象転送が完了するか取り消されるまで、2 つのソース転送スロットが占有されることに注意することが重要です。

このため、長期間ソース転送スロットを占有しないようにするために、管理対象呼び出しの設計を検討する必要があります。

MFT での Apache Ant の使用

Managed File Transfer は、ファイル転送機能を Apache Ant ツールに組み込むために使用できるタスクを提供します。

fteAnt コマンドを使用すると、既に構成済みの Managed File Transfer 環境で Ant タスクを実行できます。Ant スクリプトのファイル転送 Ant タスクを使用し、逐次実行形式のスクリプト言語から複雑なファイル転送操作を統合することができます。

Apache Ant について詳しくは、Apache Ant プロジェクト Web ページ (<https://ant.apache.org/>) を参照してください。

関連概念

252 ページの『[MFT での Ant スクリプトの使用入門](#)』

Managed File Transfer で Ant スクリプトを使用して、逐次実行形式のスクリプト言語から複雑なファイル転送操作を統合することができます。

関連資料

848 ページの『[MFT 提供の Ant タスク](#)』

Managed File Transfer では、数多くの Ant タスクが用意されており、これらのタスクを使用して、ファイル転送機能にアクセスできます。

379 ページの『[fteAnt: MFT で Ant タスクを実行します。](#)』

fteAnt コマンドは、Managed File Transfer の Ant タスクが使用可能な環境で Ant スクリプトを実行します。

254 ページの『[MFT のサンプル Ant タスク](#)』

Managed File Transfer のインストールと一緒に、多数のサンプル Ant スクリプトが提供されます。これらのサンプルは、`MQ_INSTALLATION_PATH/mqft/samples/fteant` ディレクトリにあります。それぞれのサンプル・スクリプトには `init` ターゲットがあります。ご使用の構成でこれらのスクリプトを実行するには `init` ターゲットで設定されたプロパティを編集します。

MFT での Ant スクリプトの使用入門

Managed File Transfer で Ant スクリプトを使用して、逐次実行形式のスクリプト言語から複雑なファイル転送操作を統合することができます。

Ant スクリプト

Ant スクリプト (またはビルド・ファイル) は、1 つ以上のターゲットを定義している XML 文書です。これらのターゲットには、実行するタスク・エレメントが含まれます。Managed File Transfer は、ファイル転送機能を Apache Ant に組み込むために使用できるタスクを提供します。Ant スクリプトについて詳しくは、Apache Ant プロジェクト Web ページ (<https://ant.apache.org/>) を参照してください。

Managed File Transfer タスクを使用する Ant スクリプトの例は、製品インストールのディレクトリー `MQ_INSTALLATION_PATH/mqft/samples/fteant` にあります。

プロトコル・ブリッジ・エージェントでは、Ant スクリプトはプロトコル・ブリッジ・エージェントのシステムで実行されます。これらの Ant スクリプトは、FTP または SFTP サーバー上のファイルには直接アクセスすることはありません。

名前空間

名前空間は、ファイル転送用の Ant タスクを、同じ名前を共有している可能性のある別の Ant タスクと区別するために使用します。名前空間は、Ant スクリプトのプロジェクト・タグに定義します。

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns:fte="antlib:com.ibm.wmqfte.ant.taskdefs" default="do_ping">

  <target name="do_ping">
    <fte:ping cmdqm="qm@localhost@1414@SYSTEM.DEF.SVRCONN" agent="agent1@qm1"
      rcproperty="ping.rc" timeout="15"/>
  </target>

</project>
```

属性 `xmlns:fte="antlib:com.ibm.wmqfte.ant.taskdefs"` は、Ant にライブラリー `com.ibm.wmqfte.ant.taskdefs` にある接頭部 `fte` が付いたタスクの定義を探すよう命令します。

`fte` を名前空間の接頭部として使用する必要はなく、どの値でも使用できます。名前空間の接頭部 `fte` は、すべての例およびサンプル Ant スクリプトで使用されます。

Ant スクリプトの実行

ファイル転送 Ant タスクを含む Ant スクリプトを実行するには、**fteAnt** コマンドを使用します。以下に例を示します。

```
fteAnt -file ant_script_location/ant_script_name
```

詳しくは、[379 ページの『fteAnt: MFT で Ant タスクを実行します。』](#)を参照してください。

戻りコード

ファイル転送 Ant タスクは、Managed File Transfer コマンドと同じ戻りコードを返します。詳しくは、[338 ページの『MFT の戻りコード』](#)を参照してください。

関連資料

[848 ページの『MFT 提供の Ant タスク』](#)

Managed File Transfer では、数多くの Ant タスクが用意されており、これらのタスクを使用して、ファイル転送機能にアクセスできます。

[379 ページの『fteAnt: MFT で Ant タスクを実行します。』](#)

fteAnt コマンドは、Managed File Transfer の Ant タスクが使用可能な環境で Ant スクリプトを実行します。

[254 ページの『MFT のサンプル Ant タスク』](#)

Managed File Transfer のインストールと一緒に、多数のサンプル Ant スクリプトが提供されます。これらのサンプルは、`MQ_INSTALLATION_PATH/mqft/samples/fteant` ディレクトリーにあります。それぞれのサンプル・スクリプトには `init` ターゲットがあります。ご使用の構成でこれらのスクリプトを実行するには `init` ターゲットで設定されたプロパティを編集します。

MFT のサンプル Ant タスク

Managed File Transfer のインストールと一緒に、多数のサンプル Ant スクリプトが提供されます。これらのサンプルは、`MQ_INSTALLATION_PATH/mqft/samples/fteant` ディレクトリーにあります。それぞれのサンプル・スクリプトには `init` ターゲットがあります。ご使用の構成でこれらのスクリプトを実行するには `init` ターゲットで設定されたプロパティーを編集します。

E メール

email サンプルは、ファイルを転送し、転送が失敗した場合は指定した E メール・アドレスに E メールを送信する Ant タスクの使用例を示しています。このスクリプトは、ソース・エージェントと宛先エージェントがアクティブであり、Managed File Transfer `ping` タスクを使用して転送を処理できることを確認します。両方のエージェントがアクティブである場合、スクリプトは Managed File Transfer `filecopy` タスクを使用して、元のファイルを削除することなく、ソース・エージェントと宛先エージェントとの間でファイルを転送します。転送が失敗すると、スクリプトは標準 Ant `email` タスクを使用して、失敗についての情報を含む E メールを送信します。

ハブ

hub サンプルは、`hubcopy.xml` と `hubprocess.xml` の 2 つのスクリプトで構成されています。`hubcopy.xml` スクリプトは、Ant スクリプトを使用して「ハブ・アンド・スポーク」スタイルのトポロジーを作成する方法を示しています。このサンプルでは、2 つのファイルがスポーク・マシン上で実行中のエージェントからハブ・マシン上で実行中のエージェントに転送されます。2 つのファイルは同時に転送され、転送が完了すると、`hubprocess.xml` Ant スクリプトがハブ・マシン上で実行されてファイルを処理します。両方のファイルが正しく転送されると、Ant スクリプトはそれらのファイルの内容を連結します。ファイルが正しく転送されない場合、Ant スクリプトは転送されたすべてのファイル・データを削除してクリーンアップします。この例が正しく機能するには、`hubprocess.xml` スクリプトをハブ・エージェントのコマンド・パスに置く必要があります。エージェントのコマンド・パスの設定について詳しくは、『`commandPath`』を参照してください。

librarytransfer (IBM i プラットフォームのみ)

IBM i

IBM i

`librarytransfer` サンプルは、Ant タスクを使用して 1 つの IBM i システムの IBM i ライブラリーを別の IBM i システムに転送する方法を示しています。

IBM i

IBM i の IBM WebSphere MQ File Transfer Edition 7.0.2 には、ネイティブ IBM i ライブラリー・オブジェクトの転送のための直接的なサポートが組み込まれていません。`librarytransfer` サンプルは、IBM i のネイティブの保存ファイル・サポートと、Managed File Transfer に付属する事前定義の Ant タスクを併用して、2 つの IBM i システムの間でネイティブ・ライブラリー・オブジェクトを転送します。このサンプルは、Managed File Transfer `filecopy` タスクで `<presrc>` ネスト・エレメントを使用して、ソース・エージェント・システム上の要求されたライブラリーを一時保存ファイルに保存する実行可能スクリプト `librarysave.sh` を呼び出します。保存ファイルは、`filecopy` Ant タスクによって宛先エージェント・システムに移動されます。宛先エージェント・システムでは、`<postdst>` ネスト・エレメントを使用して実行可能スクリプト `libraryrestore.sh` を呼び出し、保存ファイルに保存されているライブラリーを宛先システムに復元します。

IBM i

このサンプルを実行する前に、`librarytransfer.xml` ファイルに記述されているいくつかの構成を完了しておく必要があります。また、2 つの IBM i マシン上に Managed File Transfer 作業環境がなければなりません。セットアップには、1 つ目の IBM i マシンで実行されるソース・エージェントと、2 つ目の IBM i マシンで実行される宛先エージェントを含める必要があります。2 つのエージェントは相互通信が可能でなければなりません。

IBM i

`librarytransfer` サンプルは、次の 3 つのファイルで構成されます。

- `librarytransfer.xml`
- `librarysave.sh` (`<presrc>` 実行可能スクリプト)

- `libraryrestore.sh` (<postdst> 実行可能スクリプト)

サンプル・ファイルは、`/QIBM/ProdData/WMQFTE/V7/samples/fteant/ibmi/librarytransfer` ディレクトリーにあります。

IBM i ユーザーは、このサンプルを実行するために、以下のステップを実行する必要があります。

1. Qshell セッションを開始します。IBM i コマンド・ウィンドウで、`STRQSH` と入力します。
2. 次のようにして、ディレクトリーを `bin` ディレクトリーに変更します。

```
cd /QIBM/ProdData/WMQFTE/V7/bin
```

3. 必要な構成を行った後、次のコマンドを使用してサンプルを実行します。

```
fteant -f /QIBM/ProdData/WMQFTE/V7/samples/fteant/ibmi/librarytransfer/librarytransfer.xml
```

physicalfiletransfer (IBM i プラットフォームのみ)

IBM i `physicalfiletransfer` サンプルは、Ant タスクを使用して 1 つの IBM i システムのライブラリーから別の IBM i システムのライブラリーにソース物理ファイルまたはデータベース・ファイルを転送する方法を示しています。

IBM i IBM i の IBM WebSphere MQ File Transfer Edition 7.0.2 には、ネイティブ IBM i 上のソース物理ファイルまたはデータベース・ファイルの転送のための直接的なサポートが組み込まれていません。`physicalfiletransfer` サンプルは、IBM i のネイティブの保存ファイル・サポートと、Managed File Transfer に付属する事前定義の Ant タスクを併用して、2 つの IBM i システムの間で完全なソース物理ファイルおよびデータベース・ファイルを転送します。このサンプルでは、Managed File Transfer `filecopy` タスク内で <presrc> ネスト・エレメントを使用して、実行可能スクリプト `physicalfilesave.sh` を呼び出し、ソース・エージェント・システム上のライブラリーから要求されたソース物理ファイルまたはデータベース・ファイルを一時保存ファイルに保存します。保存ファイルは、`filecopy` Ant タスクによって宛先エージェント・システムに移動されます。宛先エージェント・システムでは、<postdst> ネストされたエレメントを使用して実行可能スクリプト `physicalfilerestore.sh` を呼び出し、保存ファイル内のファイル・オブジェクトを宛先システム上の指定されたライブラリーに復元します。

IBM i このサンプルを実行する前に、`physicalfiletransfer.xml` ファイルに記述されているいくつかの構成を完了しておく必要があります。また、2 つの IBM i システム上に Managed File Transfer 作業環境がなければなりません。セットアップには、1 つ目の IBM i システムで実行されるソース・エージェントと、2 つ目の IBM i システムで実行される宛先エージェントを含める必要があります。2 つのエージェントは相互通信が可能でなければなりません。

IBM i `physicalfiletransfer` サンプルは、次の 3 つのファイルで構成されます。

- `physicalfiletransfer.xml`
- `physicalfilesave.sh` (<presrc> 実行可能スクリプト)
- `physicalfilerestore.sh` (<postdst> 実行可能スクリプト)

サンプル・ファイルは、`/QIBM/ProdData/WMQFTE/V7/samples/fteant/ibmi/physicalfiletransfer` ディレクトリーにあります。

IBM i ユーザーは、このサンプルを実行するために、以下のステップを実行する必要があります。

1. Qshell セッションを開始します。IBM i コマンド・ウィンドウで、`STRQSH` と入力します。
2. 次のようにして、ディレクトリーを `bin` ディレクトリーに変更します。

```
cd /QIBM/ProdData/WMQFTE/V7/bin
```

3. 必要な構成を行った後、次のコマンドを使用してサンプルを実行します。

```
fteant -f /QIBM/ProdData/WMQFTE/V7/samples/fteant/ibmi/physicalfiletransfer/physicalfiletransfer.xml
```

timeout

timeout サンプルは、ファイル転送を試行し、指定したタイムアウト値より長くかかる場合は転送を取り消す Ant タスクの使用例を示しています。スクリプトは、Managed File Transfer [filecopy](#) タスクを使用してファイル転送を開始します。この転送の結果は保留になります。スクリプトは、Managed File Transfer 848 ページの『[fte:awaitoutcome](#) の Ant タスク』タスクを使用して、指定された秒数の間、転送の完了を待機します。指定された時間内に転送が完了しないと、Managed File Transfer 852 ページの『[fte:cancel](#) の Ant タスク』タスクがファイル転送を取り消すために使用されます。

vsamtransfer

z/OS

z/OS vsamtransfer サンプルは、Managed File Transfer を使用して、ある VSAM データ・セットを別の VSAM データ・セットに転送する Ant タスクの使用例を示しています。Managed File Transfer は、現在、VSAM データ・セットの転送をサポートしていません。サンプル・スクリプトは、[presrc](#) というネストされたエレメントを使用して、VSAM データ・レコードを順次データ・セットにアンロードし、実行可能ファイル `datasetcopy.sh` を呼び出します。スクリプトは、Managed File Transfer 856 ページの『[fte:filemove](#) の Ant タスク』タスクを使用して、ソース・エージェントから宛先エージェントに順次データ・セットを転送します。次にスクリプトは、[postdst](#) というネストされたエレメントを使用して、`loadvsam.jcl` スクリプトを呼び出します。この JCL (ジョブ制御言語) スクリプトは、転送されたデータ・セット・レコードを宛先の VSAM データ・セットにロードします。このサンプルでは、宛先呼び出しに JCL を使用してこの言語オプションを説明します。代わりに、2 番目のシェル・スクリプトを使用しても、同じ結果を得ることができます。

z/OS

このサンプルでは、ソースおよび宛先データ・セットは VSAM である必要はありません。ソースと宛先のデータ・セットが同じタイプであれば、サンプルはどのデータ・セットでも機能します。

z/OS

このサンプルが正しく機能するには、`datasetcopy.sh` スクリプトをソース・エージェントのコマンド・パスに置き、`loadvsam.jcl` スクリプトを宛先エージェントのコマンド・パスに置く必要があります。エージェントのコマンド・パスの設定について詳しくは、『[commandPath](#)』を参照してください。

zip

zip サンプルは、`zip.xml` と `zipfiles.xml` の 2 つのスクリプトで構成されています。このサンプルでは、Managed File Transfer 856 ページの『[fte:filemove](#) の Ant タスク』タスク内の [presrc](#) というネストされたエレメントを使用して、ファイル転送移動操作を実行する前に Ant スクリプトを実行する方法を例示します。`zip.xml` スクリプト内の [presrc](#) というネストされたエレメントによって呼び出される `zipfiles.xml` は、ディレクトリーの内容を圧縮します。`zip.xml` スクリプトはこの圧縮ファイルを転送します。このサンプルでは、`zipfiles.xml` Ant スクリプトがソース・エージェントのコマンド・パス上に存在する必要があります。これは、`zipfiles.xml` Ant スクリプトに、ソース・エージェントでディレクトリーの内容を圧縮するために使用されるターゲットが含まれているためです。エージェントのコマンド・パスの設定について詳しくは、『[commandPath](#)』を参照してください。

ユーザー出口での MFT のカスタマイズ

Managed File Transfer のフィーチャーは、ユーザー出口ルーチンと呼ばれる独自のプログラムを使用してカスタマイズできます。

Managed File Transfer は、ユーザーが作成したプログラム (ユーザー出口ルーチン) に Managed File Transfer が制御を渡すことができる、コード内のポイントを提供します。これらのポイントは、ユーザー

出口ポイント (出口点) と呼ばれます。その後 Managed File Transfer は、作成したプログラムが処理を終了した時に制御を再開できます。ユーザー出口は必ずしも使用する必要はありませんが、特定の要件を満たすために Managed File Transfer システムの機能を拡張およびカスタマイズする場合に便利です。

ファイル転送の処理中にソース・システムでユーザー出口を呼び出せるポイントが2つあり、ファイル転送の処理中に宛先システムでユーザー出口を呼び出せるポイントが2つあります。以下の表は、これらの各ユーザー出口点と、出口点を利用するために実装する必要がある Java インターフェースを要約しています。

表 23. ソース側および宛先側の出口点と Java インターフェースの要約	
出口点	実装する Java インターフェース
ソース側の出口点:	
ファイル転送全体が開始する前	SourceTransferStartExit.java
ファイル転送全体の完了後	SourceTransferEndExit.java
宛先側の出口点:	
ファイル転送全体が開始する前	DestinationTransferStartExit.java
ファイル転送全体の完了後	DestinationTransferEndExit.java

ユーザー出口は次の順序で起動されます。

1. SourceTransferStartExit
2. DestinationTransferStartExit
3. DestinationTransferEndExit
4. SourceTransferEndExit

SourceTransferStartExit および DestinationTransferStartExit 出口で行われた変更内容は、以降の出口への入力として伝搬されます。例えば、SourceTransferStartExit 出口が転送メタデータを変更する場合、変更内容は他の出口への入力転送メタデータに反映されます。

ユーザー出口とプログラム呼び出しは、以下の順序で呼び出されます。

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

注:

1. **宛先の転送終了出口** は、転送が正常に完了したか、部分的に正常に完了した場合にのみ実行されます。
2. **postDestinationCall** は、転送が正常に完了したか、または部分的に正常に完了した場合にのみ実行されます。
3. **SourceTransferEndExits** は、成功した転送、部分的に成功した転送、または失敗した転送に対して実行されます。
4. **postSourceCall** は、以下の場合にのみ呼び出されます。
 - 転送は取り消されませんでした。
 - 成功した結果または部分的に成功した結果があります。
 - 宛先転送後プログラムは正常に実行されました。

ユーザー出口のビルド

ユーザー出口をビルドするインターフェースは、`MQ_INSTALL_DIRECTORY/mqft/lib/com.ibm.wmqfte.exitroutines.api.jar`に含まれています。出口をビルドする際には、この.jarファイルをクラスパスに含める必要があります。出口を実行するには、出口を.jarファイルとして抽出してから、以下のセクションで示すディレクトリーにその.jarファイルを配置します。

ユーザー出口の場所

ユーザー出口ルーチンは、次の2つの候補となる場所に保管できます。

- `exits` ディレクトリー。それぞれの `agent` ディレクトリーの下位に `exits` ディレクトリーがあります。
例: `var\mqm\mqft\config\QM_JUPITER\agents\AGENT1\exits`
- `exitClassPath` プロパティーを設定して代わりに場所を指定できます。`exits` ディレクトリーと、`exitClassPath` で設定したクラスパスの両方に出口クラスがある場合、`exits` ディレクトリーのクラスが優先されます。これは、両方の場所に同じ名前のクラスがある場合は `exits` ディレクトリーのクラスが優先されることを意味します。

ユーザー出口を使用するようにエージェントを構成する

エージェントが呼び出すユーザー出口を指定するために設定可能なエージェント・プロパティーが4つあります。これらのエージェント・プロパティーは、`sourceTransferStartExitClasses`、`sourceTransferEndExitClasses`、`destinationTransferStartExitClasses`、および `destinationTransferEndExitClasses` です。これらのプロパティーの使用方法については、[879 ページの『ユーザー出口用の MFT エージェント・プロパティー』](#)を参照してください。

プロトコル・ブリッジ・エージェントでのユーザー出口の実行

ソース・エージェントが出口を呼び出すと、転送対象のソース・アイテムのリストが出口に渡されます。通常のエージェントの場合、これは完全修飾ファイル名のリストです。ファイルはローカル(またはマウントを介してアクセス可能)であるため、出口はファイルにアクセスして暗号化することができます。

ただし、プロトコル・ブリッジ・エージェントの場合、リスト内の項目は次の形式になります。

```
"<file server identifier>:<fully-qualified file name of the file on the remote file server>"
```

リスト内の項目ごとに、出口は最初にファイル・サーバーに接続し、(FTP や FTPS または SFTP プロトコルを使用して) ファイルをダウンロードし、ローカルで暗号化してから、暗号化されたファイルをファイル・サーバーにアップロードする必要があります。

Connect:Direct ブリッジ・エージェントでのユーザー出口の実行

Connect:Direct ブリッジ・エージェントでユーザー出口を実行することはできません。

MFT のソースと宛先のユーザー出口

ディレクトリー分離文字

ソース・ファイル仕様のディレクトリー分離文字は、`fteCreateTransfer` コマンドまたは IBM MQ Explorer でどのようにディレクトリー分離文字を指定したかに関係なく、常にスラッシュ (/) 文字を使用して表されます。出口を作成するときはそのことを考慮に入れる必要があります。例えば、`c:\a\b.txt` というソース・ファイルが存在することを確認し、`fteCreateTransfer (転送の作成)` コマンドまたは IBM MQ Explorer を使用してこのソース・ファイルを指定した場合、ファイル名は実際には `c:/a/b.txt` として保管されていることに注意してください。したがって、`c:\a\b.txt` という元のストリングを検索しても、一致するものは見つかりません。

ソース側の出口点

ファイル転送全体が開始する前

この出口は、保留中の転送のリスト内の次の項目に転送要求があり、その転送が開始されようとしている時に、ソース・エージェントにより呼び出されます。

この出口点の使用例としては、各段階で外部コマンドを使用してエージェントが読み取り/書き込み権限を持つディレクトリーにファイルを送る、宛先システム上のファイルを名前変更する、などがあります。

次の引数をこの出口に渡します。

- ソース・エージェント名
- 宛先エージェント名
- 環境メタデータ
- 転送メタデータ
- ファイル仕様 (ファイル・メタデータを含む)

この出口から次のデータが返されます。

- 更新された転送メタデータ。項目を追加、変更、および削除できます。
- ファイル仕様の更新されたリスト。ソース・ファイル名と宛先ファイル名のペアで構成されます。項目を追加、変更、および削除できます。
- 転送を続行するかどうかを指定する標識。
- 転送ログに挿入するストリング。

この出口点でユーザー出口コードを呼び出す [SourceTransferStartExit.java](#) インターフェースを実装します。

ファイル転送全体の完了後

この出口は、ファイル転送全体が完了した後にソース・エージェントにより呼び出されます。

この出口点の使用例として、転送が完了したことを通知するための E メールまたは IBM MQ メッセージの送信など、完了タスクの実行があります。

次の引数をこの出口に渡します。

- 転送出口の結果
- ソース・エージェント名
- 宛先エージェント名
- 環境メタデータ
- 転送メタデータ
- ファイル結果

この出口から次のデータが返されます。

- 転送ログに挿入する更新されたストリング。

この出口点でユーザー出口コードを呼び出す [SourceTransferEndExit.java](#) インターフェースを実装します。

宛先側の出口点

ファイル転送全体が開始する前

この出口点の使用例として、宛先での許可の検証があります。

次の引数をこの出口に渡します。

- ソース・エージェント名
- 宛先エージェント名
- 環境メタデータ

- 転送メタデータ
- ファイル仕様

この出口から次のデータが返されます。

- 更新された一連の宛先ファイル名。項目の変更はできますが、追加または削除はできません。
- 転送を続行するかどうかを指定する標識。
- 転送ログに挿入するストリング。

この出口点でユーザー出口コードを呼び出す [DestinationTransferStartExit.java](#) インターフェースを実装します。

ファイル転送全体の完了後

この出口点の使用例として、転送されたファイルを使用するバッチ処理の開始、転送に失敗した場合の E メール送信などがあります。

次の引数をこの出口に渡します。

- 転送出口の結果
- ソース・エージェント名
- 宛先エージェント名
- 環境メタデータ
- 転送メタデータ
- ファイル結果

この出口から次のデータが返されます。

- 転送ログに挿入する更新されたストリング。

この出口点でユーザー出口コードを呼び出す [DestinationTransferEndExit.java](#) インターフェースを実装します。

関連概念

[882 ページの『MFT ユーザー出口の Java インターフェース』](#)

ユーザー出口ルーチンの Java インターフェースに関する参照情報については、このセクションのトピックを参照してください。

関連資料

[263 ページの『MFT ユーザー出口のリモート・デバッグの使用可能化』](#)

ユーザー出口を作成する際、コード中の問題を見つけるために有用なデバッガーを使用したい場合があります。

[264 ページの『MFT ソース転送ユーザー出口のサンプル』](#)

[875 ページの『MFT リソース・モニター・ユーザー出口』](#)

リソース・モニターのユーザー出口を使用して、関連タスクが開始される前に、モニターのトリガー条件が満たされた場合に実行するようカスタム・コードを構成できます。

MFT 転送入出力ユーザー出口の使用

Managed File Transfer 転送入出力ユーザー出口を使用してカスタム・コードを構成し、Managed File Transfer 転送用の基礎ファイル・システムの入出力処理を実行することができます。

MFT 転送では、通常、適宜のファイル・システムと対話するための 1 つの組み込み入出力プロバイダーがエージェントによって選択され、転送が行われます。組み込み入出力プロバイダーは、次のタイプのファイル・システムをサポートします。

- 通常の UNIX タイプおよび Windows タイプのファイル・システム
-  z/OS 順次および区分データ・セット (z/OS のみ)
-  IBM i ネイティブ保存ファイル (IBM i のみ)

- IBM MQ キュー
- リモート FTP および SFTP プロトコル・サーバー (プロトコル・ブリッジ・エージェントのみ)
- リモート Connect:Direct ノード (Connect:Direct ブリッジ・エージェントのみ)

サポートされていないファイル・システムまたはカスタム入出力動作が必要なファイル・システム用に、転送入出力ユーザー出口を作成することができます。

転送入出力ユーザー出口は、ユーザー出口に既存のインフラストラクチャーを使用します。しかし、この転送入出力ユーザー出口が他のユーザー出口と異なるのは、各ファイルの転送処理中に転送入出力ユーザー出口の機能に対して複数回のアクセスが行われるということです。

読み込む入出力出口クラスを指定するには、エージェント・プロパティ `IOExitClasses` (`agent.properties` ファイルにあります) を使用します。各出口クラスは、次のようにコマンドで区切ります。

```
IOExitClasses=testExits.TestExit1,testExits.testExit2
```

転送入出力ユーザー出口の Java インターフェースは、次のとおりです。

IOExit

入出力出口が使用されているかどうかを判別するために使用されるメインの入り口点。このインスタンスによって、`IOExitPath` インスタンスが作成されます。

エージェント・プロパティ `IOExitClasses` には `IOExit` 入出力出口インターフェースを指定するだけで十分です。

IOExitPath

抽象インターフェースを表します。例えば、1つのデータ・コンテナ、または一式のデータ・コンテナを表すワイルドカードなどです。このインターフェースを実装するクラス・インスタンスを作成することはできません。このインターフェースは、パスを検査したり、派生パスをリストすることができます。`IOExitResourcePath` インターフェースと `IOExitWildcardPath` インターフェースは `IOExitPath` を拡張したものです。

IOExitChannel

`IOExitPath` リソースとの間でのデータの読み書きを有効にします。

IOExitRecordChannel

レコード単位 `IOExitPath` リソースの `IOExitChannel` インターフェースを拡張して、`IOExitPath` リソースとの間でレコード単位のデータの読み書きを有効にします。

IOExitLock

共用アクセスまたは排他的アクセスの `IOExitPath` リソースでのロックを表します。

IOExitRecordResourcePath

`IOExitResourcePath` インターフェースを拡張して、レコード単位ファイル (z/OS データ・セットなど) のデータ・コンテナを表せるようにします。このインターフェースを使用して、データの位置を指定したり、読み書き操作のための `IOExitRecordChannel` インスタンスを作成したりすることができます。

IOExitResourcePath

`IOExitPath` インターフェースを拡張して、ファイルやディレクトリーなどのデータ・コンテナを表せるようにします。このインターフェースを使用して、データの位置を指定することができます。このインターフェースでディレクトリーを表す場合は、パスのリストが返されるようにするために `listPaths` メソッドを使用できます。

IOExitWildcardPath

`IOExitPath` インターフェースを拡張して、ワイルドカードを使用したパスを表せるようにします。このインターフェースを使用して、複数の `IOExitResourcePaths` のマッチングが行えます。

IOExitProperties

Managed File Transfer が入出力の特定の側面の IOExitPath を処理する方法を決めるプロパティを指定します。例えば、中間ファイルを使用するかどうか、または転送が再開された場合にリソースを先頭から再読み取りするかどうかを指定します。

IBM i IBM i ユーザー出口のサンプル MFT

Managed File Transfer では、IBM i に固有のサンプル・ユーザー出口がインストール済み環境に用意されています。これらのサンプルは、ディレクトリー `MQMFT_install_dir/samples/ioexit-IBMi` および `MQMFT_install_dir/samples/userexit-IBMi` にあります。

com.ibm.wmqfte.exit.io.ibm.qdls.FTEQDLSExit

`com.ibm.wmqfte.exit.io.ibm.qdls.FTEQDLSExit` サンプル・ユーザー出口は、IBM i の QDLS ファイル・システム内のファイルを転送します。この出口をインストールすると、/QDLS で始まるファイルへの転送でこの出口が自動的に使用されます。

この出口をインストールするには、以下のステップを実行します。

1. `com.ibm.wmqfte.samples.ibm.ioexits.jar` ファイルを `WMQFTE_install_dir/samples/ioexit-IBMi` ディレクトリーからエージェントの `exits` ディレクトリーにコピーします。
2. `com.ibm.wmqfte.exit.io.ibm.qdls.FTEQDLSExit` を `IOExitClasses` プロパティに追加します。
3. エージェントを再始動します。

com.ibm.wmqfte.exit.user.ibm.FileMemberMonitorExit

`com.ibm.wmqfte.exit.user.ibm.FileMemberMonitorExit` サンプル・ユーザー出口は、MFT ファイル・モニターのように動作し、IBM i ライブラリーから物理ファイル・メンバーを自動的に転送します。

この出口を実行するには、(例えば `-md` パラメーターを使用して) 「`library.qsys.monitor`」メタデータ・フィールドに値を指定します。このパラメーターでは、ファイル・メンバーへの IFS スタイルのパスが使用され、ファイルおよびメンバーのワイルドカードを含めることができます。例えば、`/QSYS.LIB/FOO.LIB/BAR.FILE/*.MBR`、`/QSYS.LIB/FOO.LIB/*.FILE/BAR.MBR`、`/QSYS.LIB/FOO.LIB/*.FILE/*.MBR` のようになります。

また、このサンプル出口には、オプションのメタデータ・フィールド 「`naming.scheme.qsys.monitor`」もあり、転送中に使用される命名体系の判別に使用できます。デフォルトでは、このフィールドは 「`unix`」 に設定され、宛先ファイルが `FOO.MBR` という名前になります。また、値 「`ibmi`」 を指定して IBM i FTP FILE.MEMBER 体系を使用することもできます (例えば、`/QSYS.LIB/FOO.LIB/BAR.FILE/BAZ.MBR` は `BAR.BAZ` として転送されます)。

この出口をインストールするには、以下のステップを実行します。

1. `com.ibm.wmqfte.samples.ibm.userexits.jar` ファイルを `WMQFTE_install_dir/samples/userexit-IBMi` ディレクトリーからエージェントの `exits` ディレクトリーにコピーします。
2. `com.ibm.wmqfte.exit.user.ibm.FileMemberMonitorExit` を `agent.properties` ファイルの `sourceTransferStartExitClasses` プロパティに追加します。
3. エージェントを再始動します。

com.ibm.wmqfte.exit.user.ibm.EmptyFileDeleteExit

`com.ibm.wmqfte.exit.user.ibm.EmptyFileDeleteExit` サンプル・ユーザー出口は、ソース・ファイル・メンバーが転送の一環として削除される時に空のファイル・オブジェクトを削除します。IBM i ファイル・オブジェクトは多くのメンバーを保持している可能性があるため、MFT ではファイル・オブジェクトはディレクトリーのように扱われます。したがって、MFT を使用してファイル・オブジェクトに移動操作を実行することはできません。移動操作は、メンバー・レベルのみでサポートされます。その結果、メンバーに対して移動操作を行うと、空のファイルが残されます。これらの空のファイルを転送要求の一環として削除する場合に、このサンプル出口を使用します。

「empty.file.delete」メタデータに「true」を指定して FTEFileMember を転送すると、このサンプル出口は、親ファイルが空の場合にはその親ファイルを削除します。

この出口をインストールするには、以下のステップを実行します。

1. com.ibm.wmqfte.samples.ibm.userexits.jar ファイルを `WMQFTE_install_dir/samples/userexit-IBMi` からエージェントの `exits` ディレクトリーにコピーします。
2. com.ibm.wmqfte.exit.user.ibm.EmptyFileDeleteExit を `agent.properties` ファイルの `sourceTransferStartExitClasses` プロパティーに追加します。
3. エージェントを再始動します。

関連資料

260 ページの『MFT 転送入出力ユーザー出口の使用』

Managed File Transfer 転送入出力ユーザー出口を使用してカスタム・コードを構成し、Managed File Transfer 転送用の基礎ファイル・システムの入出力処理を実行することができます。

879 ページの『ユーザー出口用の MFT エージェント・プロパティー』

`agent.properties` ファイルの標準プロパティーに加え、ユーザー出口ルーチン用に特別に用意されたいくつかの拡張プロパティーがあります。これらのプロパティーはデフォルトでは組み込まれないため、これらのいずれかを使用する場合は、`agent.properties` ファイルを手動で編集する必要があります。`agent.properties` ファイルをそのエージェントの実行中に変更した場合、エージェントを停止および再開してその変更が反映されるようにしてください。

MFT ユーザー出口のリモート・デバッグの使用可能化

ユーザー出口を作成する際、コード中の問題を見つけるために有用なデバッガーを使用したい場合があります。

出口は、エージェントを実行する Java 仮想マシン内部で実行されるため、統合開発環境に通常組み込まれている直接的なデバッグ・サポートを使用することができません。しかし、JVM のリモート・デバッグを使用可能にして、適切なりモート・デバッガーに接続することができます。

リモート・デバッグを使用可能にするには、標準の JVM パラメーター `-Xdebug` および `-Xrunjdwp` を使用します。これらのプロパティーは、`BFG_JVM_PROPERTIES` 環境変数によってエージェントを実行する JVM に渡されます。例えば、UNIX では、以下のコマンドによって、エージェントを開始し、JVM が TCP ポート 8765 でデバッガー接続を `listen` するようにします。

```
export BFG_JVM_PROPERTIES="-Xdebug -Xrunjdwp:transport=dt_socket,server=y,address=8765"
fteStartAgent -F TEST_AGENT
```

エージェントは、デバッガーが接続するまで開始しません。Windows の場合、`export` コマンドの代わりに `set` コマンドを使用します。

また、デバッガーと JVM の間で他の通信方式を使用することもできます。例えば、JVM に接続するのではなく、逆に JVM がデバッガーへの接続を開始することもできます。また、TCP の代わりに、共有メモリーを使用することもできます。詳しくは、[Java Platform Debugger Architecture](#) 資料を参照してください。

エージェントをリモート・デバッグ・モードで開始する際は、`-F` (フォアグラウンド) パラメーターを使用する必要があります。

Eclipse デバッガーの使用

以下のステップは、Eclipse 開発環境のリモート・デバッグ機能に適用されます。また、JPDA 準拠の他のリモート・デバッガーを使用することもできます。

1. 「実行」 > 「デバッグ・ダイアログを開く」 (Eclipse のバージョンによっては「実行」 > 「デバッグの構成」または「実行」 > 「デバッグ・ダイアログ」)
2. 構成タイプのリストにある「リモート Java アプリケーション」をダブルクリックして、デバッグ構成を作成します。

3. 構成フィールドに入力し、デバッグ構成を保存します。エージェントの JVM をデバッグ・モードで既に開始している場合は、すぐに JVM に接続できます。

MFT ソース転送ユーザー出口のサンプル

```
/*
 * 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";
    }
}
```

```
}
```

プロトコル・ブリッジ資格情報ユーザー出口のサンプル

このユーザー出口のサンプルの使用方法については、[225 ページの『出口クラスを使用したファイル・サーバーの資格情報のマップ』](#)を参照してください。

```
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
    }
}

```

プロトコル・ブリッジ・プロパティ・ユーザー出口のサンプル

このユーザー出口のサンプルの使用法については、[219 ページの『ProtocolBridgePropertiesExit2: プロトコル・ファイル・サーバー・プロパティの検索』](#)を参照してください。

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;
}
}

```

エージェント・コマンド・キューにメッセージを PUT することによる MFT の制御

エージェント・コマンド・キューにメッセージを PUT することによって、Managed File Transfer を制御するアプリケーションを作成することができます。

エージェントのコマンド・キューにメッセージを PUT し、エージェントが次のいずれかの操作を実行するように要求できます。

- ファイル転送の作成
- スケジュール済みファイル転送の作成
- ファイル転送を取り消す
- スケジュール済みファイル転送の取り消し
- コマンドの呼び出し
- モニターの作成
- モニターの削除
- ping を返し、エージェントがアクティブであることを示す

これらのいずれかの操作をエージェントが実行するよう要求するには、メッセージが次のいずれかのスキーマに準拠した XML 形式でなければなりません。

FileTransfer.xsd

この形式のメッセージは、ファイル転送またはスケジュール済みファイル転送の作成、コマンドの呼び出し、およびファイル転送またはスケジュール済みファイル転送の取り消しに使用できます。詳細内容は [を参照してください](#)。

Monitor.xsd

この形式のメッセージはリソース・モニターの作成または削除に使用できます。詳細内容は [を参照してください](#)。

PingAgent.xsd

この形式のメッセージは、エージェントがアクティブであることを検査するよう、エージェントを ping するのに使用できます。詳細内容は [を参照してください](#)。

エージェントは要求メッセージに対して応答を返します。応答メッセージは、要求メッセージに定義されている応答キューに PUT されます。応答メッセージは、次のスキーマで定義された XML 形式です。

Reply.xsd

詳細内容は [を参照してください](#)。

トラブルシューティング MFT

以下の参照情報を、Managed File Transfer のエラーの診断に役立ててください。

MFT の一般的なトラブルシューティング

以下の参照情報を、Managed File Transfer のエラーの診断に役立ててください。

関連概念

[272 ページの『MFT の使用のヒント』](#)

Managed File Transfer を最大限に活用するために役立ついくつかの提案を以下に示します。

[325 ページの『MFT エージェントまたはロガーを Windows サービスとして実行するためのガイダンス』](#)
Managed File Transfer エージェント、スタンドアロン・データベース・ロガー、およびスタンドアロン・ファイル・ロガーを Windows サービスとして実行することができます。これらの Windows サービスに問題がある場合、サービス・ログ・ファイルおよびこのトピックの情報をを使用して、問題を診断することができます。

関連タスク

[273 ページの『MFT でのトレースの実行』](#)

このセクションで説明する方法を使用して、Managed File Transfer をトレースできます。

[330 ページの『Oracle データベースで MFT データベース・スキーマを更新する時にエラーが発生した場合の対応策』](#)

ftelog_tables_oracle_702_703.sql ファイルを使用してデータベース・スキーマを最新レベルに更新すると、ERROR at line 1: ORA-02289: sequence does not exist というエラー・メッセージが表示される場合があります。このエラーは、表で使用されるシーケンスおよびトリガーのスキーマが表のスキーマと異なるために発生します。

関連資料

[299 ページの『共通 MFT 問題』](#)

Managed File Transfer ネットワーク内で起きる可能性がある共通の問題。

[302 ページの『MFT エージェントが fteListAgents コマンドによってリストされない場合に行う事柄』](#)

エージェントが **fteListAgents** コマンドによってリストされないか、IBM MQ Explorer に表示されない場合、または IBM MQ Explorer の「**転送ログ**」にファイル転送が表示されない場合は、いくつかの問題判別ステップを実行して原因を調査することができます。

[304 ページの『MFT エージェント・プロセスがなくなっているのに診断情報がログに記録されていない場合』](#)

UNIX プラットフォームで、エージェント・プロセスがなくなっているのにエージェント・ログ・ファイルにその説明が一切記録されていない場合、エージェントの開始方法がその原因になっている可能性があります。

[314 ページの『ファイル転送が停止したと思われる場合の対処法』](#)

負荷の重いシステムであったり、ソース・エージェントと宛先エージェントとの間でネットワーク問題があったりする場合には、転送が待機状態またはリカバリー状態で停止しているように見えることがあります。この原因となるいくつかの要因があります。

[315 ページの『プロトコル・ブリッジ・エージェントがファイル未検出を報告する場合の対処法』](#)

プロトコル・ブリッジの接続先である SFTP サーバーまたは FTP サーバーが File not found エラー・メッセージを返したことをプロトコル・ブリッジ・エージェントが報告した場合、このメッセージは、いくつかの異なるエラー・ケースのうちの 1 つが発生したことを意味する可能性があります。

[317 ページの『キュー・リソース・モニターが開始した転送によって作成された宛先ファイルに間違ったデータが含まれる場合の対処法』](#)

リソース・モニターを作成して、キューをモニターしたり、キュー上のメッセージまたはメッセージのグループをファイルに転送したりすることができます。ファイル名は、メッセージまたはグループの最初のメッセージ上で MQMD メッセージ記述子を使用して指定することができます。メッセージからファイルへの転送が失敗し、メッセージまたはグループがキューに残されると、モニターの次の起動時に、間違ったデータが組み込まれたファイルが作成される可能性があります。

[319 ページの『SYSTEM.MANAGED.DURABLE キューにメッセージが蓄積されたり、ファイル・システムが満杯になったりする場合の対策』](#)

IBM MQ Explorer プラグインが調整キュー・マネージャーで永続サブスクリプションを使用すると、SYSTEM.MANAGED.DURABLE キューにメッセージが蓄積されていく可能性があります。大容量の Managed File Transfer ネットワークを使用している場合、IBM MQ Explorer プラグインを頻繁に使用するか、または両方を使用すると、このメッセージ・データがローカル・ファイル・システムに入力される可能性があります。

[319 ページの『パブリッシュ前のメッセージの検査』](#)

エージェントは IBM WebSphere MQ 6.0 キュー・マネージャーに接続できるため、エージェントは IBM WebSphere MQ 7.0 で導入された直接パブリケーション方式を使用しません。代わりに、エージェントは、MQRFH ヘッダーが含まれる通常のメッセージを調整キュー・マネージャーに送信します。MQRFH ヘッダーは、メッセージのペイロードをパブリッシュするように要求します。これらのメッセージは調整キュー

ー・マネージャー上の SYSTEM.FTE キューに送信され、通常、メッセージはそのキューから即座にパブリッシュされます。エラー条件がこのパブリッシュを停止する場合、パブリッシュが試行される前にキュー上のメッセージを調べて、診断に役立ててください。以下のステップを実行して、診断を行うことができます。

321 ページの『IBM i 保存ファイルの転送中に起きる可能性のあるエラー』

Managed File Transfer を使用して同じ IBM i 保存ファイルを複数回転送すると、転送が失敗することがあります。

322 ページの『メッセージ・サイズに関連する MQ 属性および MFT プロパティを設定する際のガイダンス』

さまざまなサイズのメッセージを読み書きする際の Managed File Transfer の動作に影響を与えるように、IBM MQ 属性および Managed File Transfer プロパティを変更することができます。

331 ページの『MFT ロガーのエラー処理とリジェクト』

Managed File Transfer ロガーは、メッセージごとのエラーと一般エラーの 2 つのタイプのエラーを特定します。

332 ページの『MFT ロガーが開始されたのに転送情報がログとしてデータベースに記録されていない場合の対応策』

Managed File Transfer ロガーが使用するデータベース表では、データベースのページ・サイズが 8 KB 以上でなければなりません。データベースのページ・サイズが十分な大きさでない場合、表が正しく作成されず、エラー SQLSTATE=42704 が表示されます。

471 ページの『fteDisplayVersion』

fteDisplayVersion コマンドを使用して、インストールした Managed File Transfer のバージョンを表示します。

336 ページの『BFGSS0023E エラーとその回避方法』

以前のバージョンの製品に戻すためにインストール済み環境からフィックスパックをアンインストールする場合に、インストール済み環境に関連しているエージェントがアンインストールの実行時に管理対象転送に関係していると、そのエージェントは開始できず、BFGSS0023E エラーが報告されます。このエラーは、エージェントの再始動時に BFGSS0023E メッセージが表示されないようにするためのいくつかの手順を実行することで回避できます。

338 ページの『MFT の戻りコード』

Managed File Transfer コマンド、Ant タスク、およびログ・メッセージは、機能が正常に実行されたかどうかを示す戻りコードを提供します。

MFT の使用のヒント

Managed File Transfer を最大限に活用するために役立ついくつかの提案を以下に示します。

- **agent.properties** ファイルを変更した場合は、エージェントを停止および再開してその変更が反映されるようにしてください。
- ファイル転送を開始しても、転送が進行している兆候がなく、エラーも報告されない場合は、ソース・エージェントが実行されているか確認してください。転送が表示されるにもかかわらず進行していない場合は、宛先エージェントが実行されているかどうかを確認してください。エージェントの現在の状態は、エージェントのログで調べることができます。または、**ftePingAgent** コマンドを使用してエージェントがアクティブかどうかを確認することもできます。
- **fteCancelTransfer** コマンドを使用して個々の転送を取り消す際には、**-agentName** パラメーターに、ソース・エージェントまたは宛先エージェントを使用します。ただし、**fteDeleteScheduledTransfer** コマンドを使用して転送スケジュールを削除する際には、**-agentName** パラメーターにはソース・エージェント名を使用しなければなりません。
- ファイル転送を作成する際、ソース・ファイルのパスと宛先ファイルのパスは（絶対パスまたは相対パスのどちらの場合も）、ソース・エージェントおよび宛先エージェントに対してのみ意味を持ちます。**fteCreateAgent** コマンドの発行元のシステムおよびディレクトリーには、転送されるファイルとの関連性はありません。

- デフォルトの環境セットアップでは、特に複数の転送を同時に実行する場合には、Managed File Transfer を完全にサポートできないことがあります。 エージェントでメモリー不足であることを示すエラーが発生した場合には、必要に応じて次のパラメーターを確認し、更新してください。

- **UNIX** UNIX プラットフォームでは、`ulimit -m 1048576` (約 1 GB) コマンドを実行します。この最大常駐設定サイズは、最大 25 件までの同時転送に十分対応できます (25 件の同時転送は、エージェントのデフォルトの最大転送数です)。
- すべてのプラットフォームについて、**BFG_JVM_PROPERTIES** 環境変数を次のように設定します。
`BFG_JVM_PROPERTIES="-Xmx1024M"`

デフォルトの最大数である 25 を上回る同時転送を可能にするには、**ulimit** および **BFG_JVM_PROPERTIES** を、推奨されるより大きいサイズにしてください。

注: Connect:Direct ブリッジ・エージェントの場合、同時転送のデフォルトの最大数は 5 です。

- Managed File Transfer を使用して、異なるプラットフォーム間でテキスト・モードによるファイル転送を行う場合、ソース・プラットフォームのデフォルト・ファイル・エンコード方式が宛先プラットフォームによってサポートされない場合があります。これにより、転送は失敗し、次のエラーが出されます。

```
BFGI00058E: The transfer source encoding xxx is illegal or for an unsupported character set.
```

このエラーを解決するには、環境変数を使用して、ソース・エンコード方式を宛先プラットフォームがサポートするものに設定します。ソース・システムで **BFG_JVM_PROPERTIES** システム環境変数を次のように設定します。`BFG_JVM_PROPERTIES="-Dfile.encoding=xxx"`。ここで、`xxx` は宛先プラットフォームがサポートするエンコード方式です。例えば、Solaris プラットフォームから別のプラットフォームにテキスト・モードでファイルを転送していて、ソース・ロケールが「ja」に設定されている場合、**BFG_JVM_PROPERTIES** を `BFG_JVM_PROPERTIES="-Dfile.encoding=EUC-JP"` のように設定します。ソース・ロケールが「ja_JP.PCK」に設定されている場合、**BFG_JVM_PROPERTIES** を次のように設定します。`BFG_JVM_PROPERTIES="-Dfile.encoding=Shift_JIS"`。

個々の転送に対するこのエラーは、新しい転送を始める際に **-sce** パラメーターを使用することによって解決することもできます。詳しくは、437 ページの『[fteCreateTransfer: 新規ファイル転送の開始](#)』を参照してください。

関連資料

605 ページの『[MFT Java システム・プロパティ](#)』

いくつかの Managed File Transfer コマンド・プロパティとエージェント・プロパティは Java システム・プロパティとして定義する必要があります。これは、コマンド・プロパティおよびエージェント・プロパティのメカニズムを使用できない初期の関数用の構成を定義するためです。

MFT でのトレースの実行

このセクションで説明する方法を使用して、Managed File Transfer をトレースできます。

このタスクについて

さまざまな Managed File Transfer リソースのトレース方法について詳しくは、277 ページの『[Multiplatforms での Managed File Transfer リソースのトレース](#)』および 284 ページの『[Managed File Transfer for z/OS リソースのトレース](#)』を参照してください。

fteSetAgentTraceLevel

fteSetAgentTraceLevel1 コマンドを使用して、エージェントの現在のトレース・レベルを動的に変更することができます。

目的

エージェント・トレースのオンとオフを切り替えるとき、または設定されているエージェント・トレースのレベルを変更するとき、このコマンドを使用します。 **fteSetAgentTraceLevel1** コマンドを使用すると、トレース・レベルを変更するために、エージェントをシャットダウンして再始動する必要はありません。生成されるトレース・ファイルは、`MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/`

agents/agent_name/logs/trace%PID%/trace%PID%.txt (ここで、%PID% はエージェント・インスタンスのプロセス ID) に置かれます。



重要:

Multi マルチプラットフォームで IBM WebSphere MQ 7.5 以降を使用する場合、エージェント・プロセスを実行しているユーザーのみが **fteSetAgentTraceLevel** コマンドを実行できます。

z/OS **V 9.0.2** z/OS の場合、Continuous Delivery では、IBM MQ 9.0.2 以降、以下のいずれかの方法で **fteSetAgentTraceLevel** コマンドを実行できます。

- エージェント・プロセスを実行しているものと同じユーザー ID。
- エージェント・プロパティ **adminGroup** で指定されたグループのメンバー。

V 9.0.0.1 これは、IBM MQ 9.0.0 Fix Pack 1 以降の Long Term Support リリースにも適用されません。

詳しくは、**adminGroup** プロパティ (550 ページの『MFT agent.properties ファイル』) を参照してください。

IBM WebSphere MQ 7.5 以降では、**fteSetAgentTraceLevel** コマンドは、エージェント・プロセス・コントローラーのトレースの書き込みも行います。生成されるトレース・ファイルは、MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/pctrace%PID%/pctrace%PID%.txt (ここで、%PID% はエージェント・インスタンスのプロセス ID) に置かれます。

このコマンドを使用して、エージェント・プロセスで Javacore を生成することもできます。エージェントによって、MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name ディレクトリ内に Javacore ファイルが生成されます。

トレースを実行すると、パフォーマンスに重大な影響を及ぼすおそれがあり、また大量のトレース・データが生成される可能性もあるため、トレースの実行は注意して、必要な場合にのみ実施してください。通常は、IBM サービス担当員からトレースを依頼された場合に限り、トレースを有効にします。



重要:

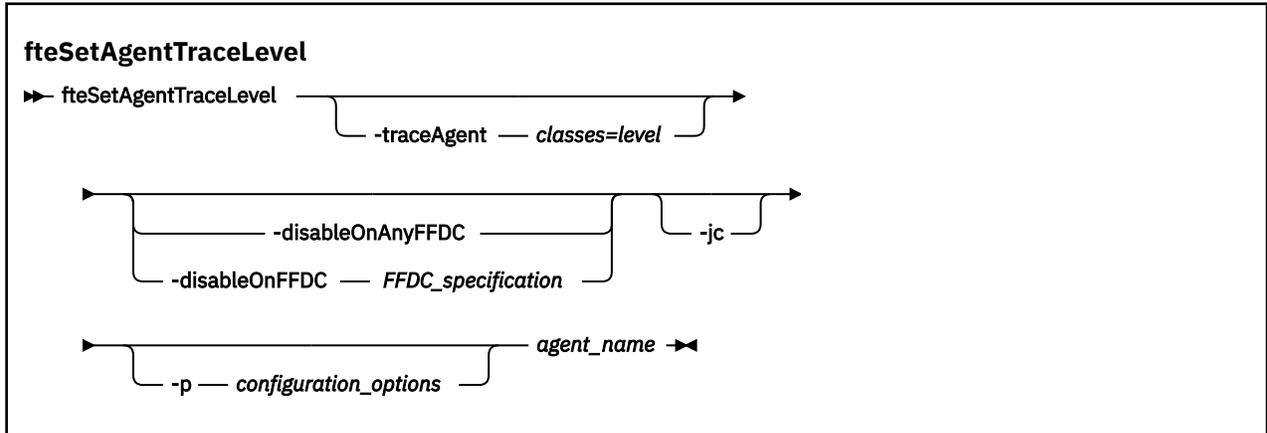
1. このコマンドは、エージェントが実行されているシステムで実行する必要があります。
2. トレースとロギングは、エージェントの再始動後は持続しません。

エージェントが終了し、プロセス・コントローラー・プロセスによって再始動された場合、必要なトレース・プロパティとログ・プロパティを含むように agent.properties ファイルが更新されるまで、動的トレースとログは有効になりません。

トレース・ファイルのサイズや保持するトレース・ファイルの数などの他のトレース・プロパティを agent.properties ファイルに設定することができます。これらのプロパティについては、『[拡張エージェント・プロパティ](#)』で説明されています。

オプションの -p パラメーターは、デフォルト・セットと異なる構成オプションのセットを使用する場合にのみこのコマンドに指定します。詳しくは、『[550 ページの『MFT agent.properties ファイル』](#)』を参照してください。

構文



Parameters

-traceAgent *classes=level*

必須。 エージェント・トレースを設定するレベルおよびトレースの適用対象クラス。

V 9.0.5 IBM MQ 9.0.5 以降では、コロンで区切られたクラス仕様のリストを指定できます。このオプションを使用すると、さまざまなクラスを異なるレベルでトレースするように設定できます。以下に例を示します。

```
fteSetAgentTraceLevel -traceAgent com.ibm.wmqfte.agent=all:com.ibm.wmqfte.cmdhandler=moderate  
AGENT1
```

このレベルのトレースを適用する対象となる、コンマで区切られたクラス仕様のリストは引き続き指定できます。このパラメーターが指定されない場合、トレース・レベルはすべてのエージェント・クラスに適用されます。次の形式を使用します。

```
classes=level
```

以下に例を示します。

```
com.ibm.wmqfte=all
```

classes を Managed File Transfer パッケージ名に置換すると、特定のパッケージのみをトレースできます。ただし、このオプションでキャプチャーされるのはエージェントの動作のサブセットのみであるため、通常はパッケージのフィルタリングを使用しないことをお勧めします。

(*classes*) が正符号 (+) で始まる場合、現在トレースされている既存のトレース・クラスすべてに、正符号に続くトレース・クラスのリストが追加されます。

有効なトレース・レベル・オプションは以下のとおりです。トレース・ファイルのサイズと詳細度の昇順にリストしています。

off

エージェント・トレースをオフに切り替えますが、ログ・ファイルへの情報の書き込みは続行されます。これはデフォルト・オプションです。

flow

エージェント内でプロセッシング・フローに関連付けられたトレース・ポイントのデータを収集します。

moderate

トレースであまり多くない量の診断情報を収集します。

詳細

トレースで詳細な診断情報を収集します。

all

すべてのエージェント・クラスでエージェント・トレースを実行するように設定します。
エージェントの完全トレースを開始するには、次のコマンドを実行します。

```
fteSetAgentTraceLevel -traceAgent =all AGENT_NAME
```

エージェントの完全トレースを停止するには、次のコマンドを実行します。

```
fteSetAgentTraceLevel -traceAgent =off AGENT_NAME
```

-disableOnAnyFFDC

オプション。このパラメーターを指定すると、First Failure Data Capture (FFDC) ファイルを生成するときに、エージェントでトレースが無効になります。

-disableOnAnyFFDC および **-disableOnFFDC** パラメーターのいずれか一方のみ指定できます。

-disableOnFFDC FFDC_specification

オプション。このパラメーターを指定すると、*FFDC_specification* に合致する First Failure Data Capture (FFDC) ファイルを生成するときに、エージェントでトレースが無効になります。
FFDC_specification は、値のコンマ区切りリストです。値のフォーマットは以下のいずれかです。

class_name

FFDCが発生したクラスの名前。例えば `com.ibm.wmqfte.classA` です。

class_name:probe_ID

クラスの名前と、FFDCの発生元のクラスにあるロケーションのプローブ ID。例えば `com.ibm.wmqfte.classB:1` です。

-disableOnAnyFFDC および **-disableOnFFDC** パラメーターのいずれか一方のみ指定できます。

-jc

オプション。エージェントが Javacore ファイルを生成するように要求します。IBM サービス・チームが問題診断のために、このパラメーターを指定してコマンドを実行するように求める場合があります。このパラメーターは、**-p** 以外のパラメーターと一緒に使用することはできません。

-p configuration_options

オプション。このパラメーターは、エージェント・トレース・レベルを設定するために使用される構成オプションのセットを決定します。このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティ・ファイルのセットを使用します。

このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。

agent_name

必須。トレース・レベルを設定する Managed File Transfer Agent の名前。

-?または-h

オプション。コマンド構文を表示します。

例

この例では、AGENT1 のすべてのクラスについて、トレース・レベルが `all` に設定されています。

```
fteSetAgentTraceLevel -traceAgent com.ibm.wmqfte=all AGENT1
```

この例では、AGENT1 の `com.ibm.wmqfte.agent.Agent` と `com.ibm.wmqfte.cmdhandler` クラスについて、トレース・レベルが `all` に設定されています。

```
fteSetAgentTraceLevel -traceAgent com.ibm.wmqfte.agent.Agent,com.ibm.wmqfte.cmdhandler=moderate AGENT1
```

以下の例では、**-traceLevel** パラメーターが `off` に設定されているため、サブクラスがトレースから除外されます。 `com.ibm.outer.inner` で始まるクラスを除いて、`com.ibm.outer` で始まるクラスはすべて、`verbose` レベルでトレースされます。

```
fteSetAgentTraceLevel -traceAgent com.ibm.outer=verbose AGENT1  
fteSetAgentTraceLevel -traceAgent +com.ibm.outer.inner=off AGENT1
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

Multi **Multiplatforms** での **Managed File Transfer** リソースのトレース

Managed File Transfer のトレース機能は、お客様の問題や懸念を IBM サポートが診断するための機能です。さまざまなリソースをトレースできます。

このタスクについて

以下を参照してください。

- エージェントのトレース方法については、[277 ページの『Multiplatforms での Managed File Transfer エージェントのトレース』](#)を参照してください。
- コマンドのトレース方法については、[280 ページの『Multiplatforms での Managed File Transfer コマンドのトレース』](#)を参照してください。

Multi **Multiplatforms** での **Managed File Transfer** エージェントのトレース

Managed File Transfer のトレース機能は、お客様の問題や懸念を IBM サポートが診断するための機能です。この機能の動作は、さまざまなコマンドやプロパティで制御できます。

このタスクについて

エージェントに関する問題を調査するためにトレース出力の提供を求められた場合は、エージェントを短時間停止することが可能かどうかに応じて、以下のいずれかのオプションを使用してください。

使用するオプションが不確かな場合は、IBM サポート担当員にお問い合わせください。お客様が直面している問題のトレースを収集する最善の方法をお知らせします。

手順

- エージェントを短時間停止することが可能な場合は、エージェントのトレースを起動時点から収集します。
詳細については [278 ページの『起動時点からの Managed File Transfer エージェント・トレースの収集』](#)を参照してください。
- エージェントを停止できない場合は、**fteSetAgentTraceLevel** コマンドを使用してトレースを動的に収集します。
詳細については [279 ページの『Managed File Transfer エージェント・トレースの動的な収集』](#)を参照してください。

エージェントを短時間停止することが可能な場合は、起動時点から Managed File Transfer エージェント・トレースを収集してください。

始める前に

トレースする必要があるエージェント用のさまざまなプロパティを `agent.properties` ファイルに設定する必要があります。

このタスクについて

起動時点からトレースを収集するには、以下の手順を実行します。

手順

1. トレースする必要があるエージェントの `agent.properties` ファイルを見つけます。

`agent.properties` ファイルは、`MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` ディレクトリー内にあります。

2. ファイルを編集して、以下のプロパティのエントリーを追加します。

- **trace**=トレース仕様

trace プロパティは、トレースする内部クラスおよびパッケージを決定します。IBM サポート担当員から特に指示されない限り、このプロパティには値 `com.ibm.wmqfte=all` を設定してください。

- **traceFiles**=使用するトレース・ファイルの数

- **traceSize**=各トレース・ファイルのサイズ (MB 単位)

traceFiles プロパティと **traceSize** プロパティは、収集するトレース・データ量を制御するために使用します。できる限り多くのトレース・データを収集するために、これらのプロパティには大きい値を設定する必要があります。

例えば、トレース仕様 `com.ibm.wmqfte=all` を使用してラッピング・トレースの 1GB を収集するには、以下の行を `agent.properties` ファイルに追加します。

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

この場合、エージェントは最大 5 つのファイル (各ファイルのサイズは 200 MB) にトレース・データを書き込みます。

これらのエージェント・プロパティについて詳しくは、[MFT agent.properties](#) ファイルを参照してください。

3. **fteStopAgent** コマンドを使用して、トレースする必要があるエージェントを停止します。
4. **fteStartAgent** コマンドを実行してエージェントを開始します。
5. 問題を再現します。
6. エージェントを停止します。
7. エージェントの `agent.properties` ファイルを編集し、ステップ 278 ページの『2』で追加した **トレース**、**トレース・ファイル**、および **トレース・サイズ** のプロパティのエントリーを削除します。これにより、次にエージェントを再始動するときにはトレースは有効にならなくなります。

タスクの結果

結果として生じるトレース・ファイルは、`MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` ディレクトリーに書き込まれます。ここで、`%PID%` はエージェントのプロセス ID です。

fteSetAgentTraceLevel コマンドを使用すると、実行中のエージェントからトレースを収集できます。これは、IBM サポートで停止できないエージェントからのトレースを参照する必要がある場合に非常に役立ちます。

このタスクについて

fteSetAgentTraceLevel コマンドを使用してエージェントからトレースを収集するには、以下の手順を実行します。

手順

1. 次のコマンドを実行して、エージェントのトレースをオンにします。

```
fteSetAgentTraceLevel -traceAgent classes=level agent_name
```

-traceAgent パラメーターは、トレースする内部クラスおよびパッケージを決定します。IBM サポート担当員から特に指示されない限り、このプロパティには値 `com.ibm.wmqfte=all` を設定してください。

2. 問題を再現します。
3. 以下のコマンドを実行して、エージェントのトレースをオフにします。

```
fteSetAgentTraceLevel -traceAgent =off agent_name
```

エージェントがビジー状態の場合、トレース・ファイルがすぐに折り返され、問題の調査に必要な情報が上書きされる可能性があります。その場合は、エージェントを停止するための時間をスケジュールしてから、以下の手順で詳述されているように続行してください。エージェントを短時間停止できない場合は、IBM サポート担当員に連絡して、生成されるトレース・データの量を減らすために使用する代替トレース仕様について相談してください。

4. トレースする必要があるエージェントの `agent.properties` ファイルを見つけます。

`agent.properties` ファイルは、`MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` ディレクトリーにあります。

5. ファイルを編集して、以下のプロパティのエントリーを追加します。

```
traceFiles=number_of_trace_files_to_use  
traceSize=size_of_each_trace_file_in_MB
```

traceFiles プロパティと **traceSize** プロパティは、収集するトレース・データ量を制御するために使用します。

traceFiles プロパティのデフォルト値は 5、**traceSize** プロパティのデフォルト値は 20 MB です。つまり、トレースを動的にオンにした場合、これらのプロパティを設定していなければ、エージェントは 5 つの折り返しトレース・ファイル (各ファイルの最大サイズは 20 MB) にトレース情報を書き込みます。

できる限り多くのトレース・データを収集するために、これらのプロパティには大きい値を設定する必要があります。

例えば、1 GB の折り返しトレースを収集するには、`agent.properties` ファイルに以下の行を追加します。

```
traceFiles=5  
traceSize=200
```

この場合、エージェントは最大 5 つのファイル (各ファイルのサイズは 200 MB) にトレース・データを書き込みます。

これらのエージェント・プロパティについて詳しくは、[MFT agent.properties ファイル](#)を参照してください

6. **fteStopAgent** コマンドを実行して、エージェントを停止します。
7. **fteStartAgent** コマンドを実行してエージェントを開始します。
8. 以下のコマンドを実行して、エージェントのトレースを有効にします。

```
fteSetAgentTraceLevel -traceAgent classes=level agent_name
```

IBM サポート担当員から特に指示されない限り、**-traceAgent** プロパティには値 `com.ibm.wmqfte=all` を設定してください。

9. 問題を再現します。
10. 以下のコマンドを実行して、エージェントのトレースをオフにします。

```
fteSetAgentTraceLevel -traceAgent =off agent_name
```

タスクの結果

生成されたトレース・ファイルが、`MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` ディレクトリーに書き込まれます。ここで、`%PID%` はエージェントのプロセス ID です。

Multi

Multiplatforms での *Managed File Transfer* コマンドのトレース

Managed File Transfer のトレース機能は、お客様の問題や懸念を IBM サポートが診断するための機能です。ユーザーがこの機能を使用してコマンドをトレースすることができます。

このタスクについて



重要: コマンドのトレースでは、コマンドによって実行された処理に関する情報のみが収集されます。エージェントがそのコマンドの処理中に何らかのアクティビティーを実行したとしても、そのアクティビティーはトレースされません。

手順

1. コマンド・プロンプトを表示し、`MQ_INSTALLATION_PATH\bin` ディレクトリーにナビゲートします。
2. コマンドを実行します。

Linux

UNIX

```
./command_name -trace classes=level -tracePath directory_path command_arguments
```

Windows

```
command_name -trace classes=level -tracePath directory_path command_arguments
```

この

- コマンド名は、トレースされるコマンドの名前です。
- `classes=level` は、使用するトレース・レベルと、トレースを有効にするクラスです。IBM サポート担当員から特に指示されない限り、`com.ibm.wmqfte=all` を設定してください。
- ディレクトリー・パスは、トレース・ファイルが書き込まれるディレクトリーです。
- `command_arguments` は、コマンドに渡す必要がある引数です。例えば、**ftePingAgent** コマンドの場合はエージェントの名前です。

タスクの結果

生成されたトレース・ファイルが、**-tracePath** パラメーターで指定したディレクトリーに書き込まれます。

トレース・ファイルは、`trace%PID%.txt.number` という名前になります。各部分の意味は次のとおりです。

- `%PID%` は、コマンドのプロセス ID です。
- の数は、トレース・ファイルのシーケンス番号です。通常、コマンドによって生成されたトレース情報は、シーケンス番号が 0 の単一のトレース・ファイル内に入れられます。

ただし、大量のトレース情報がコマンドによって生成されることがあります。そのような場合、トレースは複数のファイルに書き込まれます。現行のトレース・ファイルのシーケンス番号が 0、次に古いトレース・ファイルのシーケンス番号が 1、という順序になります。

コマンドのトレース出力は、最大 5 つの折り返しトレース・ファイルに書き込まれます。各トレース・ファイルの最大サイズは 20 MB です。

注: コマンドを実行するユーザーに、`-tracePath` パラメーターで指定したディレクトリーへの書き込み権限がない場合、トレース出力は標準エラーに書き込まれます。

例

この例では、`fteListAgents` コマンドがトレースされ、トレースが `C:\trace` ディレクトリーに書き込まれます。

```
fteListAgents -trace com.ibm.wmqfte=all -tracePath C:\trace
```

この例では、`fteCreateTransfer` コマンドがトレースされ、トレースが `/tmp` ディレクトリーに書き込まれます。

```
fteCreateTransfer -trace com.ibm.wmqfte=all -tracePath /tmp -t text -sa AGENT1  
-da AGENT2 -df /import/transferredfile.txt /export/originalfile.txt
```

`/tmp` に書き込まれたトレース・ファイルには、`fteCreateTransfer` コマンドによって実行された処理に関する情報のみが入っています。例えば、エージェントに送信された転送要求メッセージをコマンドがどのように作成したかや、要求を受信したことを示す確認応答がエージェントから送り返されるのをコマンドがどれだけ待機したかなどの情報です。転送自体に関する情報はトレース・ファイルに入っていません。

Multi

Multiplatforms での *Managed File Transfer* スタンドアロン・ロガーのトレース

Managed File Transfer のトレース機能は、お客様の問題や懸念を IBM サポートが診断するための機能です。この機能の動作は、さまざまなコマンドやプロパティーで制御できます。

このタスクについて

ロガーに関する問題を調査するためにトレース出力の提供を求められた場合は、ロガーを短時間停止することが可能かどうかに応じて、以下のいずれかのオプションを使用してください。

使用するオプションが不確かな場合は、IBM サポート担当員にお問い合わせください。お客様が直面している問題のトレースを収集する最善の方法をお知らせします。

手順

- ロガーを短時間停止することが可能な場合は、ロガーのトレースを起動時点から収集します。
[282 ページの『起動時点からの Managed File Transfer スタンドアロン・ロガー・トレースの収集』](#)を参照してください。
- ロガーを停止できない場合は、`fteSetLoggerTraceLevel` コマンドを使用してトレースを動的に収集します。
[283 ページの『Managed File Transfer スタンドアロン・ロガー・トレースの動的な収集』](#)を参照してください。

起動時点からの *Managed File Transfer* スタンドアロン・ロガー・トレースの収集
ロガーを短時間停止することが可能な場合は、起動時点から *Managed File Transfer* ロガー・トレースを収集してください。

始める前に

トレースする必要があるロガーの `logger.properties` ファイルに、いくつかのプロパティを設定する必要があります。

このタスクについて

起動時点からトレースを収集するには、以下の手順を実行します。

手順

1. トレースする必要があるロガーの `logger.properties` ファイルを見つけます。
`logger.properties` ファイルは、`MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` ディレクトリーにあります。
2. ファイルを編集して、以下のプロパティのエントリーを追加します。

- **trace**=トレース仕様

trace プロパティは、トレースする内部クラスおよびパッケージを決定します。IBM サポート担当員から特に指示されない限り、このプロパティには値 `com.ibm.wmqfte=all` を設定してください。

- **traceFiles**=使用するトレース・ファイルの数
- **traceSize**=各トレース・ファイルのサイズ (MB 単位)

traceFiles プロパティと **traceSize** プロパティは、収集するトレース・データ量を制御するために使用します。できる限り多くのトレース・データを収集するために、これらのプロパティには大きい値を設定する必要があります。

例えば、トレース仕様 `com.ibm.wmqfte=all` を使用して、1 GB の折り返しトレースを収集するには、`logger.properties` ファイルに以下の行を追加します。

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

この場合、ロガーは最大 5 つのファイル (各ファイルのサイズは 200 MB) にトレース・データを書き込みます。

これらのロガー・プロパティについて詳しくは、[MFT logger.properties ファイル](#)を参照してください

3. **fteStopLogger** コマンドを使用して、トレースする必要があるロガーを停止します。
4. **fteStartLogger** コマンドを実行してロガーを開始します。
5. 問題を再現します。
6. ロガーを停止します。
7. ロガーの `logger.properties` ファイルを編集し、ステップ 282 ページの『2』で追加した **trace**、**traceFiles**、および **traceSize** の各プロパティの項目を削除します。
これにより、次にロガーを再始動するときにはトレースは有効にならなくなります。

タスクの結果

生成されたトレース・ファイルが、`MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%` ディレクトリーに書き込まれます。ここで、`%PID%` はロガーのプロセス ID です。

fteSetLoggerTraceLevel コマンドを使用すると、実行中のロガーからトレースを収集できます。これは、IBM サポートで停止できないロガーからのトレースを参照する必要がある場合に非常に役立ちます。

このタスクについて

fteSetLoggerTraceLevel コマンドを使用してロガーからトレースを収集するには、以下の手順を実行します。

手順

1. 次のコマンドを実行して、ロガーのトレースをオンにします。

```
fteSetLoggerTraceLevel -traceLogger classes=level logger_name
```

-traceLogger パラメーターは、トレースする内部クラスおよびパッケージを決定します。IBM サポート担当員から特に指示されない限り、このプロパティには値 `com.ibm.wmqfte=all` を設定してください。

2. 問題を再現します。
3. 次のコマンドを使用してトレースをオフにします。

```
fteSetLoggerTraceLevel -traceLogger =off logger_name
```

4. ロガーがビジー状態の場合は、トレース・ファイルがすぐに折り返され、問題を調査するために必要な情報が書き込まれる可能性があります。

ロガーを短時間停止できる場合は、以下のステップを実行して、収集されるトレース・データの量を減らします。それ以外の場合は、IBM サポートに連絡して、収集されるトレース・データの量を減らすための代替トレース仕様について検討してください。

- a) ロガーを停止するための時間を設けます。
- b) トレースする必要があるロガーの `logger.properties` ファイルを見つけます。

`logger.properties` ファイルは、`MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` ディレクトリにあります。

- c) ファイルを編集して、以下のプロパティのエントリーを追加します。

traceFiles= 「使用するトレース・ファイルの数」

traceSize=`size_of_each_trace_file_in_MB`

traceFiles プロパティと **traceSize** プロパティは、収集するトレース・データ量を制御するために使用します。

traceFiles プロパティのデフォルト値は 5、**traceSize** プロパティのデフォルト値は 20 MB です。つまり、トレースを動的にオンにした場合、これらのプロパティを設定していなければ、エージェントは 5 つの折り返しトレース・ファイル (各ファイルの最大サイズは 20 MB) にトレース情報を書き込みます。

できる限り多くのトレース・データを収集するために、これらのプロパティには大きい値を設定する必要があります。

例えば、1 GB の折り返しトレースを収集するには、`logger.properties` ファイルに以下の行を追加します。

```
traceFiles=5
traceSize=200
```

この場合、ロガーは最大 5 つのファイル (各ファイルのサイズは 200 MB) にトレース・データを書き込みます。

これらのロガー・プロパティについて詳しくは、[MFT logger.properties](#) ファイルを参照してください

- d) **fteStopLogger** コマンドを実行して、ロガーを停止します。
- e) **fteStartLogger** コマンドを実行してロガーを開始します。
- f) 以下のコマンドを実行して、ロガーのトレースをオンにします。IBM サポート担当員から特に指示されない限り、**-traceLogger** プロパティには値 `com.ibm.wmqfte=all` を設定してください。

```
fteSetLoggerTraceLevel -traceLogger classes=level logger_name
```

- g) 問題を再現します。
- h) 以下のコマンドを実行して、ロガーのトレースをオフにします。

```
fteSetLoggerTraceLevel -traceLogger =off logger_name
```

タスクの結果

生成されたトレース・ファイルが、`MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%` ディレクトリーに書き込まれます。ここで、`%PID%` はロガーのプロセス ID です。

Managed File Transfer for z/OS リソースのトレース

Managed File Transfer for z/OS のトレース機能は、お客様の問題や懸念を IBM サポートが診断するための機能です。さまざまなリソースをトレースできます。

このタスクについて

以下を参照してください。

- エージェントのトレース方法については、[284 ページの『Managed File Transfer for z/OS エージェントのトレース』](#)を参照してください。
- コマンドのトレース方法については、[290 ページの『Managed File Transfer for z/OS コマンドのトレース』](#)を参照してください。

Managed File Transfer for z/OS エージェントのトレース

Managed File Transfer for z/OS のトレース機能は、お客様の問題や懸念を IBM サポートが診断するための機能です。この機能の動作は、さまざまなコマンドやプロパティで制御できます。

このタスクについて

エージェントに関する問題を調査するためにトレース出力の提供を求められた場合は、以下のいずれかのオプションを使用してください。

使用するオプションが不確かな場合は、IBM サポート担当員にお問い合わせください。お客様が直面している問題のトレースを収集する最善の方法をお知らせします。

手順

- エージェントを短時間停止することが可能な場合は、エージェントのトレースを起動時点から収集します。
詳細については [285 ページの『起動時点からの Managed File Transfer for z/OS エージェント・トレースの収集』](#)を参照してください。
- エージェントを停止できない場合は、**fteSetAgentTraceLevel** コマンドを使用してトレースを動的に収集します。
詳細については [287 ページの『Managed File Transfer for z/OS エージェント・トレースの動的な収集』](#)を参照してください。

z/OS 起動時点からの Managed File Transfer for z/OS エージェント・トレースの収集
エージェントを短時間停止することが可能な場合は、起動時点から IBM MQ Managed File Transfer エージェント・トレースを収集してください。

このタスクについて

トレースを収集する方法は、エージェントの管理に UNIX System Services (USS) と JCL のどちらを使用しているかによって異なります。

以下のオプションのどちらを使用すべきかわからない場合は、IBM サポート担当員にお問い合わせください。お客様が直面している問題にとって最善のトレース収集方法について担当員がアドバイスします。

手順

- z/OS UNIX を使用している場合は、[285 ページの『USS を使用した起動時点からのエージェント・トレースの収集』](#)を参照してください。
- JCL を使用している場合は、[286 ページの『JCL を使用した起動時点からのエージェント・トレースの収集』](#)を参照してください。

z/OS USS を使用した起動時点からのエージェント・トレースの収集

UNIX System Services (USS) を使用して管理している Managed File Transfer for z/OS エージェントのトレースを起動時点から収集するには、エージェントを起動する前にエージェントの `agent.properties` ファイルにいくつかのプロパティを設定する必要があります。

このタスクについて

起動時点からトレースを収集するには、以下の手順を実行します。

手順

1. トレースする必要があるエージェントの `agent.properties` ファイルを見つけます。
`agent.properties` ファイルは、`BFG_DATA/mqft/config/coordination_qmgr_name/agents/agent_name` ディレクトリーにあります。
2. ファイルを編集して、以下のプロパティのエントリーを追加します。
 - **trace**=トレース仕様
trace プロパティは、トレースする内部クラスおよびパッケージを決定します。IBM サポート担当員から特に指示されない限り、このプロパティには値 `com.ibm.wmqfte=all` を設定してください。
 - **traceFiles**=使用するトレース・ファイルの数
 - **traceSize**=各トレース・ファイルのサイズ (MB 単位)
traceFiles プロパティと **traceSize** プロパティは、収集するトレース・データ量を制御するために使用します。できる限り多くのトレース・データを収集するために、これらのプロパティには大きい値を設定する必要があります。

例えば、トレース仕様 `com.ibm.wmqfte=all` を使用して、1 GB の折り返しトレースを収集するには、`agent.properties` ファイルに以下の行を追加します。

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

この場合、エージェントは最大 5 つのファイル (各ファイルのサイズは 200 MB) にトレース・データを書き込みます。

これらのエージェント・プロパティについて詳しくは、[MFT agent.properties ファイル](#)を参照してください。

3. **fteStopAgent** コマンドを使用して、トレースする必要があるエージェントを停止します。
4. **fteStartAgent** コマンドを実行してエージェントを開始します。
5. 問題を再現します。
6. エージェントを停止します。
7. エージェントの **agent.properties** ファイルを編集し、ステップ 285 ページの『2』で追加した **trace**、**traceFiles**、および **traceSize** の各プロパティの項目を削除します。
これにより、次にエージェントを再始動するときにはトレースは有効にならなくなります。

タスクの結果

生成されたトレース・ファイルが、*BFG_DATA/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%* ディレクトリーに書き込まれます。ここで、%PID% はエージェントのプロセス ID です。

JCL を使用した起動時点からのエージェント・トレースの収集

始動時に JCL を使用して管理されている Managed File Transfer for z/OS エージェントのトレースを収集するには、そのエージェントが開始される前に、そのエージェントの **agent.properties** ファイルにさまざまなプロパティを設定する必要があります。

このタスクについて

起動時点からトレースを収集するには、以下の手順を実行します。

手順

1. トレースする必要のあるエージェントの **agent.properties** ファイルを見つけます。
agent.properties ファイルは、*BFG_DATA/mqft/config/coordination_qmgr_name/agents/agent_name* ディレクトリー内にあります。
2. ファイルを編集して、以下のプロパティのエントリーを追加します。
 - **trace**=トレース仕様
trace プロパティは、トレースする内部クラスおよびパッケージを決定します。IBM サポート担当員から特に指示されない限り、このプロパティには値 **com.ibm.wmqfte=all** を設定してください。
 - **traceFiles**=使用するトレース・ファイルの数
 - **traceSize**=各トレース・ファイルのサイズ (MB 単位)
traceFiles プロパティと **traceSize** プロパティは、収集するトレース・データ量を制御するために使用します。できる限り多くのトレース・データを収集するために、これらのプロパティには大きい値を設定する必要があります。

例えば、トレース仕様 **com.ibm.wmqfte=all** を使用してラッピング・トレースの 1GB を収集するには、以下の行を **agent.properties** ファイルに追加します。

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

この場合、エージェントは最大 5 つのファイル (各ファイルのサイズは 200 MB) にトレース・データを書き込みます。

これらのエージェント・プロパティについて詳しくは、[MFT agent.properties ファイル](#)を参照してください

3. トレースする必要のあるエージェントの JCL が含まれているデータ・セットを見つけます。
4. エージェントを停止するため、データ・セット内の BFGAGSP メンバーを実行依頼します。
5. データ・セット内の BFGAGST メンバーを実行依頼してエージェントを再始動します。

6. 問題を再現します。
7. データ・セット内の BFGAGSP メンバーを実行依頼してエージェントをもう一度停止します。
8. エージェントの `agent.properties` ファイルを編集し、ステップ 286 ページの『2』で追加した **トレース**、**トレース・ファイル**、および **トレース・サイズ** のプロパティのエントリーを削除します。
これにより、次にエージェントを再始動するときにはトレースは有効にならなくなります。

タスクの結果

結果として生じるトレース・ファイルは、`BFG_DATA/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` ディレクトリーに書き込まれます。ここで、`%PID%` はエージェントのプロセス ID です。

Managed File Transfer for z/OS エージェント・トレースの動的な収集

エージェントを短時間停止することが不可能な場合は、Managed File Transfer for z/OS エージェント・トレースを動的に収集する必要があります。

このタスクについて

トレースを収集する方法は、エージェントの管理に UNIX System Services (USS) と JCL のどちらを使用しているかによって異なります。

以下のオプションのどちらを使用すべきかわからない場合は、IBM サポート担当員にお問い合わせください。お客様が直面している問題にとって最善のトレース収集方法について担当員がアドバイスします。

手順

- 以下を参照してください。
 - UNIX System Services (USS) を使用している場合は、[287 ページの『USS を使用したエージェント・トレースの動的な収集』](#)を参照してください。
 - JCL を使用している場合は、[288 ページの『JCL を使用したエージェント・トレースの動的な収集』](#)を参照してください。

USS を使用したエージェント・トレースの動的な収集

fteSetAgentTraceLevel コマンドを使用すると、実行中のエージェントからトレースを収集できます。これは、IBM サポートで停止できないエージェントからのトレースを参照する必要がある場合に非常に役立ちます。

このタスクについて

fteSetAgentTraceLevel コマンドを使用して Managed File Transfer for z/OS エージェントからトレースを収集するには、以下の手順を実行します。

手順

1. 以下のコマンドを実行して、エージェントのトレースをオンにします。

```
fteSetAgentTraceLevel -traceAgent classes=level agent_name
```

`-traceAgent` パラメーターは、トレースする内部クラスおよびパッケージを決定します。IBM サポート担当員から特に指示されない限り、このプロパティには値 `com.ibm.wmqfte=all` を設定してください。

2. 問題を再現します。
3. 以下のコマンドを実行して、エージェントのトレースをオフにします。

```
fteSetAgentTraceLevel -traceAgent =off agent_name
```

4. エージェントがビジー状態の場合、トレース・ファイルがすぐに折り返され、問題の調査に必要な情報が上書きされる可能性があります。

エージェントを短時間停止できる場合は、以下のステップを実行して、収集されるトレース・データの量を減らします。それ以外の場合は、IBM サポートに連絡して、収集されるトレース・データの量を減らすための代替トレース仕様について検討してください。

- a) エージェントを停止するための時間を設けます。
- b) トレースする必要があるエージェントの `agent.properties` ファイルを見つけます。

`agent.properties` ファイルは、`BFG_DATA/mqft/config/coordination_qmgr_name/agents/agent_name` ディレクトリーにあります。

- c) ファイルを編集して、以下のプロパティーのエントリーを追加します。

traceFiles= 「使用するトレース・ファイルの数」

traceSize=`size_of_each_trace_file_in_MB`

traceFiles プロパティーと **traceSize** プロパティーは、収集するトレース・データ量を制御するために使用します。

traceFiles プロパティーのデフォルト値は 5、**traceSize** プロパティーのデフォルト値は 20 MB です。つまり、トレースを動的にオンにした場合、これらのプロパティーを設定していなければ、エージェントは 5 つの折り返しトレース・ファイル (各ファイルの最大サイズは 20 MB) にトレース情報を書き込みます。

できる限り多くのトレース・データを収集するために、これらのプロパティーには大きい値を設定する必要があります。

例えば、1 GB の折り返しトレースを収集するには、`agent.properties` ファイルに以下の行を追加します。

```
traceFiles=5
traceSize=200
```

この場合、エージェントは最大 5 つのファイル (各ファイルのサイズは 200 MB) にトレース・データを書き込みます。

これらのエージェント・プロパティーについて詳しくは、[MFT agent.properties](#) ファイルを参照してください

- d) **fteStopAgent** コマンドを実行して、エージェントを停止します。
- e) **fteStartAgent** コマンドを実行してエージェントを開始します。
- f) 以下のコマンドを実行して、エージェントのトレースをオンにします。

```
fteSetAgentTraceLevel -traceAgent trace_specification agent_name
```

- g) 問題を再現します。
- h) 以下のコマンドを実行して、エージェントのトレースをオフにします。

```
fteSetAgentTraceLevel -traceAgent =off agent_name
```

タスクの結果

生成されたトレース・ファイルが、`BFG_DATA/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` ディレクトリーに書き込まれます。ここで、`%PID%` はエージェントのプロセス ID です。

JCL を使用したエージェント・トレースの動的な収集

トレースを必要とするエージェント用の JCL を含むデータ・セット内の BFGAGTC メンバーを使用して、実行中の Managed File Transfer for z/OS エージェントからトレースを収集することができます。これは、IBM サポートが、停止できないエージェントからのトレースを参照する必要がある場合に非常に便利です。

このタスクについて

BFGAGTC メンバーを使用してエージェントからトレースを収集するには、以下の手順を実行します。

手順

1. トレースする必要があるエージェントの JCL が含まれているデータ・セットを見つけます。
2. データ・セット内の BFGAGTC メンバーを編集して、次のテキストが含まれている行を見つけます。

```
-traceAgent
```

この次のテキストに、トレースする内部クラスおよびパッケージのリストが含まれています。 デフォルトでは、このリストは次のように設定されています。

```
com.ibm.wmqfte=all
```

IBM サポート担当員から特に指示されない限り、この値はそのままにしておいてください。

3. BFGAGTC メンバーを実行依頼します。
4. 問題を再現します。
5. BFGAGTC メンバーをもう一度編集して、次のように **-traceAgent** パラメーターを **=off** に設定します。

```
-traceAgent =off +
```

6. BFGAGTC メンバーをもう一度実行依頼して、トレースをオフにします。
7. エージェントがビジー状態の場合は、トレース・ファイルがすぐに折り返され、問題を調査するために必要な情報が上書きされる可能性があります。

このような場合は、以下の 2 つのオプションがあります。

a) 最初のオプションは次のとおりです。

- i) エージェントを停止するための時間を設けます。
- ii) トレースする必要があるエージェントの `agent.properties` ファイルを見つけます。
`agent.properties` ファイルは、`BFG_DATA/mqft/config/coordination_qmgr_name/agents/agent_name` ディレクトリーにあります。
- iii) ファイルを編集して、以下のプロパティーのエントリーを追加します。

- **traceFiles**=使用するトレース・ファイルの数
- **traceSize**=各トレース・ファイルのサイズ (MB 単位)

traceFiles プロパティーと **traceSize** プロパティーは、収集するトレース・データ量を制御するために使用します。

traceFiles プロパティーのデフォルト値は 5、**traceSize** プロパティーのデフォルト値は 20 MB です。つまり、トレースを動的にオンにした場合、これらのプロパティーを設定していなければ、エージェントは 5 つの折り返しトレース・ファイル (各ファイルの最大サイズは 20 MB) にトレース情報を書き込みます。

できる限り多くのトレース・データを収集するために、これらのプロパティーには大きい値を設定する必要があります。

例えば、1 GB の折り返しトレースを収集するには、`agent.properties` ファイルに以下の行を追加します。

```
traceFiles=5  
traceSize=200
```

この場合、エージェントは最大 5 つのファイル (各ファイルのサイズは 200 MB) にトレース・データを書き込みます。

これらのエージェント・プロパティについて詳しくは、[MFT agent.properties](#) ファイルを参照してください

- iv) トレースする必要があるエージェントの JCL が含まれているデータ・セットを見つけます。
- v) エージェントを停止するため、データ・セット内の BFGAGSP メンバーを実行依頼します。
- vi) データ・セット内の BFGAGST メンバーを実行依頼してエージェントを再始動します。
- vii) データ・セット内の BFGAGTC メンバーを編集して、次のテキストが含まれている行を見つけます。

```
-traceAgent
```

この次のテキストに、トレースする内部クラスおよびパッケージのリストが含まれています。デフォルトでは、このリストは次のように設定されています。

```
com.ibm.wmqfte=all
```

IBM サポート担当員から特に指示されない限り、この値はそのままにしておいてください。

- viii) トレースを有効にする準備ができたなら、BFGAGTC メンバーを実行依頼します。
- ix) 問題を再現します。
- x) BFGAGTC メンバーをもう一度編集して、次のように **-traceAgent** パラメーターを *=off* に設定します。

```
-traceAgent =off +
```

xi) BFGAGTC メンバーをもう一度実行依頼して、トレースをオフにします。

- b) エージェントを短時間停止することが不可能な場合は、2つ目のオプションとして、IBM サポート担当員に連絡します。

そして、生成されるトレース・データの量を減らすために、別のトレース仕様の使用について話し合ってください。

タスクの結果

生成されたトレース・ファイルが、`BFG_DATA/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%` ディレクトリーに書き込まれます。ここで、`%PID%` はエージェントのプロセス ID です。

Managed File Transfer for z/OS コマンドのトレース

Managed File Transfer for z/OS のトレース機能は、お客様の問題や懸念を IBM サポートが診断するための機能です。ユーザーがこの機能を使用してコマンドをトレースすることができます。

このタスクについて



重要: コマンドのトレースでは、コマンドによって実行された処理に関する情報のみが収集されます。エージェントがそのコマンドの処理中に何らかのアクティビティを実行したとしても、そのアクティビティはトレースされません。

トレースを収集する方法は、コマンドの実行に UNIX System Services (USS) と JCL のどちらを使用しているかによって異なります。

使用するオプションが不確かな場合は、IBM サポート担当員にお問い合わせください。お客様が直面している問題のトレースを収集する最善の方法をお知らせします。

手順

- z/OS UNIX を使用している場合は、[285 ページの『USS を使用した起動時点からのエージェント・トレースの収集』](#)を参照してください。
- JCL を使用している場合は、[286 ページの『JCL を使用した起動時点からのエージェント・トレースの収集』](#)を参照してください。

UNIX System Services (USS) を使用して Managed File Transfer for z/OS コマンドのトレースを収集するには、以下の手順を実行します。

手順

1. コマンド・プロンプトを表示し、`BFG_PROD/bin` ディレクトリーに移動します。
2. コマンドを実行します。

```
./command_name -trace classes=level -tracePath directory_path command_arguments
```

説明:

- コマンド名は、トレースされるコマンドの名前です。
- `classes=level` は、使用するトレース・レベルと、トレースを有効にするクラスです。IBM サポート担当員から特に指示されない限り、`com.ibm.wmqfte=all` を設定してください。
- ディレクトリー・パスは、トレース・ファイルが書き込まれるディレクトリーです。
- `command_arguments` は、コマンドに渡す必要がある引数です。例えば、**ftePingAgent** コマンドの場合はエージェントの名前です。

タスクの結果

生成されたトレース・ファイルが、**-tracePath** パラメーターで指定したディレクトリーに書き込まれます。

トレース・ファイルは、`trace%PID%.txt.number` という名前になります。各部分の意味は次のとおりです。

- `%PID%` は、コマンドのプロセス ID です。
- の数は、トレース・ファイルのシーケンス番号です。通常、コマンドによって生成されたトレース情報は、シーケンス番号が 0 の単一のトレース・ファイル内に入れられます。

ただし、大量のトレース情報がコマンドによって生成されることがあります。そのような場合、トレースは複数のファイルに書き込まれます。現行のトレース・ファイルのシーケンス番号が 0、次に古いトレース・ファイルのシーケンス番号が 1、という順序になります。

コマンドのトレース出力は、最大 5 つの折り返しトレース・ファイルに書き込まれます。各トレース・ファイルの最大サイズは 20 MB です。

注: コマンドを実行するユーザーに、**-tracePath** パラメーターで指定したディレクトリーへの書き込み権限がない場合、トレース出力は標準エラーに書き込まれます。

例

この例では、**fteListAgents** コマンドをトレースし、トレースを `/u/fteuser` ディレクトリーに書き込みます。

```
./fteListAgents -trace com.ibm.wmqfte=all -tracePath /u/fteuser
```

この例では、**fteCreateTransfer** コマンドをトレースし、トレースを `/tmp` ディレクトリーに書き込みます。

```
./fteCreateTransfer -trace com.ibm.wmqfte=all -tracePath /tmp -t text -sa AGENT1  
-da AGENT2 -df /tmp/IEEUJV.txt "'SYS1.SAMPLIB(IEEUJV)'"
```

`/tmp` に書き込まれたトレース・ファイルには、**fteCreateTransfer** コマンドによって実行された処理に関する情報のみが入っています。例えば、エージェントに送信された転送要求メッセージをコマンドがどのように作成したかや、要求を受信したことを示す確認応答がエージェントから送り返されるのをコマ

ンドがどれだけ待機したかなどの情報です。転送自体に関する情報はトレース・ファイルに入っていません。

z/OS JCL を使用したコマンドのトレースの収集

JCL を使用して実行依頼している Managed File Transfer for z/OS コマンドのトレースを収集するには、以下の手順を実行する必要があります。

手順

1. トレースする必要があるコマンドの JCL が含まれているデータ・セットを見つけてます。
2. データ・セット内で、そのコマンドに対応するメンバーを見つけてます。
3. メンバーを編集して、トレースする必要があるコマンドの名前が含まれている行を見つけてます。その行を変更して、コマンド名と + 符号の間に次のテキストを含めます。

```
-trace classes=level -tracePath directory_path
```

ここで、

- `classes=level` は、使用するトレース・レベルと、トレースを有効にするクラスです。IBM サポート担当員から特に指示されない限り、`com.ibm.wmqfte=all` を設定してください。
 - `directory_path` は、トレース・ファイルを書き込む USS ディレクトリーです。
4. メンバーを実行依頼します。
 5. 問題が再現されたら、メンバーをもう一度編集して次のテキストを削除します。

```
-trace classes=level -tracePath directory_path
```

これは、手順 [292 ページ](#) の『3』で追加したテキストです。

タスクの結果

生成されたトレース・ファイルが、**-tracePath** パラメーターで指定したディレクトリーに書き込まれます。

トレース・ファイルは、`trace%PID%.txt.number` という名前になります。各部分の意味は次のとおりです。

- `%PID%` は、コマンドのプロセス ID です。
- の数は、トレース・ファイルのシーケンス番号です。通常、コマンドによって生成されたトレース情報は、シーケンス番号が 0 の単一のトレース・ファイル内に入れられます。

ただし、大量のトレース情報がコマンドによって生成されることがあります。そのような場合、トレースは複数のファイルに書き込まれます。現行のトレース・ファイルのシーケンス番号が 0、次に古いトレース・ファイルのシーケンス番号が 1、という順序になります。

コマンドのトレース出力は、最大 5 つの折り返しトレース・ファイルに書き込まれます。各トレース・ファイルの最大サイズは 20 MB です。

注: コマンドを実行するユーザーに、**-tracePath** パラメーターで指定したディレクトリーへの書き込み権限がない場合、トレース出力は標準エラーに書き込まれます。

例

この例では、**ftelistmonitors** コマンドをトレースするようにメンバー BFGMNL1 を変更しています。

```
/******  
/* <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
/*
//

```

このメンバーを実行依頼すると、**fteListMonitors** コマンドが USS ディレクトリー /u/trace にトレースを書き込みます。

z/OS Managed File Transfer for z/OS スタンドアロン・データベース・ログアーのトレース
 Managed File Transfer for z/OS のトレース機能は、お客様の問題や懸念を IBM サポートが診断するための機能です。この機能の動作は、さまざまなコマンドやプロパティーで制御できます。

このタスクについて

スタンドアロン・データベース・ログアーに関する問題を調査するためにトレース出力の提供を求められた場合は、以下のいずれかのオプションを使用してください。

使用するオプションが不確かな場合は、IBM サポート担当員にお問い合わせください。お客様が直面している問題のトレースを収集する最善の方法をお知らせします。

手順

- ログアーを短時間停止することが可能な場合は、ログアーのトレースを起動時点から収集します。
 詳細については [293 ページの『起動時点からの Managed File Transfer for z/OS スタンドアロン・データベース・ログアー・トレースの収集』](#) を参照してください。
- ログアーを停止できない場合は、**fteSetLoggerTraceLevel** コマンドを使用してトレースを動的に収集します。
 詳細については [296 ページの『Managed File Transfer for z/OS スタンドアロン・データベース・ログアー・トレースの動的な収集』](#) を参照してください。

z/OS 起動時点からの Managed File Transfer for z/OS スタンドアロン・データベース・ログアー・トレースの収集
 ログアーを短時間停止することが可能な場合は、起動時点から IBM MQ Managed File Transfer ログアー・トレースを収集してください。

このタスクについて

トレースを収集する方法は、ログアーの管理に UNIX System Services (USS) と JCL のどちらを使用しているかによって異なります。

以下のオプションのどちらを使用すべきかわからない場合は、IBM サポート担当員にお問い合わせください。お客様が直面している問題にとって最善のトレース収集方法について担当員がアドバイスします。

手順

- 以下を参照してください。
 - UNIX System Services (USS) を使用している場合は、[294 ページの『USS を使用した起動時点からのスタンドアロン・データベース・ログアー・トレースの収集』](#) を参照してください。

- JCL を使用している場合は、295 ページの『[JCL を使用した起動時点からのスタンドアロン・データベース・ロガー・トレースの収集](#)』を参照してください。

z/OS USS を使用した起動時点からのスタンドアロン・データベース・ロガー・トレースの収集
Unix System Services (USS) を使用して管理している Managed File Transfer for z/OS ロガーのトレースを
起動時点から収集するには、ロガーを起動する前にロガーの `logger.properties` ファイルにいくつかの
プロパティを設定する必要があります。

このタスクについて

起動時点からトレースを収集するには、以下の手順を実行します。

手順

1. トレースする必要があるロガーの `logger.properties` ファイルを見つけます。
`logger.properties` ファイルは、`BFG_DATA/mqft/config/coordination_qmgr_name/loggers/logger_name` ディレクトリーにあります。
2. ファイルを編集して、以下のプロパティのエントリーを追加します。
 - **trace**=トレース仕様
trace プロパティは、トレースする内部クラスおよびパッケージを決定します。IBM サポート担当員から特に指示されない限り、このプロパティには値 `com.ibm.wmqfte=all` を設定してください。
 - **traceFiles**=使用するトレース・ファイルの数
 - **traceSize**=各トレース・ファイルのサイズ (MB 単位)
traceFiles プロパティと **traceSize** プロパティは、収集するトレース・データ量を制御するために使用します。できる限り多くのトレース・データを収集するために、これらのプロパティには大きい値を設定する必要があります。例えば、トレース仕様 `com.ibm.wmqfte=all` を使用して、1 GB の折り返しトレースを収集するには、`logger.properties` ファイルに以下の行を追加します。

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

この場合、ロガーは最大 5 つのファイル (各ファイルのサイズは 200 MB) にトレース・データを書き込みます。
これらのロガー・プロパティについて詳しくは、[MFT logger.properties ファイル](#) を参照してください。
3. **fteStopLogger** コマンドを使用して、トレースする必要があるロガーを停止します。
4. **fteStartLogger** コマンドを実行してロガーを開始します。
5. 問題を再現します。
6. ロガーを停止します。
7. ロガーの `logger.properties` ファイルを編集し、ステップ 294 ページの『2』で追加した **trace**、**traceFiles**、および **traceSize** の各プロパティの項目を削除します。
これにより、次にロガーを再始動するときにはトレースは有効にならなくなります。

タスクの結果

生成されたトレース・ファイルが、`BFG_DATA/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/tracePID%` ディレクトリーに書き込まれます。ここで、`%PID%` はロガーのプロセス ID です。

JCL を使用した起動時点からのスタンドアロン・データベース・ロガー・トレースの収集 JCL を使用して管理している Managed File Transfer for z/OS ロガーのトレースを起動時点から収集するには、ロガーを起動する前にロガーの `logger.properties` ファイルにいくつかのプロパティを設定する必要があります。

このタスクについて

起動時点からトレースを収集するには、以下の手順を実行します。

手順

1. トレースする必要があるロガーの `logger.properties` ファイルを見つけます。

`logger.properties` ファイルは、`BFG_DATA/mqft/config/coordination_qmgr_name/loggers/logger_name` ディレクトリーにあります。

2. ファイルを編集して、以下のプロパティのエントリーを追加します。

- **trace**=トレース仕様

trace プロパティは、トレースする内部クラスおよびパッケージを決定します。IBM サポート担当員から特に指示されない限り、このプロパティには値 `com.ibm.wmqfte=all` を設定してください。

- **traceFiles**=使用するトレース・ファイルの数

- **traceSize**=各トレース・ファイルのサイズ (MB 単位)

traceFiles プロパティと **traceSize** プロパティは、収集するトレース・データ量を制御するために使用します。できる限り多くのトレース・データを収集するために、これらのプロパティには大きい値を設定する必要があります。

例えば、トレース仕様 `com.ibm.wmqfte=all` を使用して、1 GB の折り返しトレースを収集するには、`logger.properties` ファイルに以下の行を追加します。

```
trace=com.ibm.wmqfte=all
traceFiles=5
traceSize=200
```

この場合、ロガーは最大 5 つのファイル (各ファイルのサイズは 200 MB) にトレース・データを書き込みます。

これらのロガー・プロパティについて詳しくは、[MFT logger.properties ファイル](#) を参照してください。

3. トレースする必要があるロガーの JCL が含まれているデータ・セットを見つけます。
4. ロガーを停止するため、データ・セット内の BFGLGSP メンバーを実行依頼します。
5. データ・セット内の BFGLGST メンバーを実行依頼してロガーを再始動します。
6. 問題を再現します。
7. 再び、ロガーを停止するため、データ・セット内の BFGLGSP メンバーを実行依頼します。
8. ロガーの `logger.properties` ファイルを編集し、ステップ 295 ページの『2』で追加した **trace**、**traceFiles**、および **traceSize** の各プロパティの項目を削除します。
これにより、次にロガーを再始動するときにはトレースは有効にならなくなります。

タスクの結果

生成されたトレース・ファイルが、`BFG_DATA/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%` ディレクトリーに書き込まれます。ここで、`%PID%` はロガーのプロセス ID です。

z/OS Managed File Transfer for z/OS スタンドアロン・データベース・ロガー・トレースの動的な収集

ロガーを短時間停止することが不可能な場合は、Managed File Transfer for z/OS ロガー・トレースを動的に収集する必要があります。

このタスクについて

トレースを収集する方法は、ロガーの管理に Unix System Services (USS) と JCL のどちらを使用しているかによって異なります。

以下のオプションのどちらを使用すべきかわからない場合は、IBM サポート担当員にお問い合わせください。お客様が直面している問題にとって最善のトレース収集方法について担当員がアドバイスします。

手順

- 以下を参照してください。
 - UNIX System Services (USS) を使用している場合は、[296 ページの『USS を使用したスタンドアロン・データベース・ロガー・トレースの動的な収集』](#)を参照してください。
 - JCL を使用している場合は、[297 ページの『JCL を使用したスタンドアロン・データベース・ロガー・トレースの動的な収集』](#)を参照してください。

z/OS USS を使用したスタンドアロン・データベース・ロガー・トレースの動的な収集
(USS) では、**fteSetLoggerTraceLevel** コマンドを使用して、実行中のロガーからトレースを収集できます。これは、IBM サポートで停止できないロガーからのトレースを参照する必要がある場合に非常に役立ちます。

このタスクについて

fteSetLoggerTraceLevel コマンドを使用して Managed File Transfer for z/OS ロガーからトレースを収集するには、以下の手順を実行します。

手順

1. 以下のコマンドを実行して、ロガーのトレースをオンにします。

```
fteSetLoggerTraceLevel -traceLogger classes=level logger_name
```

-traceLogger パラメーターは、トレースする内部クラスおよびパッケージを決定します。IBM サポート担当員から特に指示されない限り、このプロパティには値 `com.ibm.wmqfte=all` を設定してください。

2. 問題を再現します。
3. 以下のコマンドを実行して、ロガーのトレースをオフにします。

```
fteSetLoggerTraceLevel -traceLogger =off logger_name
```

4. ロガーが使用中の場合は、トレース・ファイルがすぐに折り返され、問題の調査に必要な情報が上書きされる可能性があります。

ロガーを短時間停止できる場合は、以下のステップを実行して、収集されるトレース・データの量を減らします。それ以外の場合は、IBM サポートに連絡して、収集されるトレース・データの量を減らすための代替トレース仕様について検討してください。

- a) ロガーを停止するための時間を設けます。
- b) トレースする必要があるロガーの `logger.properties` ファイルを見つけます。

`logger.properties` ファイルは、`BFG_DATA/mqft/config/coordination_qmgr_name/loggers/logger_name` ディレクトリーにあります。

- c) ファイルを編集して、以下のプロパティのエントリーを追加します。

traceFiles=「使用するトレース・ファイルの数」

traceSize=*size_of_each_trace_file_in_MB*

traceFiles プロパティと **traceSize** プロパティは、収集するトレース・データ量を制御するために使用します。

traceFiles プロパティのデフォルト値は 5、**traceSize** プロパティのデフォルト値は 20 MB です。つまり、トレースを動的にオンにした場合、これらのプロパティを設定していなければ、エージェントは 5 つの折り返しトレース・ファイル (各ファイルの最大サイズは 20 MB) にトレース情報を書き込みます。

できる限り多くのトレース・データを収集するために、これらのプロパティには大きい値を設定する必要があります。

例えば、1 GB の折り返しトレースを収集するには、`logger.properties` ファイルに以下の行を追加します。

```
traceFiles=5
traceSize=200
```

この場合、ロガーは最大 5 つのファイル (各ファイルのサイズは 200 MB) にトレース・データを書き込みます。

これらのロガー・プロパティについて詳しくは、[MFT logger.properties ファイル](#)を参照してください

- d) **fteStopLogger** コマンドを実行して、ロガーを停止します。
- e) **fteStartLogger** コマンドを実行してロガーを開始します。
- f) 以下のコマンドを実行して、ロガーのトレースをオンにします。

```
fteSetLoggerTraceLevel -traceLogger trace_specification logger_name
```

- g) 問題を再現します。
- h) 以下のコマンドを実行して、ロガーのトレースをオフにします。

```
fteSetLoggerTraceLevel -traceLogger =off logger_name
```

タスクの結果

トレース・ファイルは、`BFG_DATA/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%` ディレクトリーに書き込まれます。ここで、`%PID%` はロガーのプロセス ID です。

z/OS JCL を使用したスタンドアロン・データベース・ロガー・トレースの動的な収集
トレースする必要があるロガーの JCL が含まれているデータ・セット内の BFGLGTC メンバーを使用して、実行中の Managed File Transfer for z/OS ロガーからトレースを収集できます。これは、IBM サポートで停止できないロガーからのトレースを参照する必要がある場合に非常に役立ちます。

このタスクについて

BFGLGTC メンバーを使用してロガーからトレースを収集するには、以下の手順を実行します。

手順

1. トレースする必要があるロガーの JCL が含まれているデータ・セットを見つけます。
2. データ・セット内の BFGLGTC メンバーを編集して、次のテキストが含まれている行を見つけます。

```
-traceLogger
```

この次のテキストに、トレースする内部クラスおよびパッケージのリストが含まれています。デフォルトでは、このリストは次のように設定されています。

```
com.ibm.wmqfte=all
```

IBM サポート担当員から特に指示されない限り、この値はそのままにしておいてください。

3. BFGLGTC メンバーを実行依頼します。
4. 問題を再現します。
5. BFGLGTC メンバーをもう一度編集して、次のように **-traceLogger** パラメーターを **=off** に設定します。

```
-traceLogger =off +
```

6. BFGLGTC メンバーをもう一度実行依頼して、トレースをオフにします。
7. ロガーがビジー状態の場合は、トレース・ファイルがすぐに折り返され、問題を調査するために必要な情報が上書きされる可能性があります。

このような場合は、以下の2つのオプションがあります。

a) 最初のオプションは次のとおりです。

- i) ロガーを停止するための時間を設けます。
- ii) トレースする必要があるロガーの `logger.properties` ファイルを見つけます。
`logger.properties` ファイルは、`BFG_DATA/mqft/config/coordination_qmgr_name/loggers/logger_name` ディレクトリーにあります。
- iii) ファイルを編集して、以下のプロパティーのエントリーを追加します。

- **traceFiles**=使用するトレース・ファイルの数
- **traceSize**=各トレース・ファイルのサイズ (MB 単位)

traceFiles プロパティーと **traceSize** プロパティーは、収集するトレース・データ量を制御するために使用します。

traceFiles プロパティーのデフォルト値は5、**traceSize** プロパティーのデフォルト値は20 MB です。つまり、トレースを動的にオンにした場合、これらのプロパティーを設定していなければ、ロガーは5つの折り返しトレース・ファイル (各ファイルの最大サイズは20 MB) にトレース情報を書き込みます。

できる限り多くのトレース・データを収集するために、これらのプロパティーには大きい値を設定する必要があります。

例えば、1 GB の折り返しトレースを収集するには、`logger.properties` ファイルに以下の行を追加します。

```
traceFiles=5  
traceSize=200
```

この場合、ロガーは最大5つのファイル (各ファイルのサイズは200 MB) にトレース・データを書き込みます。

これらのロガー・プロパティーについて詳しくは、[MFT logger.properties ファイル](#)を参照してください

- iv) トレースする必要があるロガーの JCL が含まれているデータ・セットを見つけます。
- v) ロガーを停止するため、データ・セット内の BFGLGSP メンバーを実行依頼します。
- vi) データ・セット内の BFGLGST メンバーを実行依頼してロガーを再始動します。
- vii) データ・セット内の BFGLGTC メンバーを編集して、次のテキストが含まれている行を見つけます。

```
-traceLogger
```

この次のテキストに、トレースする内部クラスおよびパッケージのリストが含まれています。デフォルトでは、このリストは次のように設定されています。

```
com.ibm.wmqfte=all
```

IBM サポート担当員から特に指示されない限り、この値はそのままにしておいてください。

viii) トレースを有効にする準備ができたなら、BFGLGTC メンバーを実行依頼します。

ix) 問題を再現します。

x) BFGLGTC メンバーをもう一度編集して、次のように **-traceLogger** パラメーターを *=off* に設定します。

```
-traceLogger =off +
```

xi) BFGLGTC メンバーをもう一度実行依頼して、トレースをオフにします。

b) ロガーを短時間停止することが不可能な場合は、2つ目のオプションとして、IBM サポート担当員に連絡します。

そして、生成されるトレース・データの量を減らすために、別のトレース仕様の使用について話し合ってください。

タスクの結果

生成されたトレース・ファイルが、`BFG_DATA/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%` ディレクトリーに書き込まれます。ここで、`%PID%` はロガーのプロセス ID です。

共通 MFT 問題

Managed File Transfer ネットワーク内で起きる可能性がある共通の問題。

- テキスト転送が、次のエラーを出して失敗する場合があります。

BFGI00060E: テキスト・データ変換が失敗しました

考えられる原因は次の2つのうちいずれかです。

1. ソース・ファイル内の1つ以上の文字を、ソース・ファイルのコード・ページから宛先ファイルのコード・ページに変換できません。この問題は、コード・ページの文字セットが異なり、その相互間で特定の文字を変換できない場合に発生します。

一部の文字が変換されなくても構わないのであれば、宛先エージェントに置換文字シーケンスを定義して、転送が失敗しないようにすることができます。置換文字シーケンスを定義するには、エージェント・プロパティー **textReplacementCharacterSequence** を指定します。詳しくは、[552 ページの表 45](#) を参照してください。

2. ソース・ファイルのエンコード方式が、ソース・エージェントのデフォルト・エンコード方式と一致していません。この場合、デフォルト設定を使用してテキスト転送を実行すると、文字データが壊れます。ソース・エージェントとはエンコード方式が異なるソース・ファイルを転送するには、次のステップのいずれかを実行します。
 - a. 転送定義ファイルでファイル・エンコード方式を指定します。詳しくは、[161 ページの『転送定義ファイルの使用』](#)を参照してください。
 - b. **fteCreateTransfer** コマンドで **-sce** パラメーターを使用して、ファイル・エンコード方式を指定します。詳しくは、[437 ページの『fteCreateTransfer: 新規ファイル転送の開始』](#)のトピックを参照してください。
 - c. ファイル・エンコード方式を Ant 移動またはコピー・タスクの一部として指定します。詳しくは、[252 ページの『MFTでのApache Antの使用』](#)を参照してください。

転送用に正しいソース・ファイル・エンコード方式を選択しているかどうかを確認するには、次のステップを実行します。

1. 宛先ファイルのエンコード方式を UTF-8 に設定します。

2. ファイルをテキスト・モードで転送します。
3. UTF-8 ファイル・ビューアーを使用して、ファイルの内容を表示します。ファイル内のすべての文字が正しく表示される場合、ソース・ファイルのエンコード方式は正しいということになります。

- **fteCreateAgent** コマンドから、次の出力が表示された場合:

```
BFGMQ1007I: 調整キュー・マネージャーに接続できないか、または調整キュー・マネージャーが
connection attempt.
The IBM MQ reason code was 2058. The agent's presence will not be published.
```

このメッセージは、調整キュー・マネージャーに接続できなかったことを示し、理由として IBM MQ の理由コードを提供しています。この情報メッセージは、調整キュー・マネージャーが現在使用不可であるか、構成が誤って定義されたことを示す可能性があります。

- ユーザー出口ルーチンを使用していて、ユーザー出口の呼び出し中または出口が呼び出された直後に障害が発生した場合 (例えばプロダクト障害や電力の切断など) には、そのユーザー出口が複数回呼び出される可能性があります。
- (固定 IP アドレスではなく) DHCP によって割り当てられた IP アドレスを使用するシステム上にキュー・マネージャーを持つエージェントがあり、かつそのエージェントがクライアントの TCP/IP 接続を使用してそのシステムに接続する場合には、次のシステム環境変数を設定して、エージェントを始動する必要があります。

- **Windows** Windows の場合:

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

- **UNIX** UNIX の場合:

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

ここで、*value* は JVM のキャッシュされた DNS 値の各フラッシュ間の時間間隔 (秒単位) です。キュー・マネージャー・システムの IP アドレスが何らかの理由で (例えば、ネットワーク障害、IP リースの満了、またはシステムのリブートのために) 再割り当てされた場合、エージェントは、キュー・マネージャーへの接続が失われたことを報告します。JVM DNS キャッシュがフラッシュされると、エージェントは正常に再接続できます。この環境変数が設定されていない場合、このシナリオで、JVM を再始動せずにエージェントを再接続することはできません。この動作は、JVM が内部的にホスト名の IP アドレスをキャッシュに入れ、デフォルトでそれらをリフレッシュしないためです。

- **fteStartAgent** コマンドを実行して以下のエラー・メッセージが表示された場合、Managed File Transfer と競合する他のライブラリー・パスが、ご使用の環境に含まれている可能性があります。

```
BFGCL0001E: 内部エラーが発生しました。 例外: 'CC=2;RC=2495;AMQ8568:
ネイティブ JNI ライブラリー「mqjbnd」が見つかりませんでした。 [3=mqjbnd]
```

LD_LIBRARY_PATH または LIBPATH 環境変数に、64 ビット版ライブラリーを 32 ビット版ライブラリーより先に参照するように設定している場合に、(現在、ほとんどのプラットフォームで使用されている) 32 ビット版の Java を使用してエージェントを実行すると、このエラーが発生します。

この問題を解決するには、Managed File Transfer エージェントのプロパティー `javaLibraryPath` を、ライブラリーの正しい場所を参照するように設定します。例えば、AIX の `mqjbnd` の場合、`/usr/mqm/java/lib` と設定します。Linux の `mqjbnd` の場合、`/opt/mqm/java/lib` に設定します。

- エージェントのプロパティー・ファイルに `authorityChecking=true` を指定することによって、ユーザー権限検査を有効にすると、関連する権限キューに対して必要な権限をユーザーが持っている場合であっても、権限検査はすべて失敗します。
 - エージェントを実行するユーザーは、エージェント・キュー・マネージャーに対して、ALT_USER アクセス制御を持つようにしてください。
- エージェント・プロパティー・ファイルに `authorityChecking=true` を指定してユーザー権限検査を有効にし、IBM MQ エラー・メッセージがエージェントの `output0.log` ファイルに書き込まれる場合は、以下のいずれかのアクションを実行します。
 - メッセージを無視します。エージェントに影響はありません。

- エージェントを実行するユーザーに、そのエージェントに属する SYSTEM.FTE.AUTH* キューに対する GET 権限を付与します。
- エージェントのプロパティ・ファイルを編集したのに、エージェントがそれらの編集を認識しない場合は、次のようにしてください。
 - エージェントを再始動し、エージェントが新しいプロパティを読み取ることを確認します。

z/OS

z/OS

- z/OS 上でエージェントを使用して PDS または PDSE データ・セットへ転送しているときに異常終了する場合、システムのディスク・スペースが限られている可能性があります。おそらく異常終了時のシステム完了コードは B14、戻りコードは 0C になっていて、空きスペースがないことを示しています。
 順次データ・セットに転送している場合、転送は失敗し、スペース不足の状態が示されますが、エージェントは引き続き作動可能です。
- z/OS でエージェントを使用していて、WMQFTEP タスクがいくつかの Java コア・ダンプを生成してから応答不能になった場合は、OMVS システム・サービス APAR OA43472 を適用します。
- z/OS で構成または管理スクリプトの実行中に次の出力が表示された場合:

```
FSUM7332 syntax error: got (, expecting Newline
```

この出力は、構成または管理スクリプトが実行されている環境で環境変数 `_BPXK_AUTOCVT=ON` が設定されていないことを示しています。この環境変数および設定方法については、[87 ページの『MFT の環境変数 \(z/OS\)』](#)を参照してください。

JZOS でよくある MFT 問題

ここでは、JZOS で問題が発生した場合の対処案をいくつか示します。

- JZOS が正常に処理できなかった場合:
 - JCL に `PARM='+T'` を追加します。以下に例を示します。

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

- 環境ファイルに `set -x` を追加します。

- 次の状態になった場合:

```
JVMJZBL1038E Child shell process exited with exit code: 1 (子シェル・プロセスが終了コード 1 で終了した)
```

```
JVMJZBL1042E JZOS batch launcher failed, return code=102 (JZOS バッチ・ランチャーが失敗した (戻りコード=102))
```

これは、環境ファイルおよび Managed File Transfer コマンドに誤りがあることを意味します。指定したパスが無効である可能性があります。

- 環境ファイルで、**BFG_PROD** の値を見つけます。

1. OMVS に入り、`ls -ltr` コマンドを使用します。

例えば、**BFG_PROD** が `/HMF8800/` である場合は、次のコマンドを入力します。

```
ls -ltr HMF8800/bin/fteBatch
```

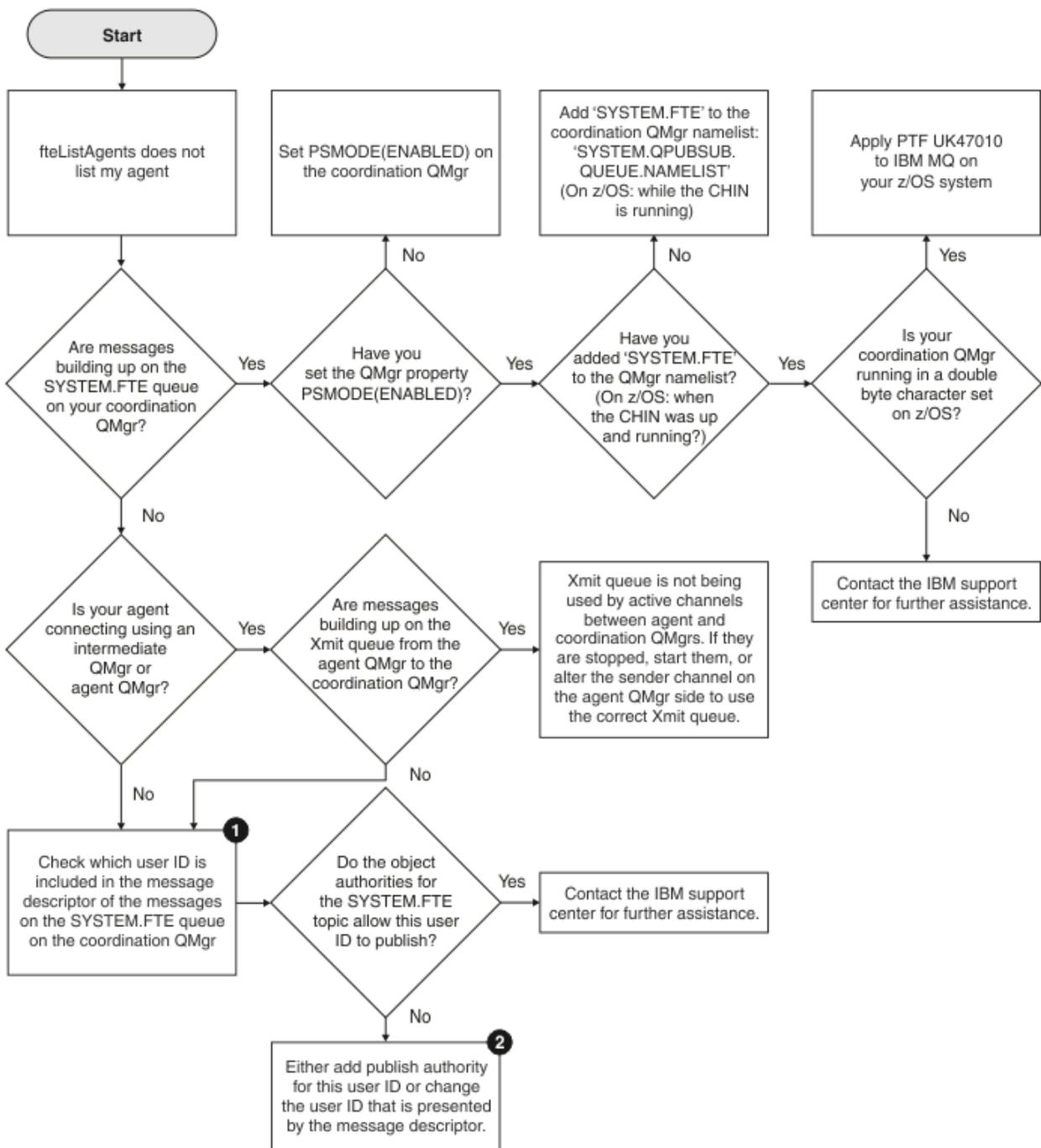
2. このファイルが存在することと、そのファイルに対する読み取り権限がバッチ・ジョブにあることを確認します。
 3. 問題を解決します。
- それでも JCL が正しく処理できない場合:
 1. USS にファイル (例えば `myenv`) を作成し、エディターを使用してこの `myenv` ファイルに環境ファイル内の情報をコピーします。

2. このファイルを保存します。
3. コマンド行で、コマンド **chmod +x myenv** を使用してファイルの実行を許可します。
4. コマンド **. myenv** を発行します。このコマンドは、ピリオド、ブランク、ファイル名であることに注意してください。
このコマンドを実行すると、myenv ファイル内のエラーが報告されます。
5. myenv と環境ファイルの両方のエラーを修正します。

MFT エージェントが **fteListAgents** コマンドによってリストされない場合に行う事柄

エージェントが **fteListAgents** コマンドによってリストされないか、IBM MQ Explorer に表示されない場合、または IBM MQ Explorer の「**転送ログ**」にファイル転送が表示されない場合は、いくつかの問題判別ステップを実行して原因を調査することができます。

以下のフローチャートを使用すると、問題を診断し、次に実行するアクションを決定する手助けとなります。



フローチャートの手掛かり:

- 提示されるユーザー ID を検査する方法について詳しくは、『319 ページの『パブリッシュ前のメッセージの検査』』を参照してください。ユーザー ID は、MQ ユーザー名の 12 文字の長さ制限に準拠する必要があります。ユーザー名が 12 文字より長い場合は (例えば、Administrator)、ユーザー名は認証検査前に切り捨てられます。例として、Administrator を使用した場合にキュー・マネージャー・エラー・ログに追加されるエラー・メッセージを次に示します。

AMQ8075: Authorization failed because the SID for entity 'administrato' cannot be obtained.

- SYSTEM.FTE キューに必要な権限について詳しくは、『364 ページの『MFT エージェント・ログ・メッセージおよび状況メッセージをパブリッシュする権限』』を参照してください。

UNIX MFT エージェント・プロセスがなくなっているのに診断情報がログに記録されていない場合

UNIX プラットフォームで、エージェント・プロセスがなくなっているのにエージェント・ログ・ファイルにその説明が一切記録されていない場合、エージェントの開始方法がその原因になっている可能性があります。

エージェントの診断情報は、以下の方法で確認できます。

- エージェントのログ・ファイルにエージェントの停止が記録されているかどうかを確認します。
- エージェントのロック・ファイル `agent.lck` がまだ存在しているかどうかを確認します。

例えば、エージェントをシェル・スクリプトから開始した場合、そのスクリプトに関連付けられているすべての子プロセス(エージェント・プロセスを含む)が、スクリプトの完了時に削除されます。そのエージェントを呼び出したスクリプトの完了後もエージェントが続けて実行されるようにするには、以下の手順を実行します。

1. **fteStartAgent** コマンドの前に **nohup** コマンドを付けて、**fteStartAgent** プロセス(およびそのすべての子プロセス)とスクリプトの関連付けを解除します。

これで、これ以降にスクリプトが終了しても、エージェントは実行を継続します。

エージェントが UNKNOWN 状態にあると表示される場合の処理

エージェントが実行中であり、**ftePingAgent** コマンドに正常に応答し、項目が正常に転送されています。ただし、**fteListAgents** コマンドと **fteShowAgentDetails** コマンド、および IBM MQ Explorer Managed File Transfer プラグインは、エージェントが UNKNOWN 状態であることを報告します。

この問題が発生する理由

各エージェントは、定期的に、調整キュー・マネージャー上の SYSTEM.FTE トピックにその状況をパブリッシュします。エージェントがその状況をパブリッシュする頻度は、以下のエージェント・プロパティーによって制御されます。

agentStatusPublishRateLimit

ファイル転送状況が変更されたためにエージェントがその状況を再公開する最大速度(秒単位)。このプロパティーのデフォルト値は 30 秒です。

agentStatusPublishRateMin

エージェントがその状況をパブリッシュする最小速度(秒単位)。この値は、

agentStatusPublishRateLimit プロパティーの値以上でなければなりません。

agentStatusPublishRateMin プロパティーのデフォルト値は 300 秒(つまり 5 分)です。

fteListAgents コマンド、**fteShowAgentDetails** コマンド、および IBM MQ Explorer Managed File Transfer (MFT) プラグインは、これらの資料を使用してエージェントの状況を判別します。これを実行するために、コマンドとプラグインは以下のステップを実行します。

1. 調整キュー・マネージャーに接続します。
2. SYSTEM.FTE トピックにサブスクライブします。
3. エージェント状況のパブリッシュの受信。
4. 調整キュー・マネージャー上に一時キューを作成します。
5. メッセージを一時キューに書き込み、調整キュー・マネージャー・システム上で現在時刻を取得するために、書き込み時間を保管します。
6. 一時キューを閉じます。
7. エージェントの状況を判別するには、資料内の情報、および現在の時刻を使用します。
8. 調整キュー・マネージャーから切断します。

エージェントが時代遅れの状態メッセージと見なされる場合、発表された時間と現在の時間との間の違いが、(状態メッセージに含まれる)エージェント属性 **agentStatusPublishRateMin** の値に加えて、高度な協調キューマネージャ属性 **agentStatusJitterTolerance** の値よりも大きい。

デフォルトでは、**agentStatusJitterTolerance** プロパティには 3000 ミリ秒 (3 秒) の値が設定されています。

agentStatusPublishRateMin プロパティと **agentStatusJitterTolerance** プロパティがデフォルト値に設定されている場合、エージェントが公開された時刻と現在時刻の差が 303 秒 (つまり 5 分 3 秒) を超えると、エージェントの状況は失効したと見なされます。

失効した状況メッセージがあるエージェントは、**fteListAgents** コマンド、**fteShowAgentDetails** コマンド、および IBM MQ Explorer MFT プラグインによって、UNKNOWN 状態であると報告されます。

エージェントの状況の公開は、以下のいずれかの理由で不整合になる場合があります。

1. エージェント・キュー・マネージャーが実行されているシステムと、調整キュー・マネージャーが配置されているシステムとの間のシステム時刻には、大きな違いがあります。
2. エージェント・キュー・マネージャーと調整キュー・マネージャーとの間のチャンネルは停止されます (これにより、新しい状況メッセージが調整キュー・マネージャーに到達しなくなります)。
3. 許可の問題により、エージェントが調整キュー・マネージャーの SYSTEM.FTE トピックにその状況をパブリッシュできなくなります。
4. エージェント障害が発生しました。

問題のトラブルシューティング

エージェントの状況が UNKNOWN として報告される理由を判別するには、いくつかのステップを実行する必要があります。

1. エージェント・システムにログオンして、エージェントが実行中であるかどうかを確認します。エージェントが停止している場合は、エージェントが実行されていない理由を調べてください。再度実行されると、その状況が正しく報告されているかどうかを確認します。
2. 調整キュー・マネージャーが実行中であることを確認します。そうでない場合は、再度起動して、**fteListAgents** または **fteShowAgentDetails** コマンド、または IBM MQ Explorer MFT プラグインを使用して、プロキシ状態が正しく報告されているかどうかを確認しましょう。
3. エージェントおよび調整キュー・マネージャーが実行されている場合は、エラー・ログを調べて、エージェントがその状況メッセージをパブリッシュするのを妨げている許可の問題があるかどうかを確認してください。ログに、許可の問題が発生していることが示されている場合は、エージェント・プロセスを実行しているユーザーが、調整キュー・マネージャー上の SYSTEM.FTE トピックにメッセージをパブリッシュするための正しい権限を持っていることを確認してください。

キュー・マネージャーのエラー・ログに許可の問題が報告されない場合は、状況メッセージが IBM MQ ネットワークにスタックされていないことを確認してください。エージェント・キュー・マネージャーから調整キュー・マネージャーへのメッセージの経路指定に使用される、送信側チャンネルと受信側チャンネルがすべて実行中であることを確認します。

チャンネルが実行中の場合は、チャンネルに関連付けられている伝送キューを確認し、状況メッセージがこれらのキューにスタックされていないことを確認します。また、キュー・マネージャーのデッド・レター・キューを確認して、状況メッセージが何らかの理由でそこに置かれていないことを確認する必要があります。

4. チャンネルが実行中で、状況メッセージが IBM MQ ネットワークを流れている場合、次に確認すべきことは、キュー・マネージャーのキューに入れられたパブリッシュ/サブスクライブ・エンジンがメッセージをピックアップしていることです。

調整キュー・マネージャーを定義するために使用される **fteSetupCoordination** コマンドは、キューに入れられたパブリッシュ/サブスクライブ・エンジンがパブリケーションを受信するように構成するために調整キュー・マネージャーで実行する必要がある MQSC コマンドをいくつか提供します。これらのコマンドは、以下のステップを実行します。

- SYSTEM.FTE トピックとそれに関連したトピック・ストリング。

- SYSTEM.FTE。
- キュー・マネージャーの **PSMODE** 属性を ENABLED に設定して、キューに入れられたパブリッシュ/サブスクライブ・エンジンを有効にします。
- SYSTEM.QPUBSUB.QUEUE.NAMELIST 名前リスト。これには、新しい SYSTEM.FTE キュー。

これについて詳しくは (実行する必要がある MQSC コマンドを含む)、[514 ページの『fteSetupCoordination』](#) を参照してください。

SYSTEM.FTE キューの場合は、SYSTEM.QPUBSUB.QUEUE.NAMELIST 名前リストが正しくセットアップされ、そのキューの項目が含まれています。項目が欠落している場合、キューに入れられたパブリッシュ/サブスクライブ・エンジンは、エージェントからの着信状況メッセージを検出せず、それらのメッセージを処理しません。

また、キュー・マネージャーの **PSMODE** 属性が ENABLED に設定されていることも確認してください。これにより、キューに入れられたパブリッシュ/サブスクライブ・エンジンがオンになります。

5. チャンネルが実行中で、状況メッセージが IBM MQ ネットワーク キュー・マネージャーのキューに入れられたパブリッシュ/サブスクライブ・エンジンによって SYSTEM.FTE キューから取得されているを流れている場合は、以下のトレースを収集します。

- エージェント・プロパティ **agentStatusPublishRateMin** の値の 3 倍に相当する期間をカバーする、エージェントからの IBM MQ MFT トレース。これにより、エージェントがその状況を含む少なくとも 3 つのメッセージをパブリッシュするときの時間を確実にカバーするようになります。トレース仕様を使用して、トレースを動的に収集する必要があります。

```
com.ibm.wmqfte.statestore.impl.FTEAgentStatusPublisher,
com.ibm.wmqfte.utils.AgentStatusDetails,
com.ibm.wmqfte.wmqiface.AgentPublicationUtils,
com.ibm.wmqfte.wmqiface.RFHMessageFactory=all
```

注：これらのストリングを使用すると、トレースの量が少なくなります。

IBM MQ for Multiplatforms 上で実行されているエージェントのトレースを使用可能にする方法については、[279 ページの『Managed File Transfer エージェント・トレースの動的な収集』](#) を参照してください。

IBM MQ for z/OS 上で実行されているエージェントのトレースを使用可能にする方法については、[287 ページの『Managed File Transfer for z/OS エージェント・トレースの動的な収集』](#) を参照してください。

- エージェント・キュー・マネージャーから調整キュー・マネージャーに状況メッセージを経路指定するために使用されるキュー・マネージャーの並行トレース。
- **fteListAgents** コマンドのトレース。エージェントが UNKNOWN 状態であることが示されている時間をカバーします。トレース仕様を使用してトレースを収集する必要があります。

```
com.ibm.wmqfte=all
```

IBM MQ for Multiplatforms 情報に関して実行するコマンドのトレースを使用可能にする方法については、[280 ページの『Multiplatforms での Managed File Transfer コマンドのトレース』](#) を参照してください。

IBM MQ for z/OS 情報に関して実行するコマンドのトレースを使用可能にする方法については、[290 ページの『Managed File Transfer for z/OS コマンドのトレース』](#) を参照してください。

トレースが収集されると、IBM サポートがトレースを分析のために使用できるようにする必要があります。

関連情報

[fteListAgents](#)

[fteShowAgentDetails](#)

MFT エージェントの状況値

[MFT agent.properties](#) ファイル

ftePingAgent がタイムアウトして BFGCL0214I メッセージが報告される場合の対処法

ftePingAgent は、IBM MQ Managed File Transfer に用意されている便利なコマンド行ユーティリティです。これを使用すると、エージェントが到達可能かどうか、およびエージェントが要求に回答できるかどうかを確認できます。

コマンドの動作

ftePingAgent コマンドを使用して、エージェントが到達可能かどうか、およびエージェントが要求を処理できるかどうかを確認できます。コマンドを実行すると、以下のステップが実行されます。

- Managed File Transfer (MFT) トポロジーのコマンド・キュー・マネージャーに接続します。
- コマンド・キュー・マネージャー上に一時応答キューを作成します。

デフォルトでは、この一時キューには接頭部 WMQFTE で始まる名前が付けられます。ただし、これを変更するには、インストール済み環境の **MFT command.properties** ファイルで **dynamicQueuePrefix** プロパティを設定します。

- コマンド・キュー・マネージャーを経由して、エージェント・キュー・マネージャー上のキュー **SYSTEM.FTE.COMMAND.agent_name** に Ping MFT エージェント要求メッセージを送信します。要求メッセージには、一時応答キューの名前が含まれています。
- 応答メッセージが一時応答キューに到着するのを待ちます。

エージェント内のスレッドの1つは **CommandHandler** です。このスレッドは、エージェントの **SYSTEM.FTE.COMMAND.エージェント名** キューからメッセージを取得し、処理します。

このスレッドは、Ping MFT エージェント要求を含むメッセージを受信すると、MFT エージェント応答メッセージを作成し、それをコマンド・キュー・マネージャーの一時キューに送信します。このメッセージは、エージェントのキュー・マネージャーを通過します。

メッセージが一時キューに到着すると、**ftePingAgent** コマンドによって取り出されます。次に、コマンドは終了する前に、以下に示すようなメッセージをコンソールに書き込みます。

BFGCL0213I: agent < agent_name > が 0.088 秒で ping に応答しました。

次の2つの図は、このフローを表しています。

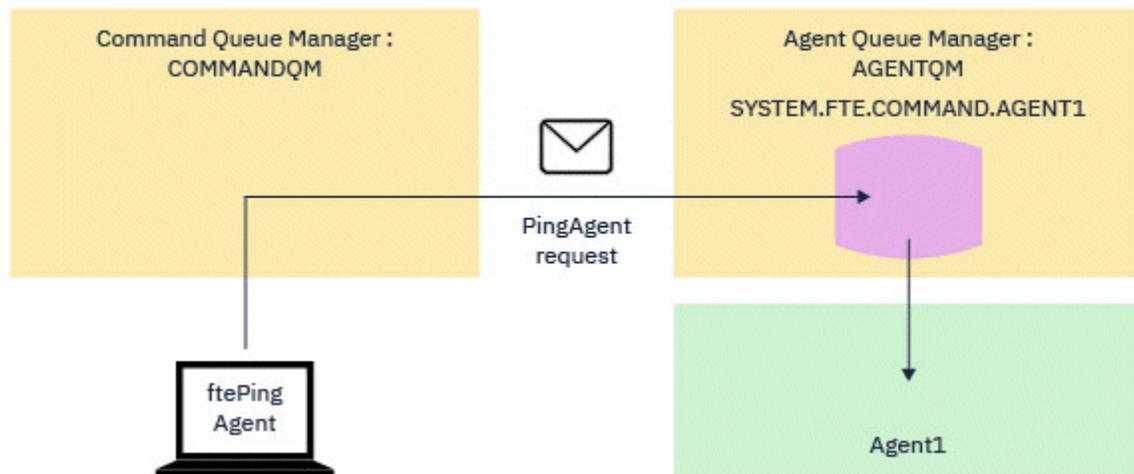


図 7. pingAgent 要求がコマンド・キュー・マネージャーを経由してエージェント・キュー・マネージャー上の **SYSTEM.FTE.COMMAND.agent_name** キューに届く



図 8. pingAgent 応答がエージェント・キュー・マネージャーを経由してコマンド・キュー・マネージャーに戻される

コマンドがタイムアウトになる場合の対処法

デフォルトでは、**ftePingAgent** コマンドは、MFT エージェント応答メッセージが一時キューに到着するまで 5 秒間待機します。応答メッセージが 5 秒以内に到着しない場合、コマンドは BFGCL0214I メッセージをコンソールに書き込みます。以下に、メッセージの例を示します。

BFGCL0214I: agent AGENT1 が 5 秒後に ping に応答しませんでした。

以下のステップを使用して、応答メッセージが到着しなかった理由を調べることができます。

- 最初に行うことは、エージェントが実行中であることを確認することです。そうでない場合、コマンドによって送信された Ping MFT エージェント要求に応答できません。
- エージェントが実行中で、要求の処理でビジー状態になっている場合、Ping MFT エージェント要求を取得して応答を返送するのに 5 秒より長くかかることがあります。

これが該当するかどうかを確認するには、**-w** パラメーターを使用してより長い待機間隔を指定し、**ftePingAgent** コマンドを再実行する必要があります。例えば、60 秒の待機間隔を指定するには、以下のコマンドを発行します。

```
ftePingAgent -w 60 AGENT1
```

- それでもコマンドがタイムアウトになる場合は、コマンド・キュー・マネージャーとエージェント・キュー・マネージャーの間の IBM MQ ネットワーク経由のパスを確認してください。パス内の 1 つ以上のチャンネルで障害が発生した場合、Ping MFT エージェント要求メッセージまたは MFT エージェント応答メッセージ (あるいはその両方) が伝送キューのどこかでスタックされます。この場合は、チャンネルを再始動して、**ftePingAgent** コマンドを再実行する必要があります。

上記のステップを実行した後もコマンドが BFGCL0214I メッセージを報告する場合は、IBM MQ ネットワークを流れる Ping MFT エージェント要求メッセージと MFT エージェント応答メッセージを追跡して、以下を確認する必要があります。

- ping MFT エージェント・メッセージが SYSTEM.FTE.COMMAND.agent_name キューに到達するかどうか。
- エージェントがキューからメッセージをピックアップし、MFT エージェントの応答メッセージを返送する場合。

これを行うには、以下のステップを実行する必要があります。

- コマンド・キュー・マネージャーとエージェント・キュー・マネージャーの両方でキュー・マネージャー・トレースを有効にします。
- トレース仕様 com.ibm.wmqfte=all を使用して、エージェントのトレースを動的に有効にします。

これを行う方法は、エージェントが実行されているプラットフォームによって異なります。実行中のエージェントの場合:

- IBM MQ for Multiplatforms を使用している場合は、[279 ページの『Managed File Transfer エージェント・トレースの動的な収集』](#)を参照してください。
- IBM MQ for z/OS を使用している場合は、[287 ページの『Managed File Transfer for z/OS エージェント・トレースの動的な収集』](#)を参照してください。
- 次に、トレース仕様 `com.ibm.wmqfte=all` を使用して、トレースを有効にして **ftePingAgent** コマンドを実行します。コマンドのトレースについては、以下を参照してください。
 - IBM MQ for Multiplatforms を使用している場合は、[280 ページの『Multiplatforms での Managed File Transfer コマンドのトレース』](#)を参照してください。
 - IBM MQ for z/OS を使用している場合は、[290 ページの『Managed File Transfer for z/OS コマンドのトレース』](#)を参照してください。

コマンドがタイムアウトになると、エージェント・トレースとキュー・マネージャー・トレースを停止します。その後、エージェント・トレースおよびキュー・マネージャー・トレースを、コマンドからのトレースとともに、分析のために IBM サポートが使用できるようにする必要があります。

MFT エージェントまたはロガーの構成が安全でない場合の対策

構成ファイルに機密情報が含まれている、構成ファイルが鍵ストア・ファイルまたはトラストストア・ファイルである、構成ファイルに対する読み取り権限、書き込み権限、または削除権限がシステム全体に設定されているという状態を Managed File Transfer プロセスが起動時に検出すると、プロセスは開始に失敗します。そうした状態が起動時には検出されなかったが、実行時に検出された場合は、Managed File Transfer によって警告メッセージが生成され、構成ファイルの内容が無視されます。これは、エージェントの実行中に構成が変更された場合に構成を再ロードする、プロトコル・ブリッジおよび Connect:Direct ブリッジのケイバビリティに関係します。

以下の確認を実行して、問題の原因を判別してください。

1. 表示されたエラー・メッセージから、安全でないと報告された構成ファイルを識別します。
2. ファイル・アクセス許可が要件と一致していることを確認します。詳細内容は [を参照してください](#)。
3. エージェントまたはロガーを再始動します。また、プロトコル・ブリッジまたは Connect:Direct 資格情報ファイルの場合は、次の再ロードを待ちます。

例

この例のエラー・メッセージが出された場合は、データベース・ロガーが起動に失敗します。

```
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'.
```

この例のエラー・メッセージが出された場合は、プロトコル・ブリッジ・エージェントが起動に失敗します。

```
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'.
```

関連資料

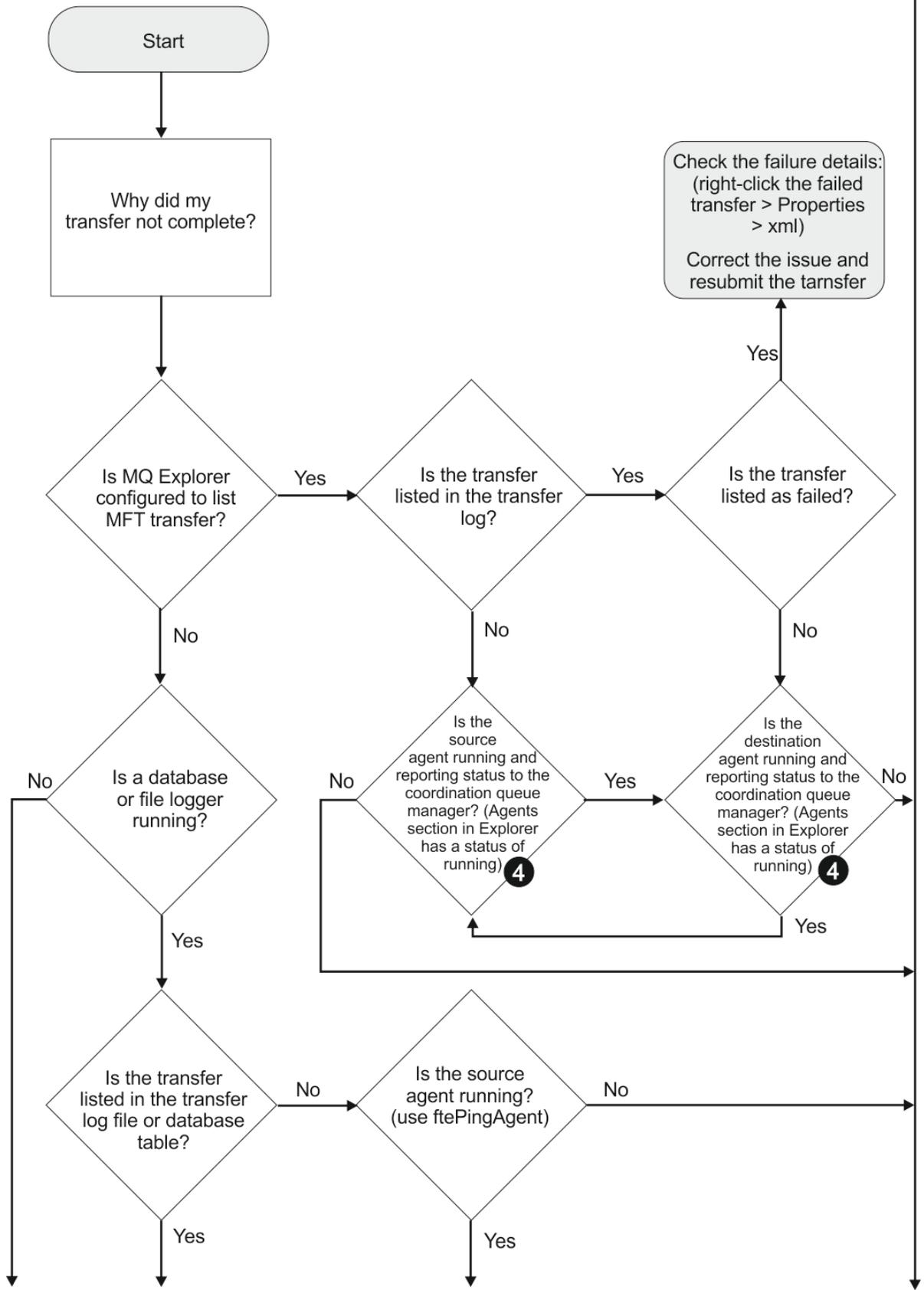
[363 ページの『機密構成情報にアクセスするための MFT アクセス権』](#)

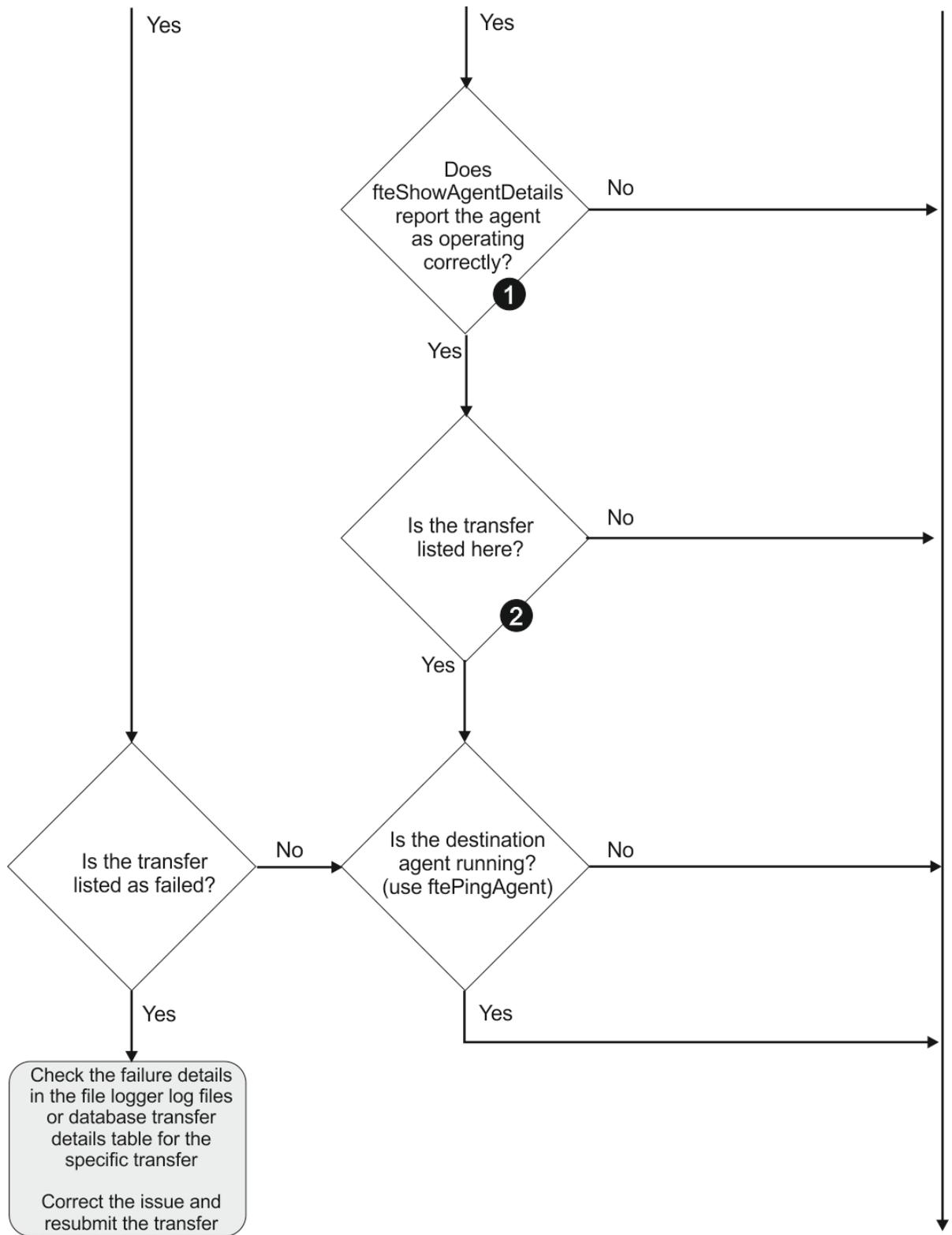
機密構成情報を保管するために使用するファイル(つまり IBM MQ 構成ツリーから参照されるファイル)に対する読み取り権限、書き込み権限、または削除権限(可能な場合)は、システム全体に対して付与してはなりません。この制限は、トラストストア・ファイルと鍵ストア・ファイルにも当てはまります。

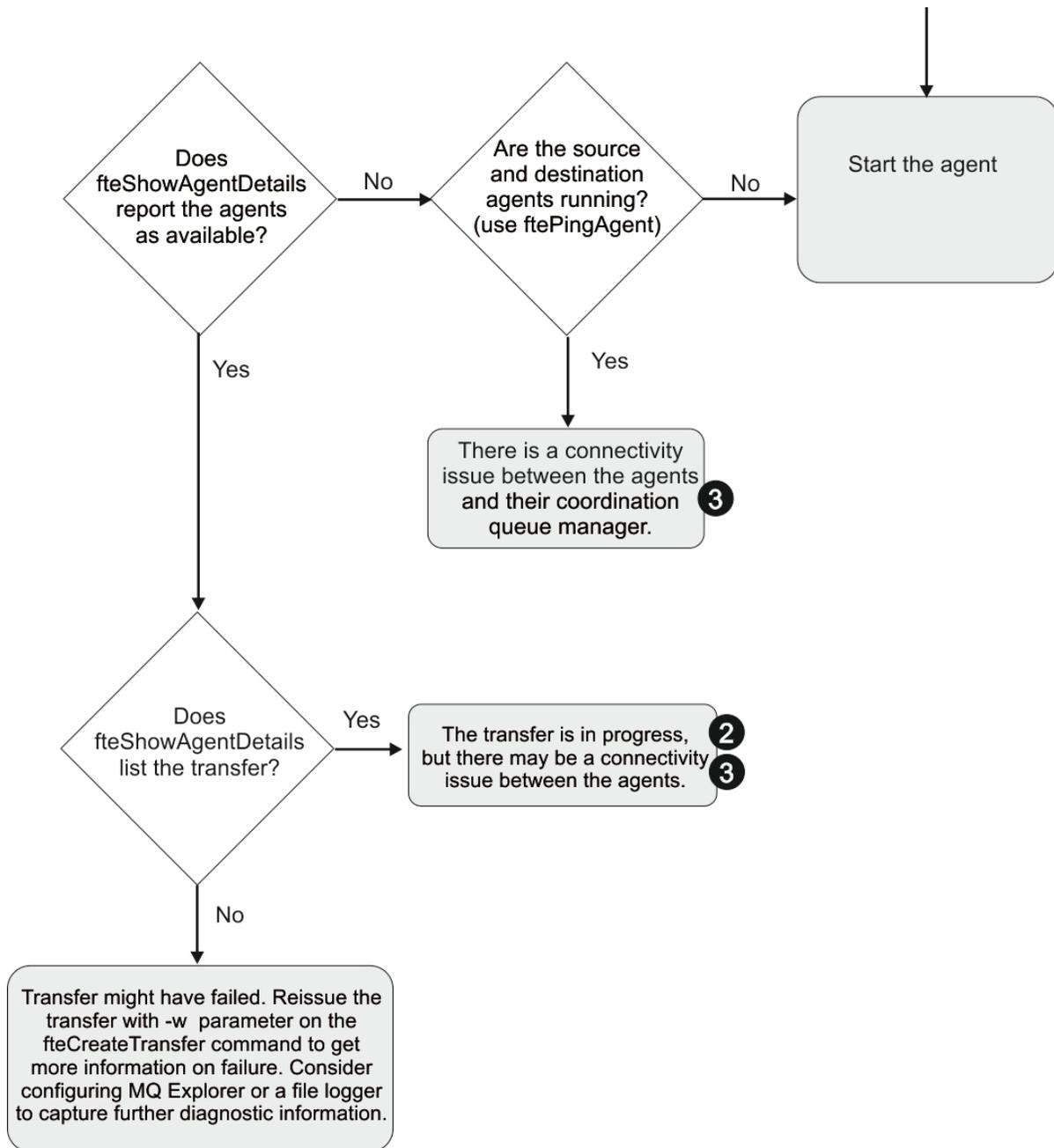
転送が未完了の場合の対策

転送が完了しなかった場合は、原因を調査するためにいくつかの問題判別ステップを実行することができます。

以下のフローチャートを使用すると、問題を診断し、次に実行するアクションを決定する手助けとなります。







フローチャートの手掛かり:

1. エージェントの `output0.log` にエラーが報告されていないか調べます。正常に開始したとエージェントから報告されたのに、IBM MQ Explorer でも **`fteShowAgentDetails`** でもエージェントが実行されていると報告されない場合は、エージェント・キュー・マネージャーと調整キュー・マネージャーの間の接続を確認してください。キュー・マネージャー間のチャンネルが使用不可になっている可能性があります。
2. ソース・エージェントには In progress 転送として転送 ID がリストされているもの、宛先エージェントにはリストされていない場合は、ソース・キュー・マネージャーと宛先キュー・マネージャーとの間の接続に問題がある可能性があります。 `command.properties` ファイル内で、コマンド・キュー・マネージャーとして宛先エージェント・キュー・マネージャーを使用して、宛先エージェント・マシンからソース・エージェントに対して **`ftePingAgent`** コマンドを使用します。また、このコマンドを逆方向に (ソースから宛先に向けて) 実行することもできます。

3. ソースと宛先の両方のエージェントに In progress として転送 ID がリストされている場合は、転送は開始されているため、ソース・キュー・マネージャーと宛先キュー・マネージャーの間に接続の問題が生じていることになります。command.properties ファイル内で、コマンド・キュー・マネージャーとして宛先エージェント・キュー・マネージャーを使用して、宛先エージェント・マシンからソース・エージェントに対して **ftePingAgent** コマンドを使用します。また、このコマンドを逆方向に (ソースから宛先に向けて) 実行することもできます。
4. このループを既に繰り返している場合は、いずれかのステートメントが以下の状態に関係していないか確認します。
 - ソースと宛先の両方のエージェントが Running として報告されているが、転送がリストされていない。転送要求がエージェント・コマンド・キューに到達しなかったか、または Running と報告されたエージェントがコマンド・キューの監視を既に行っていません。ソース・エージェント output0.log にエラーがないか調べてください。転送元の同じマシンからソース・エージェントに対して **ftePingAgent** コマンドを使用して、コマンド・キュー・マネージャーとエージェント・キュー・マネージャーの間の接続を確認し、エージェントがコマンド・キューを処理していることを確認します。
 - ソースと宛先の両方のエージェントが Running として報告され、転送が In progress、recovering としてリストされている。command.properties ファイル内で、コマンド・キュー・マネージャーとして宛先エージェント・キュー・マネージャーを使用して、宛先エージェント・マシンからソース・エージェントに対して **ftePingAgent** コマンドを使用します。また、このコマンドを逆方向に (ソースから宛先に向けて) 実行することもできます。

ファイル転送が停止したと思われる場合の対処法

負荷の重いシステムであったり、ソース・エージェントと宛先エージェントとの間でネットワーク問題があったりする場合には、転送が待機状態またはリカバリー状態で停止しているように見えることがあります。この原因となるいくつかの要因があります。

以下の確認を実行して、問題の原因を判別してください。

1. **ftePingAgent** コマンドを使用するか、または IBM MQ Explorer の「エージェント」パネルでエージェント名を右クリックして「Ping」を選択することにより、ソース・エージェントおよび宛先エージェントがアクティブであり、新しい要求に応答しているかどうかを確認します。エージェント・ログを参照して、現行のネットワーク接続に問題があるかどうかを確認します。
2. 宛先エージェントが能力どおりに稼働しているかどうかを確認します。多数のソース・エージェントがあり、そのすべてが同じ宛先エージェントへのファイル転送を要求しているという可能性があります。**fteShowAgentDetails** コマンドに **-v** (冗長) パラメーターを指定するか、または IBM MQ Explorer の「エージェント」パネルでエージェント名を右クリックして「プロパティ」を選択することにより、エージェントの現行の転送アクティビティを確認します。実行中の宛先転送の数が、エージェントの宛先転送の最大数またはそれに近い数である場合、それがソース・エージェントの一部の転送が停止しているように見える原因です。
3. プロトコル・ブリッジ・エージェントとの間でやり取りされる転送は、プロトコル・ファイル・サーバーとの通信に問題がある場合には、リカバリー状態になります。エージェント・ログを参照して、現行の接続に問題があるかどうかを確認します。
4. 転送はエージェントにより優先順位に従って処理されます。したがって負荷のかかったシステムでは、エージェントに高優先順位の転送負荷がある場合には、低優先順位の転送はしばらくの間待機状態となることがあります。最終的には低優先順位の転送は、それより新しく優先順位が高い転送があったとしても、しばらく待機していれば開始されることになります。

スケジュール済みファイル転送が実行されなかったり遅延したりする場合の対策

スケジュール済み転送があり、予定した時点で実行されなかったり遅延したりする場合は、エージェントがそのコマンド・キュー上のコマンドを処理していることがその理由である可能性があります。エージェントがビジー状態なので、スケジュール済み転送が検査されず、したがって実行されません。

この問題に対処するには、以下のいずれかのステップを実行してください。

- `agent.properties` ファイル内の `maxSchedulerRunDelay` プロパティを構成して、エージェントがスケジュール済み転送がないか検査するのを待つ最大間隔を分単位で設定します。このプロパティを設定すると、エージェントがビジー状態の場合でも、スケジュール済み転送がないかエージェントが検査し続けます。このプロパティについては、[550 ページの『MFT agent.properties ファイル』](#)を参照してください。
 - または、スケジュール済み転送の代わりにリソース・モニターを使用します。リソース・モニターの働きはスケジュール済み転送とは異なり、エージェントがビジー状態であることの影響を受けません。例えば、宛先システムに最新のファイルが必要な場合、リソース・モニターを使用するとネットワーク・トラフィックが少なくなります。その理由は、ファイルが自動的に転送されるのではなく、新しいバージョンが使用可能になった時点に限り転送されるからです。しかし、リソース・モニターは、プロトコル・ブリッジ・エージェントや `Connect:Direct` ブリッジ・エージェントではサポートされません。
- 詳細については [170 ページの『MFT リソースのモニター』](#) を参照してください。

プロトコル・ブリッジ・エージェントがファイル未検出を報告する場合の対処法

プロトコル・ブリッジの接続先である SFTP サーバーまたは FTP サーバーが `File not found` エラー・メッセージを返したことをプロトコル・ブリッジ・エージェントが報告した場合、このメッセージは、いくつかの異なるエラー・ケースのうちの 1 つが発生したことを意味する可能性があります。

以下の考えられるシナリオでは、SFTP サーバーまたは FTP サーバーから `File not found` エラーが返される可能性があります。

- ファイルが存在しない。転送しようとしているファイルが、SFTP または FTP サーバーをホストしているシステム上に存在することを確認してください。
- ファイル・パスが存在しない。SFTP または FTP サーバーをホストしているシステム上にファイル・パスが存在することを確認してください。また、転送要求にファイル・パスを正しく入力したことを確認してください。必要であれば、ファイル・パスを訂正して、転送要求を再び実行依頼してください。
- ファイルが別のアプリケーションによってロックされている。ファイルが別のアプリケーションによってロックされていないかどうか、確認してください。ファイルがロックされていない状態になるまで待つから、転送要求を再び実行依頼してください。
- ファイル・アクセス許可により、そのファイルの読み取りが許可されていない。そのファイルに正しいファイル・アクセス許可があることを確認してください。必要であれば、ファイル・アクセス許可を変更して、転送要求を再び実行依頼してください。
- SFTP または FTP サーバーが、仮想化されたルート・パスを使用している。転送要求に相対ファイル・パスが指定されると、プロトコル・ブリッジ・エージェントは、プロトコル・サーバーへのログインに使用されたホーム・ディレクトリーに基づいて、相対パスを絶対ファイル・パスに変換しようとします。Managed File Transfer プロトコル・ブリッジ・エージェントは、絶対ファイル・パスを使用したファイルへのアクセスが可能な SFTP または FTP サーバーのみをサポートできます。現行ディレクトリーに基づいたファイルへのアクセスのみが可能なプロトコル・サーバーは、プロトコル・ブリッジ・エージェントではサポートされません。

関連概念

[214 ページの『プロトコル・ブリッジ』](#)

プロトコル・ブリッジを使用すれば、Managed File Transfer (MFT) ネットワークから、MFT ネットワークの外部（ローカル・ドメインとリモート・ロケーションの両方）にあるファイル・サーバーに格納されているファイルにアクセスできます。このファイル・サーバーでは、FTP、FTPS、または SFTP ネットワーク・プロトコルを使用できます。それぞれのファイル・サーバーで少なくとも 1 つの専用エージェントが必要です。この専用エージェントは、プロトコル・ブリッジ・エージェントとして知られています。ブリッジ・エージェントは、複数のファイル・サーバーと相互作用できます。

MFT ディレクトリー・リソース・モニターがファイルをトリガーしない場合の対応策

ディレクトリー・リソース・モニターは、トリガー仕様に一致するファイルについて、ディレクトリーをポーリングします。トリガー仕様に一致する各ファイルについて、エージェントに対する転送要求が生成されます。要求が実行依頼されると、ファイルに対するトリガーは、ファイルが変更されるまで無視されます。

ファイルがトリガーされない理由

1. ディレクトリー・リソース・モニターがトリガー仕様に一致するファイルを検出しましたが、生成された転送要求が無効で、エージェントが要求を処理できませんでした。以下の理由が考えられます。

- 宛先エージェントが無効である
- 宛先エージェントが欠落している
- プログラム呼び出しによって転送が取り消された

これらすべての例では、ディレクトリー・リソース・モニターはトリガーするファイルを処理済みとしてマークし、転送が失敗した場合でもファイルを無視します。

2. ファイルが、リソース・モニター・トリガー仕様の範囲外である。以下の理由が考えられます。

- トリガー・パターンが不正である
- 不正なディレクトリーをモニターしている
- ファイル・アクセス許可が不十分である
- リモート・ファイル・システムに接続できない

1つのファイルが2回目の転送をトリガーする理由

トリガー・ファイルは、次のような理由で Managed File Transfer 転送要求を生成することがあります。

- 以前には存在しなかったトリガー・ファイルが検出された。
- トリガー・ファイルが更新され、最終変更日時が変更された。

2回目のトリガーが発生する可能性のあるシナリオには、次のようなものがあります。

- ファイルが削除されて置き換えられた。
- あるアプリケーションによってロックされていたファイルが、別のアプリケーションによってアンロックされた。
- モニター・ファイル・システムで障害が起きた。例えば、ネットワーク接続で障害が起きると、ファイルがいったん削除されてから置き換えられたように見えます。
- ファイル・ディレクトリーが別のアプリケーションによって更新されたため、最終変更日時が変更された。

エージェントのすべてのリソース・モニターに対する情報レベルの出力を設定するコマンド

V 9.0.3

この例では、特定のリソース・モニターの名前を指定していないため、すべてのリソース・モニターがモニター対象です。エージェントの名前は AGENT1 です。

```
fteSetAgentLogLevel -logMonitor=info AGENT1
```

logMonitor パラメーターの詳細、および各種オプションの使用法の例については、『[fteSetAgentLogLevel](#)』を参照してください。

エージェントのすべてのリソース・モニターに対する情報レベルの出力の例

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

```

関連資料

499 ページの『[fteSetAgentLogLevel: MFT プロトコル・ブリッジ・エージェントのログ・レベルの設定](#)』
fteSetAgentLogLevel コマンドを使用して、Managed File Transfer プロトコル・ブリッジ・エージェントと FTP/SFTP/FTPS ファイル・サーバーの間のファイル転送に関する診断情報のロギングを有効または無効にします。

キュー・リソース・モニターが開始した転送によって作成された宛先ファイルに間違ったデータが含まれる場合の対処法

リソース・モニターを作成して、キューをモニターしたり、キュー上のメッセージまたはメッセージのグループをファイルに転送したりすることができます。ファイル名は、メッセージまたはグループの最初のメッセージ上で MQMD メッセージ記述子を使用して指定することができます。メッセージからファイルへの転送が失敗し、メッセージまたはグループがキューに残されると、モニターの次の起動時に、間違ったデータが組み込まれたファイルが作成される可能性があります。

この問題が発生する理由

1. メッセージからファイルへの転送が失敗し、メッセージまたはグループがキューに残されます。
2. 新しいメッセージまたはグループがキューに入ります。
3. その新しいメッセージまたはグループがリソース・モニターを起動します。
4. リソース・モニターが、新しいメッセージまたはグループから取得した MQMD メッセージ記述子と、キューに残っていた最初のメッセージまたはグループから取得したデータを使用する新しい転送を作成します。
5. 間違ったデータが組み込まれたファイルが作成されます。

この問題を回避する方法

この問題を回避するには、**fteCreateTransfer** コマンドを使用して転送定義ファイルを手動で作成し、ファイルの <queue> エレメントを編集して属性 `groupId="${GROUPID}"` を組み込む必要があります。
fteCreateMonitor コマンドを使用して、その転送定義ファイルを送信します。

例

この例では、ソース・エージェント (モニター・エージェントでもある) は AGENT_MON で、宛先エージェントは AGENT_DEST、宛先ファイル名は `/out/files/${WMQFTEFileName}` です。この例では、メッセージに MQMD メッセージ記述子 WMQFTEFileName セットがあります。モニター対象のキューは LIVE_QUEUE です。

1. 次のコマンドを実行して、転送定義ファイルを作成します。

```

fteCreateTransfer -sa AGENT_MON -da AGENT_DEST -df "/out/files/${WMQFTEFileName}"
                 -de error -gt /tmp/TransferDefinition1.xml -sqgi -sq LIVE_QUEUE

```

転送定義ファイル `/tmp/TransferDefinition1.xml` が生成されます。

2. <queue> エlementを編集して、属性 groupId="\$\${GROUPID}"を組み込みます。行を次のように変更します。

```
<queue useGroups="true">LIVE_QUEUE</queue>
```

終了

```
<queue useGroups="true" groupId="$${GROUPID}">LIVE_QUEUE</queue>
```

この属性は、転送がキュー上の最初のグループまたはメッセージではなく、キューから転送を起動したグループまたはメッセージを読み取るために必要です。

3. 次のコマンドを実行して、モニターを作成します。

```
fteCreateMonitor -ma AGENT_MON -mq LIVE_QUEUE -mn QueueMon1 -mt /tmp/TransferDefinition1.xml  
-tr completeGroups -dv WMQFTEFileName=UNKNOWN
```

このモニターは 60 秒おきにキューをポーリングして、新規グループまたはメッセージがキューに到着しているかどうかを確認します。

宛先キューがクラスター・キューであるか、クラスター・キューの別名である場合の対処法

Managed File Transfer を使用してファイルをキューに転送するときに、クラスター・キュー (またはクラスター・キューへの別名) である宛先を使用する場合には、理由コード 2085 または 2082 を受け取ります。IBM WebSphere MQ 7.5.0 Fix Pack 4 以降、プロパティ enableClusterQueueInputOutput を true に設定すると、この問題は解決されます。

この問題が発生する理由

-dq には明示的なキュー・マネージャー名がないのに、宛先エージェントのキュー・マネージャー名が **-dq** パラメーターのキュー名に付加されています。理由コード 2085、または 2082 は、ローカル・クラスター・キューを持たないクラスター MQ queueManager への接続時に、MQOPEN 呼び出しで queueManager オブジェクトを指定できないために発生します。

この問題を回避する方法

1. キュー・マネージャー上にクラスター・キューを作成します。
2. クラスター・キューを指すリモート・キュー定義をセットアップします。

例

以下の例では、リモート・キュー定義を使用します。

構成:

- ソース・エージェント: *SAGENT*
- ソース・エージェントのキュー・マネージャー: *SQM*
- 宛先エージェント: *DAGENT*
- 宛先エージェントのキュー・マネージャー: *DQM*
- キュー・マネージャー *SQM* 上で、転送の宛先キューは *CQ6* です。

(クラスター・キュー *CQ6* が既に *SQM* に定義されていることを前提に) *SQM* 内のクラスター・キュー *CQ6* に対する、*DQM* 上の *Q6_SQM* というリモート・キュー定義を定義するには、*DQM* キュー・マネージャー上で次の MQSC コマンドを発行します。

```
define qremote(Q6_SQM) rname(CQ6) rqnname(SQM) xmitq(SQM)
```

注: rname はクラスター・キューを指します。

これで、キューに転送できます。以下に例を示します。

```
fteCreateTransfer -sa SAGENT -sm SQM -da DAGENT -dm DQM -dq Q6_SQM /tmp/single_record.txt
```

SYSTEM.MANAGED.DURABLE キューにメッセージが蓄積されたり、ファイル・システムが満杯になったりする場合の対策

IBM MQ Explorer・プラグインが調整キュー・マネージャーで永続サブスクリプションを使用すると、SYSTEM.MANAGED.DURABLE キューにメッセージが蓄積されていく可能性があります。大容量の Managed File Transfer ネットワークを使用している場合、IBM MQ Explorer プラグインを頻繁に使用するか、または両方を使用すると、このメッセージ・データがローカル・ファイル・システムに入力される可能性があります。

SYSTEM.MANAGED.DURABLE キューに蓄積されたメッセージを削除するには、以下のいずれかの操作を実行できます。

- 永続サブスクリプションを使用する IBM MQ Explorer を開始します。IBM MQ Explorer に対応した Managed File Transfer プラグインがそのキューのメッセージをコンシュームします。
- 手動でキューからメッセージを削除します。

V9.0.2

永続キューにメッセージが蓄積されないようにするには、以下のいずれかの方法を実行します。

- IBM MQ Explorer・プラグインで調整キュー・マネージャーに対する非永続サブスクリプションを使用するように指定します。IBM MQ Explorer で以下の手順を実行してください。
 1. ウィンドウ > 「設定」 > **IBM MQ Explorer > Managed File Transfer** を選択します。
 2. 「転送ログ・サブスクリプション・タイプ」リストから「NON_DURABLE」を選択します。
- IBM MQ Explorer MFT プラグインで作成された調整キュー・マネージャーから永続サブスクリプションを消去します。

永続サブスクリプションの名前には、そのサブスクリプションが IBM MQ Explorer MFT プラグインで作成されたこと、ホスト名、およびユーザーの名前を示す接頭部が付きます (例えば、MQExplorer_MFT_Plugin_HOST_TJWatson)。

関連資料

105 ページの『MFT ログ・メッセージの保持』

Managed File Transfer は、ファイル転送進行情報およびログ情報を調整キュー・マネージャーに送信します。調整キュー・マネージャーは、この情報を SYSTEM.FTE トピックに一致するサブスクリプションにパブリッシュします。サブスクリプションがない場合、この情報は保持されません。

パブリッシュ前のメッセージの検査

エージェントは IBM WebSphere MQ 6.0 キュー・マネージャーに接続できるため、エージェントは IBM WebSphere MQ 7.0 で導入された直接パブリケーション方式を使用しません。代わりに、エージェントは、MQRFH ヘッダーが含まれる通常のメッセージを調整キュー・マネージャーに送信します。MQRFH ヘッダーは、メッセージのペイロードをパブリッシュするように要求します。これらのメッセージは調整キュー・マネージャー上の SYSTEM.FTE キューに送信され、通常、メッセージはそのキューから即座にパブリッシュされます。エラー条件がこのパブリッシュを停止する場合、パブリッシュが試行される前にキュー上のメッセージを調べて、診断に役立ててください。以下のステップを実行して、診断を行うことができます。

1. 調整キュー・マネージャーでパブリッシュ/サブスクライブ・エンジンを使用不可にする。

このステップを実行するには、IBM MQ Explorer または MQSC コマンドを使用します。これにより、キュー・マネージャー上のすべてのパブリッシュ/サブスクライブ・アクティビティが一時的に停止され

る点に注意してください。調整キュー・マネージャーが他の目的でも使用されている場合、停止されるアクティビティーには、Managed File Transfer とは関係のないアクティビティーも含まれます。

IBM MQ Explorer:

- a. 「ナビゲーター」ビューで、調整キュー・マネージャーを右クリックして、「プロパティー」を選択します。
- b. 「プロパティ」のペインから、「パブリッシュ/サブスクライブ」を選択します。
- c. 「パブリッシュ/サブスクライブ・モード」リストから「互換性」を選択します。

MQSC:

```
ALTER QMGR PSMODE(COMPAT)
```

2. 別のメッセージを送信します。

パブリッシュの問題がある Managed File Transfer アクションを実行します。例えば、エージェント登録の場合、エージェントが開始される時にはいつでもメッセージが送信されます (登録メッセージを生成するために、エージェントの削除と作成を繰り返す必要はありません)。パブリッシュ/サブスクライブ・エンジンが使用不可にされているため、パブリッシュは行われません。

3. 調整キュー・マネージャー上で SYSTEM.FTE キューを参照する。

調整キュー・マネージャーの SYSTEM.FTE キューを参照するには、IBM MQ Explorer を使用する必要があります。

IBM MQ Explorer:

- a. 「ナビゲーター」ビューで、調整キュー・マネージャーを展開して、「キュー」をクリックします。「コンテンツ」ビューで、SYSTEM.FTE キューを右クリックして、「メッセージの参照 (Browse Messages)」を選択します。「メッセージ・ブラウザー」ウィンドウが開いて、パブリッシュされることになっていたメッセージが表示されます。
- b. 「ユーザー ID」列には、メッセージ記述子に含まれるユーザー ID が表示されます。パブリッシュが失敗する一般的な理由は、このユーザー ID に SYSTEM.FTE トピックに対するパブリッシュ許可がないことです。
- c. 各メッセージの詳細 (パブリッシュされることになっている XML を含む) を調べるには、メッセージを右クリックして、「プロパティー」を選択します。

メッセージの内容を検査するための MQSC コマンドはありません。IBM MQ Explorer がない場合、キューを表示して、検出されるメッセージのすべての局面を表示できる別のプログラムを使用する必要があります。**amqsbcbg** サンプル・プログラムがインストールされている場合は、トピック『[キューのブラウズ](#)』で説明されているように、このプログラムを使用できます。UserIdentifier 行にユーザー ID が示されます。あるいは、**dmpmqmsg** を使用することもできます。この場合、メッセージのユーザー ID は以下のような行で表示されます。

```
A RTM MQ24
A USR JOHNDOE
A ACC 1A0FD4D8F2F4C3C8C9D5F1F9C6F7C1C3F3F00019F7AC3000000000000000000
```

例の 2 行目は、そのメッセージのメッセージ記述子ユーザー ID です。

4. 調整キュー・マネージャーのパブリッシュ/サブスクライブ・エンジンを再度使用可能化します。

このステップを実行するには、IBM MQ Explorer または MQSC コマンドを使用します。調整キュー・マネージャーでパブリッシュ/サブスクライブ・エンジンを再度使用可能化した後、SYSTEM.FTE キューのメッセージが即座に処理されます。

IBM MQ Explorer:

- a. 「ナビゲーター」ビューで、調整キュー・マネージャーを右クリックして、「プロパティー」を選択します。

- b. 「プロパティ」のペインから、「パブリッシュ/サブスクライブ」を選択します。
- c. 「パブリッシュ/サブスクライブ・モード」リストから「**Enabled**」を選択します。

MQSC:

```
ALTER QMGR PSMODE(ENABLED)
```

IBM i IBM i 保存ファイルの転送中に起きる可能性のあるエラー

Managed File Transfer を使用して同じ IBM i 保存ファイルを複数回転送すると、転送が失敗することがあります。

Managed File Transfer で、次のエラーのいずれかまたはその両方が起きている可能性があります。

- 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"

これらのエラーは、同じ IBM i 保存ファイルを転送する要求を MFT エージェントに対して同時に複数発行した場合に発生することがあります。同じ保存ファイルを同時に複数回転送するには、複数のソース・エージェントを使用する必要があります。並行して実行される転送それぞれに対して別々のソース・エージェントを使用してください。

同じ保存ファイルを1つのソース・エージェントで複数回転送するには、前の転送要求が完了するまで待ち、完了してから次の新規転送要求を実行依頼しなければなりません。

V 9.0.1 再配布可能 MFT エージェントの構成時に起こる可能性のあるエラー

Redistributable Managed File Transfer Agent の構成時のエラー・メッセージ

Windows 用ネイティブ・ライブラリーをロードできない

Windows Redistributable Managed File Transfer Agent を使用するには、次の Microsoft ライブラリーをシステムにインストールする必要があります。

- Microsoft Visual C++ 再頒布可能パッケージ 2008
- Microsoft Visual C++ 再頒布可能パッケージ 2012

これらのライブラリーは、Microsoft から入手できます。[The latest supported Visual C++ downloads](#) を参照してください。

これらのライブラリーがインストールされていない状態で MFT コマンドを実行しようとする、次のエラーが報告されます。

- 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.

Microsoft ライブラリーがインストールされていることを確認してください。ライブラリーがインストールされていない場合は、インストールしてからコマンドを再実行してください。

バインディング・モードの使用はサポートされない

Redistributable Managed File Transfer Agent は、クライアント・モードでのみ IBM MQ に接続できます。バインディング・モードでコマンドを実行しようとする、次のエラーが報告されます。

- 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

コマンドを発行するときには、キュー・マネージャーのホスト、ポート、名前、およびチャンネル名を指定する必要があります。

メッセージ・サイズに関連する MQ 属性および MFT プロパティを設定する際のガイダンス

さまざまなサイズのメッセージを読み書きする際の Managed File Transfer の動作に影響を与えるように、IBM MQ 属性および Managed File Transfer プロパティを変更することができます。

ソース・キューから読み取るメッセージまたは宛先キューに書き込むメッセージのサイズが 1048576 バイト (1 MB) を超す場合は、Managed File Transfer Agent プロパティ **maxInputOutputMessageLength** の値を、読み書きする最大メッセージ・サイズ以上の値に増やす必要があります。

ソース・キューのメッセージが 1048576 バイトより大きい場合は、ソース・エージェントに **maxInputOutputMessageLength** プロパティを設定する必要があります。宛先キューのメッセージが 1048576 バイトより大きい場合は、宛先エージェントに **maxInputOutputMessageLength** プロパティを設定する必要があります。 **maxInputOutputMessageLength** プロパティについては、[拡張エージェント・プロパティ](#)を参照してください。

- エージェントが読み書きするキューがエージェント・キュー・マネージャーに対してローカルである場合、IBM MQ キュー・マネージャー、キュー、およびチャンネルの **MAXMSGL** 属性の変更が必要な場合があります。

ソース・キューまたは宛先キューの最大メッセージ・サイズの値が、**maxInputOutputMessageLength** エージェント・プロパティの値以上であることを確認します。

以下の IBM MQ 属性の各値 (バイト単位) について、

- エージェント・キュー・マネージャーの最大メッセージ・サイズ
- `SYSTEM.FTE.STATE.agent_name` キューの最大メッセージ・サイズ
- クライアント・チャンネルの最大メッセージ・サイズ (使用中のエージェントがクライアント・モードでキュー・マネージャーに接続している場合)

次の計算の結果以上であることを確認します。

ファイルからメッセージへの転送 (サポートされるファイル・サイズの最大値は 100 MB) の場合:
maxInputOutputMessageLength の値

メッセージからファイルへの転送の場合:

$3 * (\text{maxInputOutputMessageLength}) + 1048576$ の値

(この計算は、1つの状態メッセージに3つのチェックポイントを保管でき、各チェックポイントはデータのメッセージ量の最大サイズまでバッファに入れる必要があるかもしれない、という事実から導き出されたものです。)

- エージェントの書き込み先のキューがリモート・キューである場合、IBM MQ キュー・マネージャー、キュー、およびチャンネルの **MAXMSGL** 属性の変更が必要な場合があります。

以下の各 IBM MQ 属性の値が、**maxInputOutputMessageLength** エージェント・プロパティの値以上であることを確認します。

- エージェント・キュー・マネージャーのリモート・キュー・マネージャー伝送キューの最大メッセージ・サイズ
- エージェント・キュー・マネージャーからリモート・キュー・マネージャーへのチャンネルの最大メッセージ・サイズ
- リモート・キュー・マネージャーの宛先キューの最大メッセージ・サイズ
- リモート・キュー・マネージャーの最大メッセージ・サイズ

以下の IBM MQ 属性の各値 (バイト単位) について、

- エージェント・キュー・マネージャーの最大メッセージ・サイズ
- `SYSTEM.FTE.STATE.agent_name` キューの最大メッセージ・サイズ
- クライアント・チャネルの最大メッセージ・サイズ (使用中のエージェントがクライアント・モードでキュー・マネージャーに接続している場合)

次の計算の結果以上であることを確認します。

ファイルからメッセージへの転送 (サポートされるファイル・サイズの最大値は **100 MB**) の場合:
`maxInputOutputMessageLength` の値

メッセージからファイルへの転送の場合:

$3 * (\text{maxInputOutputMessageLength}) + 1048576$ の値

(この計算は、1つの状態メッセージに3つのチェックポイントを保管でき、各チェックポイントはデータのメッセージ量の最大サイズまでバッファーに入れる必要があるかもしれない、という事実から導き出されたものです。)

上記のプロパティのいずれかの値を超えた場合、エージェントは停止し、次のエラーがエージェント・イベント・ログに記録されます。

```
BFGUT0002E: An internal error has occurred. Product failure data was captured in file
"FFDC.FTE.20100928170828514.8172766022149157013.log".
BFGSS0025E: An internal error has occurred. The exception is: cc=2 rc=2010 op=put - MQPUT to
SYSTEM.FTE.STATE.agent_name
BFGAG0061E: The agent ended abnormally
```

以下の IBM MQ 理由コードがエージェント・イベント・ログのこのメッセージに含まれている場合があります。

- **rc=2010** この理由コードは `MQRC_DATA_LENGTH_ERROR` に対応し、クライアント・チャネルの最大メッセージ・サイズの値を超えたことを示します。この問題を解決するには、エージェント・キュー・マネージャーのクライアント・チャネルの最大メッセージ・サイズが次の計算の結果以上であることを確認します。

```
3 * (maxInputOutputMessageLength) + 1048576
```

- **rc=2030** この理由コードは `MQRC_MSG_TOO_BIG_FOR_Q` に対応し、`SYSTEM.FTE.STATE.agent_name` キューの最大メッセージ・サイズの値を超えたことを示します。この問題を解決するには、`SYSTEM.FTE.STATE.agent_name` キューの最大メッセージ・サイズが次の計算の結果以上であることを確認します。

```
3 * (maxInputOutputMessageLength) + 1048576
```

- **rc=2031** この理由コードは `MQRC_MSG_TOO_BIG_FOR_Q_MGR` に対応し、エージェント・キュー・マネージャーの最大メッセージ・サイズの値を超えたことを示します。この問題を解決するには、エージェント・キュー・マネージャーの最大メッセージ・サイズが次の計算の結果以上であることを確認します。

```
3 * (maxInputOutputMessageLength) + 1048576
```

小さいメッセージを多数転送する場合

エージェントがキューとの間で読み書きするメッセージの平均サイズが 1310 バイト未満であり、エージェントが 10000 を超えるメッセージを読み書きする場合、キュー・マネージャーの非コミット・メッセージの最大数属性を大きくするか、チェックポイント間隔のデータ量を削減する必要があります。

エージェントがキューとの間でメッセージを読み書きする場合、対応する **GET** または **PUT** は、まとめてトランザクションにグループ化されます。トランザクション内の **GET** または **PUT** の数は、チェックポイント間隔内のすべてのデータを処理するために必要な数によって決定されます。チェックポイント間隔のデータの概算の量は、次の計算を使用して、エージェント・プロパティから決定されます。

```
Checkpoint interval data size (in bytes) = agentCheckpointInterval * agentFrameSize *
                                           agentWindowSize * agentChunkSize.
```

デフォルトのチェックポイント・データ・サイズは、 $1 * 5 * 10 * 262144$ バイト = 13107200 バイト (12.5 MB) です。キュー・マネージャーがサポートする、トランザクションの非コミット・メッセージの最大数は、**MaxUncommittedMsgs** キュー・マネージャー属性によって制御されます。この属性のデフォルト値は 10000 メッセージです。平均メッセージ・サイズが約 1310 バイトより小さい場合、10000 を超すメッセージが書き込まれると、非コミット・メッセージのデフォルトの最大数を超えます。

MaxUncommittedMsgs の制限を超えた場合、エージェントは停止し、次のエラーがエージェント・イベント・ログに記録されます。

```
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.
```

理由コード 2024 は MQRC_SYNCPOINT_LIMIT_REACHED に対応します。

この問題を解決するには、以下のいずれかの処置を行います。

- キューとの間で読み書きしているエージェントが接続するキュー・マネージャーの **MaxUncommittedMsgs** キュー・マネージャー属性の値を増やします。 [MaxUncommittedMsgs \(MQLONG\)](#) を参照してください。
- チェックポイント間隔のデータ量を削減します。削減のためには、以下の 1 つ以上のエージェント・プロパティの値を減らします。
 - agentCheckpointInterval
 - agentFrameSize
 - agentWindowSize
 - agentChunkSize

これらのエージェント・プロパティについて詳しくは、[拡張エージェント・プロパティ](#) を参照してください。

メッセージを永続的にキューに書き込む場合

メッセージをキューに転送し、永続的にキューに書き込む場合は、チェックポイント間隔のすべてのデータをログに記録できるように、キュー・マネージャーのログ・ファイル・スペースのサイズを増やさなければならない場合があります。

キュー・マネージャーのログ・ファイル・スペースを超えた場合、エージェントは停止し、次のエラーがエージェント・イベント・ログに記録されます。

```
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
```

理由コード '2102' は MQRC_RESOURCE_PROBLEM に対応します。

この問題を解決するには、宛先エージェント・キュー・マネージャーのログ・ファイル・スペースのサイズを増やします。

MFT で UAC と仮想ストアを使用する場合の注意点

Windows Server 2008 R2 などのオペレーティング・システムには、ユーザー・アカウント制御 (UAC) 機能があります。このセキュリティ・インフラストラクチャーは、その機能の 1 つとして、中央の Program

Files ディレクトリーに保管されたユーザー・データをユーザー・ロケーション (仮想ストアと呼ばれる) に転送します。

Managed File Transfer ツールのみを使用してデータ構造を管理する場合、Managed File Transfer は UAC や仮想ストアの影響を受けません。ただし、IBM MQ の管理者以外が標準オペレーティング・システム・ツールを使用してディレクトリー構造を変更または再ビルドした場合は、新しい構造が仮想ストアに転送される可能性があります。これにより、以下の 1 つ以上の状態が発生することがあります。

- ユーザー (IBM MQ 管理者を含む) が、予期される場所でファイルを見つけることができない。
- エージェントを開始できず、メッセージ BFGCL0315 が報告されるが、対応する理由コードが提供されない。
- エージェントから報告された場所にログ・ファイルがない。
- **-F** パラメーターを使用してエージェントが開始されていない可能性があれば、次のメッセージが報告されます。

```
The current directory is invalid
```

これらのすべての状態を修正するには、次のようにします。

- IBM MQ 管理者の場合は、**fteDeleteAgent** コマンドおよび **fteCreateAgent** コマンドを使用して、エージェント構造を再ビルドします。
- オペレーティング・システム管理者の場合は、影響を受けたユーザーの仮想ストア内の IBM MQ エントリーを削除します。例えば、Windows では、仮想ストアのロケーションは次のようになります：
%USERPROFILE%\AppData\Local\VirtualStore\

関連資料

[462 ページの『fteDeleteAgent』](#)

fteDeleteAgent コマンドは、Managed File Transfer Agent とその構成を削除します。エージェントがプロトコル・ブリッジ・エージェントの場合は、ユーザー資格情報ファイルがファイル・システムに残ります。

[388 ページの『fteCreateAgent \(MFT エージェントの作成\)』](#)

fteCreateAgent コマンドは、Managed File Transfer Agent とその関連構成を作成します。

Windows MFT エージェントまたはロガーを Windows サービスとして実行するためのガイダンス

Managed File Transfer エージェント、スタンドアロン・データベース・ロガー、およびスタンドアロン・ファイル・ロガーを Windows サービスとして実行することができます。これらの Windows サービスに問題がある場合、サービス・ログ・ファイルおよびこのトピックの情報をを使用して、問題を診断することができます。

エージェント、スタンドアロン・ロガー、またはスタンドアロン・ファイル・ロガーを Windows サービスとして実行するように構成する方法については、[155 ページの『Windows サービスとしての MFT エージェントの開始』](#) および [492 ページの『fteModifyLogger \(Windows サービスとしての MFT ロガーの実行\)』](#) を参照してください。

注: 再配布可能エージェントが Windows サービスとして実行される場合、このサービスを機能させるには、システム環境で **BFG_DATA** 環境変数を設定する必要があります。

ログ・ファイルの場所

fteCreateAgent、**fteCreateCDAgent**、**fteCreateBridgeAgent**、**fteModifyAgent**、**FteCreateLogger**、または **fteModifyLogger** コマンドを使用してエージェントまたはロガーを Windows サービスとして実行する場合、**-sl** パラメーターを使用してロギングのレベルを選択できます。

このパラメーターに指定できる値は、`error`、`info`、`warn`、および `debug` です。デフォルト値は `info` です。

Windows サービスのログ・ファイルには、ファイル名 `servicedate.log` が含まれ、`date` はサービスが開始した日付です。エージェントのファイルは、ディレクトリー `MQ_DATA_PATH\mqft\logs\coordination_qmgr_name\agents\agent_name` に書き込まれます。このディレクトリーは、Managed File Transfer Agent トレース・ファイルが書き込まれるディレクトリーと同じです。ローガーのファイルは、ディレクトリー `MQ_DATA_PATH\mqft\logs\coordination_qmgr_name\loggers\logger_name` に書き込まれます。

エージェントまたはスタンドアロン・ローガーを Windows サービスとして開始しようとして問題がある場合は、`-sl` パラメーターを使用してロギング・レベルを `debug` に設定してみてください。追加情報は `servicedate.log` ファイルに書き込まれます。

注：ロギング・レベルが `debug` に設定されている場合、Windows サービスの実行に使用するユーザー・アカウントおよびパスワードがログ・ファイルにプレーン・テキストで表示されます。

ログ・ファイルの数

`fteCreateAgent`、`fteCreateCD` エージェント、`fteCreateBridgeAgent`、`fteModifyAgent`、`FteCreateLogger`、または `fteModifyLogger` コマンドを使用して、エージェントまたはスタンドアロン・ローガーを Windows サービスとして実行する場合、`-sj` パラメーターを使用してログ・ファイルの数を選択できます。ログ・ファイルの数を変更するには、コマンドの一部として `-sj -Dcom.ibm.wmqfte.daemon.windows.windowsServiceLogFiles=number` というテキストを指定します。ここで、`number` は、正整数で表されるログ・ファイルの数です。ログ・ファイルの数を指定しない場合、デフォルトは 5 です。

「サービスとしてログオン」権限

サービスを実行するために使用する Windows アカウントには、**Log on as a service** 権限が必要です。`fteStartAgent`、`fteStartLogger` コマンド、または Windows の `Sc.exe` コマンドのいずれかを使用してサービスを開始しようとしたが、この権限を持たないユーザー・アカウントを使用している場合、「サービス」ウィンドウが開きます。開始しようとしたサービスがエージェントを実行する場合は、このウィンドウには次のメッセージが含まれます。

```
Unable to start Windows service mqmftAgentAGENT@QMGR.  
System error 1069: The service did not start due to a logon failure.
```

このメッセージでは、`AGENT` はエージェント名、`QMGR` はエージェント・キュー・マネージャー名です。スタンドアロン・ローガーをサービスとして実行しようとする場合、同様のメッセージが生成され、エージェントではなくローガーを参照します。

このエラーを回避するには、サービスを実行するために使用する Windows アカウントに「**Log on as a service**」権限を付与します。例えば、Windows 7 では以下のステップを実行します。

1. 「スタート」メニューから、「管理ツール」 > 「ローカルセキュリティポリシー」をクリックします。
2. 「セキュリティの設定」ペインで、「ローカルポリシー」を展開し、「ユーザー権利の割り当て」をクリックします。
3. 「Policy and Security Setting」ペインで、「サービスとしてログオン」をダブルクリックします。
4. ユーザーまたはグループの追加をクリックしてから、サービスを実行するユーザーを、**Log on as a service** 権限を持つユーザーのリストに追加します。ユーザ名は、`fteCreateAgent`、`fteCreateCDAgent`、`fteCreateBridgeAgent`、`fteModifyAgent`、`fteCreateLogger`、または `fteModifyLogger` コマンドを使用するときに提供される。

注：エラー System error 1069: The service did not start due to a logon failure. は、パスワードの誤りが原因である可能性もあります。

Windows アカウント・パスワードの非表示

エージェントまたはスタンドアロン・ロガーを Windows サービスとして実行するように構成するときに、使用するユーザー名とパスワードを指定します。次の例では、エージェント・キュー・マネージャー QMGR1 を含むエージェント AGENT1 が作成され、Windows サービスとして実行するように構成されます。

```
fteCreateAgent -agentName AGENT1 -agentQMGr QMGR1 -s -su fteuser -sp ftepassword
```

この例では、Windows サービスは、ユーザー名 `fteuser` とその関連パスワード `ftepassword` を使用して実行されます。**fteCreateAgent** コマンド、または **-s** パラメーターを受け入れる他のコマンドの 1 つを実行する場合、Windows アカウントのパスワードをプレーン・テキストで指定します。パスワードを表示したくない場合は、次の手順を実行します。

1. **-sp** パラメーターを指定せずに、コマンド (**fteCreateAgent**、**fteCreateCD エージェント**、**fteCreateBridgeAgent**、**fteModifyAgent**、**FteCreateLogger**、または **fteModifyLogger**) を実行します。以下に例を示します。

```
fteCreateAgent -agentName AGENT1 -agentQMGr QMGR1 -s -su fteuser
```

注: このコマンドは、サービスを正常に開始するには、Windows サービス・ツールを使用してパスワードを設定する必要があることを警告するメッセージを生成します。

2. Windows の「サービス」ウィンドウを開きます。
3. サービスのリストで、エージェントまたはスタンドアロン・ロガー・サービスを右クリックし、「プロパティ」を選択します。エージェント・サービスの表示名は Managed File Transfer Agent AGENT @ QMGR です。ここで、AGENT はエージェント名、QMGR はエージェント・キュー・マネージャー名です。ロガー・サービスの表示名は Managed File Transfer Logger for property set coordination_qmgr_name です。ここで、調整キュー・マネージャー名は、プロパティー・セットとして使用するスタンドアロン・ロガーに指定した調整キュー・マネージャーです。プロパティー・セットについて詳しくは、530 ページの『fteStartLogger』および 492 ページの『fteModifyLogger (Windows サービスとしての MFT ロガーの実行)』を参照してください。
4. 「プロパティ」ウィンドウで、「ログオン」タブを選択します。
5. サービスを実行するユーザー・アカウントのパスワードを、「パスワード」と「パスワードの確認入力」フィールドに入力します。パスワード文字は、入力すると非表示になります。
6. 「OK」をクリックします。

既知の問題

JAVA_HOME システム環境変数の使用に関する問題 (IBM WebSphere MQ 7.5.0 Fix Pack 1 以前の Managed File Transfer にのみ該当)。

JAVA_HOME システム環境変数を設定してはなりません。設定すると、エージェントまたはロガー Windows Service が開始されない場合があります。エージェントまたはロガー Windows Service は、IBM MQ Java ランタイムを使用して実行する必要があります。

Windows エージェントまたはロガーの JVM オプションの更新に関する注意点

fteModifyAgent または **fteModifyLogger** コマンドの **-sj** パラメーターを使用して、Java システム・プロパティーを更新、追加、または削除することにより、エージェントまたはロガーの既存の Windows サービス定義を変更する場合、既存の Windows サービスは、代わりに新規サービスが作成される前に最初に削除され、エージェントまたはロガーのプロパティー・ファイルが新規 Windows サービスのプロパティーで更新されます。新しい Windows サービス定義は、エージェントまたはロガーのプロパティー・ファイルで定義されている更新後の Windows サービス・プロパティーと整合していなければなりません。

IBM MQ 9.0.0 Fix Pack 4 以降、**fteModifyAgent** コマンドまたは **fteModifyLogger** コマンドの **-sj** パラメーターを使用してエージェントまたはロガーの JVM オプションに対して行われた更新が、オプションが正しく指定されていることを確認するために検証されるように、APAR IT22423 で追加のチェックが追加さ

れました。プロパティが無効だったり、妥当性を検証できなかったりした場合は、**fteModifyAgent** または **fteModifyLogger** コマンドが失敗し、該当するエラー・メッセージが表示されます。

JVM プロパティが有効で、既存の Windows サービスの削除が成功したとしても、**fteModifyAgent** コマンドまたは **fteModifyLogger** コマンドによって新しい Windows サービスを作成する時に障害が発生した場合は、そのコマンドが、エージェントまたはログャーのプロパティ・ファイルから、新しい代わりの Windows サービスを定義したプロパティを削除しようとしています。その場合は、エラー・メッセージが返されます。エージェントまたはログャーが変更されなかったことや、古い Windows サービスは削除されたものの、新しい Windows サービスが作成されなかったため、エージェントまたはログャーを Windows サービスとして実行できないことを説明したエラー・メッセージです。Windows サービス定義の状態が、エージェントまたはログャーのプロパティ・ファイルで定義されている Windows サービス・プロパティと整合していることを手動で確認し、不整合があれば適切な処置を実行してください。

関連情報

[fteModifyAgent: Windows サービスとしての MFT エージェントの実行](#)

[fteModifyLogger: Windows サービスとしての MFT ログャーの実行](#)

エージェントの過負荷を回避するように MFT リソース・モニターを構成するためのガイダンス

エージェントの負荷を軽減するように、Managed File Transfer リソース・モニターのプロパティおよびパラメーター値を構成できます。エージェントの負荷を軽減すると、そのエージェントのパフォーマンスが向上します。使用できる設定はいくつかあります。現在のシステム構成に最適な設定を見つけ出すために、試行錯誤が必要になる場合があります。

リソース・モニターの概要

リソース・モニターでディレクトリーまたはキューをポーリングするときには、エージェントによって以下のステージが実行されます。

- トリガー・パターンに一致するすべてのファイル (例えば、ディレクトリー内のすべての *.txt ファイル) を検出します。または、キュー上の完全に揃ったメッセージ・グループをすべて検出します。
- ファイルが新規であるか変更済みであるかを判別します。または、キュー上の新規グループを判別します。
- 前の 2 つの段階で基準に一致したファイルまたはグループの転送を開始します。
- 既に転送されているファイルおよびグループのリストに追加します。これにより、変更が加えられるまで再度転送されることがなくなります。

ディレクトリー・モニターの場合、ソース・ディレクトリー内のファイル数が多くなり、トリガー・パターンが広がるほど、エージェントが構文解析し、既に転送されたファイルのリストと比較して比較する必要があるファイルのリストが大きくなります。

キュー・モニターの場合は、キュー上のグループが増えると、エージェントで転送済みのグループのリストと比較する必要があるグループのリストが増えます。

考慮する必要がある主要な設定

- エージェント・プロパティ **monitorMaxResourcesInPoll** を使用して、エージェントで各ポーリングに含めるファイルまたはグループの最大数を設定します。このパラメーターを使用すると、ポーリング間隔における転送数が制限されます。また、エージェントでその数のファイルまたはグループの転送を開始する前に実行しなければならない解析の量が削減されます。ディレクトリー・モニターまたはキュー・モニターでの次回ポーリング時には、エージェントの対象が、その次のセットのファイルまたはグループに移ります。エージェント・プロパティ **monitorMaxResourcesInPoll** は IBM WebSphere MQ File Transfer Edition 7.0.4 以降で使用可能です。以前のバージョンの IBM WebSphere MQ File Transfer Edition の場合は、APAR IC78011 の暫定修正として入手できます。
- ディレクトリー・モニターを作成するときには、構成する転送定義に **delete** のソースの後処理が含まれていることを確認してください。この後処理を設定すると、ファイル転送が完了したときに、ファイルがモニター対象のディレクトリーから削除され、エージェントの内部リストに保持されなくなります。

- ディレクトリー・モニターを作成するときには、**fteCreateMonitor** コマンドで **-rl** パラメーターを使用して、エージェントで繰り返す必要のあるディレクトリーのレベル数を制限します。このパラメーターを使用すると、下位のディレクトリーが不要にスキャンされることがなくなります。

リソース・モニター作成時の追加の考慮事項

リソース・モニターのポーリング・プロセスでは、エージェント・リソースを消費します。モニターのポーリング間隔が長くなると、エージェントに課される負荷が軽減されます。ただし、ポーリング間隔ごとに生成される転送の数が多くなりすぎないように、ポーリング間隔の設定を適切にする必要があります。リソース・モニターのポーリング間隔を設定するときには、以下について考慮してください。

- ファイルがディレクトリーまたはキュー上のグループに配置された後、どのようなタイミングで転送を開始する必要があるか。
- ディレクトリーまたはキュー上のグループにファイルを配置する速度。
- エージェントの最大転送速度。エージェントは、モニターによって生成されるすべての転送を処理できる必要があります。

ポーリング間隔はリソース・モニターを作成するとき指定します。リソース・モニター作成コマンド **fteCreateMonitor** で **-pi** パラメーター (ポーリング間隔) および **-pu** パラメーター (ポーリング間隔単位) を指定します。現在の構成に最適な設定を判別するために、試行を繰り返すことが必要な場合があります。

リソース・モニターを実行するエージェントの負荷が大きい場合、その安定度を高めるためのオプションでは、**maxSourceTransfers** のエージェント・プロパティー値を小さくします。このオプションを使用すると、エージェントでは、リソース・モニターの処理時間とファイル転送の処理時間が分けられます。エージェント・プロパティー **maxSourceTransfers** の値を大きくすると、ファイル転送の処理時間が長くなり、リソース・モニターの処理時間が短くなります。エージェント・プロパティー **maxSourceTransfers** の値を小さくすると、エージェントで並行して行われる転送の数が少なくなりますが、リソース・モニターをポーリングするための処理時間が十分ある必要があります。このエージェント・プロパティーの値を小さくする場合は、キューに入れられる転送の数が増える可能性があるため、エージェント・プロパティー **maxQueuedTransfers** の値を大きくすることを考慮してください。

モニターを最適化した後、一部の転送がリカバリー状態になったら、エージェントのタイムアウト値を大きくすることを考慮してください。エージェントの負荷が大きいと、宛先エージェントに対して転送の開始をネゴシエーションするときに、転送がタイムアウトになる可能性があります。このタイムアウトが起きると、転送がリカバリー状態になり、転送の完了が遅延します。エージェント・プロパティー **maxTransferNegotiationTime** では、ソース・エージェントが宛先エージェントからの応答を待機する時間を指定します。この時間を超過すると、転送がリカバリー状態になります。このプロパティーのデフォルト値は 30000 ミリ秒 (30 秒) です。プロパティーの値を大きくすると (例えば、300000 ミリ秒 (5 分) に設定すると)、タイムアウトになることなく転送を続行でき、リカバリー状態になりません。

関連概念

[161 ページの『転送定義ファイルの使用』](#)

ファイル転送を作成するために使用できる転送定義ファイルを指定できます。転送定義ファイルは、転送を作成するために必要な情報の一部またはすべてを定義した XML ファイルです。

関連タスク

[170 ページの『MFT リソースのモニター』](#)

キューやディレクトリーなどの Managed File Transfer リソースをモニターできます。そのリソースで条件が満たされると、リソース・モニターがファイル転送などのタスクを開始します。 **fteCreateMonitor** コマンドを使用するか、IBM MQ Explorer に対応した Managed File Transfer プラグインの「モニター」ビューを使用して、リソース・モニターを作成できます。

関連資料

[413 ページの『fteCreateMonitor: MFT リソース・モニターの作成』](#)

fteCreateMonitor コマンドは、コマンド行から新規リソース・モニターを作成して開始します。

Managed File Transfer を使用してリソース (例えば、ディレクトリーの内容) をモニターして、トリガー条件が満たされたときに、指定したタスク (ファイル転送など) を開始することができます。

変数置換によって複数のファイルが1つのファイル名に送られる場合の対応策

Managed File Transfer で、ディレクトリーをモニターして複数のファイルをソースから宛先の場所に転送する時に、`${FileName}` の変数置換を使用する場合は、変数置換の結果をテストする必要があります。変数置換をテストしなければならないのは、変数置換の結果として、予期しない組み合わせのファイル転送コマンドが呼び出されることがあるからです。

問題が発生しているかどうかを確認するために、複数のファイルが転送されているのに、宛先に届いているファイルが1つしかない、というケースを探してください。複数のファイルを同じ宛先の場所(同じファイル名)に転送しようとして失敗したことを示すエラーがファイル転送ログに書き込まれているかもしれません。

この問題が発生する理由

MFT ディレクトリー・モニターによって複数のファイルが処理される場合は、モニター対象のディレクトリーで検出されたすべてのファイルに対してタスク XML が実行されます。`${FileName}` が XML タスク・ファイルの宛先だけで指定され、ソースでは指定されていない場合は、ファイルごとに転送が複数回(ファイル名の組み合わせごとに1回ずつ)呼び出されます。

以下に例を示します。

```
<source disposition="delete" recursive="false">
  <file>e:\temp</file>
</source>
<destination exist="overwrite" type="file">
  <file>s:\outdir\${FileName}</file>
</destination>
```

この問題を回避する方法

`${FileName}` の変数置換をソースや宛先で使用する時に、同じファイル名のバリエーションが1つの宛先に届くようにしたい場合は、タスク XML 定義のソースと宛先の両方で `${FileName}` を指定してください。

以下の例では、`e:\temp\<filename>` からファイルを取り出し、それを `s:\outdir\<filename>.out` に転送します。

```
<source disposition="delete" recursive="false">
  <file>e:\temp\${FileName}</file>
</source>
<destination exist="overwrite" type="file">
  <file>s:\outdir\${FileName}.out</file>
</destination>
```

関連概念

[180 ページの『変数置換を使用した MFT リソース・モニター・タスクのカスタマイズ』](#)

アクティブなリソース・モニターのトリガー条件が満たされると、定義されたタスクが呼び出されます。毎回同じ宛先エージェントまたは同じ宛先ファイル名を使用して転送またはコマンド・タスクを呼び出すことができますが、実行時にタスク定義を変更することもできます。これは、タスク定義 XML に変数名を挿入することで行います。トリガー条件が満たされているとモニターが判断し、タスク定義に変数名が含まれている場合は、変数名を変数値と置換してから、タスクを呼び出します。

[183 ページの『例: リソース・モニター定義の変数置換』](#)

XML と IBM MQ Explorer を使用したリソース・モニター定義の変数置換の例。

Oracle データベースで MFT データベース・スキーマを更新する時にエラーが発生した場合の対応策

`ftelog_tables_oracle_702_703.sql` ファイルを使用してデータベース・スキーマを最新レベルに更新すると、`ERROR at line 1: ORA-02289: sequence does not exist` というエラー・メッセージが表示される場合があります。このエラーは、表で使用されるシーケンスおよびトリガーのスキーマが表のスキーマと異なるために発生します。

このタスクについて

この問題を修正するには、`ftelog_tables_oracle_702_703.sql` を実行する前に、その内容を編集する必要があります。

手順

1. Managed File Transfer データベース・ロガー表で使用されるシーケンスおよびトリガーが、どのスキーマに配置されているかを調べます。
 - Db2 では、コントロール・センターを使用して表およびスキーマを表示することができます。
 - Oracle では、Enterprise Manager を使用して表およびスキーマを表示することができます。
2. `ftelog_tables_oracle_702_703.sql` ファイルをテキスト・エディターで開きます。
3. `SELECT FTELOG.sequence_name.nextval` というテキストが出現するすべての箇所で、テキスト `FTELOG` を既存のシーケンスが属するスキーマの名前に置き換えます。
4. テキスト `CREATE OR REPLACE TRIGGER FTELOG.trigger_name` の各出現箇所の前に、テキスト `DROP TRIGGER schema_name.trigger_name` を挿入します。ここで、`schema_name` は既存のトリガーが配置されているスキーマの名前です。
5. 編集した `ftelog_tables_oracle_702_703.sql` ファイルを使用してデータベース表を更新します。

MFT ロガーのエラー処理とリジェクト

Managed File Transfer ロガーは、メッセージごとのエラーと一般エラーの 2 つのタイプのエラーを特定します。

エラーは、1 つまたは少数の個々のメッセージの問題によって発生する可能性があります。メッセージごとのエラーとして識別される状況の例として次のものがあります。

- データの必須項目である結果コードがメッセージから欠落している。
- 転送で長さ 3000 文字のジョブ名が指定されたが、関連付けられているデータベース列に大きすぎて入らない
- 転送で進行中メッセージを受信したが、開始された転送のレコードがない (恐らく転送開始メッセージが誤った経路でルーティングされたか遅延しているため)
- メッセージを受信したが、Managed File Transfer ログ・メッセージではない

一般エラーは、メッセージごとのエラー以外のこれらすべてのエラーです。これらのエラーは、構成上の問題やプログラム・エラーが原因で発生する可能性があります。

ロガーは、メッセージごとのエラーを検出すると、リジェクト・キューにメッセージを置くことによりメッセージをリジェクトします。出力ログには何も書き込まれないため、定期的にはリジェクト・キューを検査するか、継続的にモニターしてリジェクトされたメッセージを検出してください。

データベースにメッセージが正常に書き込まれずに、あまりに多くのメッセージが連続してリジェクトされる場合は、一般エラーとして扱われます。例えば、ジョブ名として常に 10 文字コードを使用するが、間違えてジョブ名の列を 2 文字幅に再構成してしまったサイトを考えてみましょう。幅が大きすぎるデータは通常メッセージごとのエラーとなりますが、この場合、構成上の問題は一般的であるため、一般エラーとして検出されます。**`wmqfte.max.consecutive.reject`** プロパティを使用すると、一般エラーとみなされるまでに必要な連続メッセージごとのエラーの数を調整できます。

一般エラーが検出されると、ロガーはまだキュー・マネージャーにコミットされていないすべてのメッセージをロールバックし、定期的には再試行します。ロガーを **`-F`** パラメーターを指定してフォアグラウンド・モードで開始した場合、問題を識別するメッセージが出力ログおよびコンソールに書き込まれます。

ロガーの出力ログの場所は、そのロガーがスタンドアロンか JEE データベース・ロガーかによって異なります。スタンドアロン・データベース・ロガーの場合は、`MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name` ディレクトリー内にあります。JEE データベース・ロガーの場合は、そのアプリケーション・サーバーの標準出力ログに含まれます。

リジェクト・キュー

メッセージごとのエラーの原因となったメッセージはリジェクト・キューに移動されます。リジェクトされたそれぞれのメッセージでは、メッセージがリジェクトされた理由を示すためにメッセージ・プロパティが設定されています。プロパティのフルネームは **usr.WMQFTE_ReasonForRejection** ですが、usr です。は、一部のコンテキスト (JMS および IBM MQ Explorer を含む) では省略されます。

IBM MQ Explorer を使用している場合は、リジェクト・キューを右クリックして「**メッセージの参照 (Browse Messages)**」をクリックすると、その内容を表示できます。メッセージがリジェクトされた理由を表示するには、メッセージをダブルクリックしてプロパティ・ダイアログを開き、「**指定したプロパティ (Named Properties)**」ページを選択します。 **WMQFTE_ReasonForRejection** という名前のプロパティがあります。別の方法として、この情報を自動的に取得するようにモニタリング・ツールを作成または構成することもできます。

場合によっては、リジェクト・キューのメッセージを再処理する場合があります。このトピックで前述したデータベース内の 2 文字のジョブ名列の例では、データベース列の幅を大きくすれば、メッセージを正常に処理できます。別の例として、transfer-complete メッセージが、関連付けられている transfer-start メッセージが欠落しているためにリジェクトされた場合、transfer-start メッセージが後で着信する場合があります。transfer-complete を再処理すると、成功します。

メッセージを再処理するには、メッセージをリジェクト・キューから入力キューに移動します。通常インストールで、ロガーが独自の管理対象サブスクリプションを作成した場合、入力キューはキュー・マネージャーによって定義され、名前が SYSTEM.MANAGED.DURABLE.49998CFF20006204 のようになります。入力キューを識別するには、サブスクリプション SYSTEM.FTE.DATABASELogger.AUTO のプロパティで「**宛先名 (Destination name)**」を確認するか、次の MQSC コマンドを使用します。

```
DISPLAY SUB(SYSTEM.FTE.DATABASELogger.AUTO) DEST
```

メッセージをキュー間で移動する方法の 1 つは、[MA01 SupportPac](#) を使用することです。例:

```
q -IFTE.REJECT -oSYSTEM.MANAGED.DURABLE.49998CFF20006204
```

リジェクト・キューには、さまざまな理由でリジェクトされたメッセージが含まれている場合があります。解決されているのは一部のみです。この場合はまだすべてのメッセージを再処理でき、現在、受け入れ可能なメッセージは消費されます。受け入れ不可能なメッセージは再度リジェクト・キューに移動されます。

転送ログ内の Malformed ログ・メッセージは、ロガーによって記録されません。これらのメッセージは重要であるとみなされないため、リジェクト・キューに送られます。転送ログ・メッセージの詳細については、『748 ページの『ファイル転送ログ・メッセージ・フォーマット』』を参照してください。

MFT ロガーが開始されたのに転送情報がログとしてデータベースに記録されていない場合の対応策

Managed File Transfer ロガーが使用するデータベース表では、データベースのページ・サイズが 8 KB 以上でなければなりません。データベースのページ・サイズが十分な大きさでない場合、表が正しく作成されず、エラー SQLSTATE=42704 が表示されます。

Java Platform, Enterprise Edition データベース・ロガーを使用している場合、WebSphere Application Server システム・アウト・ログに次のメッセージが表示される可能性があります。スタンドアロン・データベース・ロガーを使用している場合、output0.log ファイルに次のエラーが表示される可能性があります。

```
DB2 SQL Error: SQLCODE=-204, SQLSTATE=42704
SQLERRMC=FTELOG.TRANSFER_EVENT, DRIVER=3.40.152
```

SQLSTATE 値 42704 は、ロガーが存在することが想定される表 (この場合 FTELOG.TRANSFER_EVENT) が存在しないことを示します。

この問題を解決するには、次のステップを実行します。

1. 表が存在し、完全であることを確認します。 ロガーが使用する表およびその列については、[667 ページの『MFT データベース・ロガー表』](#)を参照してください。
2. 表が存在しない、または不完全である場合、データベースのページ・サイズを確認します。
3. データベース・サイズが 8 KB より小さい場合、データベースのページ・サイズを大きくします。
 - データベースがテスト・システム上にあるか、データが含まれていない場合、その表を除去するか、ページ・サイズが 8 KB より大きいデータベースを再作成することができます。
 - ページ・サイズを大きくする方法については、[20 ページの『Windows、UNIX、Linux で使用する Db2 のログ・データベースのページ・サイズの拡張』](#) **z/OS** または [22 ページの『z/OS 上の Db2 上のデータベース表の IBM MQ 8.0 へのマイグレーション』](#)を参照してください。

MFT が AMS の鍵ストア構成ファイルから鍵ストア・プロパティーを読み取らない場合の対応策

Java AMS をクライアント・モードで実行するには、鍵ストアの構成ファイルがデフォルトの場所がない場合に、それを `MQS_KEYSTORE_CONF` 変数で指定する必要があります。場所が指定されていない場合、Managed File Transfer Agent ログにエラー・メッセージ "Failed to read keystore properties from the keystore configuration file." が示されます。

鍵ストアの構成ファイルのデフォルトの場所は、`home_directory/.mq/keystore.conf` です。鍵ストアの構成ファイルの場所がデフォルトの場所でない場合は、以下の手順を実行します。

1. FTE エージェントをクライアント・モードで開始します。
2. AMS セキュリティーを `SYSTEM.FTE.DATA.<agent name>` キューに適用します。鍵ストアの構成ファイルがこの場所がない場合、すべての転送が確認応答なしで失敗します。
3. `fteStartAgent` コマンドに関して、システム変数 `BFG_JVM_PROPERTIES` を `BFG_JVM_PROPERTIES=-DMQS_KEYSTORE_CONF=path to keystore_config file` に設定します。
4. `fteStartAgent` コマンドに関して、システム変数 `MQS_KEYSTORE_CONF` を `MQS_KEYSTORE_CONF=path to keystore_config file` に設定します。すべてのエージェントが、どのモードで実行されているかにかかわらず、確実に実行されるようにするには、これを設定する必要があります。

注: Java AMS がバインディング・モードで実行されている場合、鍵ストアの構成ファイルがデフォルトの場所がないと、キュー・マネージャーのエラー・ログにエラー AMQ9062 が表示されます。

ネイティブ・メモリーの消耗が原因で、MFT エージェントが `java.lang.OutOfMemoryError` で異常終了した場合の対策

ファイルからファイルへの転送、メッセージからファイルへの転送、ファイルからメッセージへの転送など、多数の管理された転送要求を処理している間に、エージェントが異常終了 (ABENDS) して `java.lang.OutOfMemoryError` を報告しますが、その時点で合計 RAM メモリーは十分に使用されていませんでした。この例外は、ネイティブ・メモリーを使い果たすことが原因で発生します。

問題の診断

この問題が発生すると、影響を受けるエージェントは ABEND し、根本原因に関する詳細を提供する 2 つのファイルを生成します。

- ABEND ファイル。このファイルの名前は、命名規則 `ABEND.FTE.date_timestamp.identifier.log` に準拠しています。

Multi マルチプラットフォームでは、このファイルは `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/ffdc` ディレクトリーに書き込まれます。

z/OS z/OS では、このファイルは USS ロケーション `$BFG_CONFIG/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/ffdc` に書き込まれます。

- Javacore ファイル。このファイルの名前の形式は、
javacore.datestamp.timestamp.pid.identifier.txt です。

Multi マルチプラットフォームでは、このファイルは MQ_DATA_PATH/mqft/logs/
coordination_qmgr_name/agents/agent_name ディレクトリーに書き込まれます。

z/OS z/OS では、このファイルは USS ロケーションの \$BFG_CONFIG/mqft/logs/
coordination_qmgr_name/agents/agent_name ディレクトリーに書き込まれます。

ABEND および Javacore のペアには、以下に示す例のような情報が含まれています。

例: ペア 1

Abend ファイル

```
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 ファイル

```
NULL -----
0SECTION TITLE subcomponent dump routine
NULL =====
1TISIGINFO Dump Event "systhrow" (00040000) Detail "java/lang/OutOfMemoryError" "native memory
exhausted" received
1TIDATETIME Date: 2020/01/09 at 11:35:18
1TIFILENAME Javacore filename:
C:\ProgramData\IBM\MQ\mqft\logs\COORDQM\agents\AGENT1\javacore.20200109.113518.14148.0002.txt
```

例: ペア 2

ABEND ファイル

```
Filename:
C:\ProgramData\IBM\MQ\mqft\logs\COORDQM\agents\AGENT1\logs\ffdc\ABEND.FTE.20200109143700286.3177895731698
464509.log
Level: p900-005-180821
Time: 09/01/2020 14:37:00:286 GMT
Thread: 918 (AgentStatusPublisher)
Class: com.ibm.wmqfte.thread.FTETHread
Instance: bc10bc1
Method: uncaughtException
Probe: ABEND_001
Cause: java.lang.OutOfMemoryError: Failed to create a thread: retVal -1073741830, errno 12

java.lang.OutOfMemoryError: Failed to create a thread: retVal -1073741830, errno 12
at java.lang.Thread.startImpl(Native Method)
at java.lang.Thread.start(Thread.java)
```

Javacore ファイル

```
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\COORDQM\agents\AGENT1\javacore.20200109.143700.2652.0003.txt
```

この問題が発生する理由

この問題は、エージェントが実行されているシステムでネイティブ・ヒープ・メモリーを使い果たしたことが原因で発生します。

Java ヒープ・メモリーとネイティブ・ヒープ・メモリーの違いについて詳しくは、[732 ページの『MFT エージェントが Java ヒープ・メモリーとネイティブ・ヒープ・メモリーを使用する方法』](#)を参照してください。

問題を回避する方法

ネイティブ・メモリーの枯渇が原因で `java.lang.OutOfMemoryError` が原因で MFT エージェントが停止する可能性を減らすために、いくつかのアクションを実行できます。

1. MFT エージェントを実行している JVM の Java ヒープのサイズを削減します。

割り振られた Java ヒープのサイズが大きいほど、ネイティブ・ヒープに使用できるメモリーは少なくなります。エージェントが使用する Java ヒープのサイズを小さくすると、ネイティブ・ヒープ用により多くのメモリーを解放できます。

デフォルトでは、エージェントの Java ヒープは 512 MB に設定されています。これを変更して値を大きくした場合は、値を減らして、実動のようなワークロードでテストすることを検討してください。

- エージェントを通常のプロセスとして実行する場合に Java ヒープを下げる、または変更するには、以下のようにします。

BFG_JVM_PROPERTIES 環境変数を設定して、オプション・ディレクトリーを JVM に渡します。例えば、Windows で、最大ヒープ・サイズを 1024 MB に設定するには、**fteStartAgent** コマンドを使用する前に以下のコマンドを実行します。

```
set BFG_JVM_PROPERTIES="-Xmx1024M"
```

BFG_JVM_PROPERTIES 環境変数を使用して Java システム・プロパティーを設定する方法については、[MFT Java システム・プロパティー](#)を参照してください。

- エージェントを Windows サービスとして実行する場合に Java ヒープを下げる、または変更するには、以下のようにします。

エージェントを Windows サービスとして実行している JVM にオプションを渡すには、**fteModifyAgent** コマンドで指定された **-sj** パラメーターを使用してエージェントを変更します。

以下の例では、**-sj** パラメーターを指定した **fteModifyAgent** コマンドを使用して、Windows サービス構成エージェントを実行している JVM の Java ヒープの最大サイズを設定します。

```
fteModifyAgent.cmd -agentName AGENT1 -s -su user1 -sp passwd0rd -sj -Xmx1024M
```

エージェントが再始動された後、エージェントの `output0.log` ファイルを確認することにより、これが正常に設定されていることをチェックできます。「現在の環境の表示の始まり」セクションで、次のように値 1024 MB がレポートされます。

```
The maximum amount of memory that the Java virtual machine will attempt to use is: '1024'MB
```

2. ネイティブ・メモリー使用の制限

多くの場合、ネイティブ・ヒープの不足によって引き起こされる `java.lang.OutOfMemoryErrors` は、エージェントが BINDINGS トランスポートを使用してエージェント・キュー・マネージャーに接続すると発生します。エージェントは、BINDINGS トランスポートを使用するように構成されている場合、キュー・マネージャーと通信する必要があるときにはいつでもネイティブ・メソッドを呼び出します。

これは、キュー・マネージャーへの接続が増加し、メッセージ通信が増加するため、エージェントのワークロードが増加するに従ってネイティブ・メモリー使用量が増加することを意味します。この状態では、ワークロードを削減することが役立ちます。これを行うには、以下のエージェント・プロパティをデフォルトの 25 よりも低い値に設定します。

- `maxSourceTransfers`
- `maxDestinationTransfers`

これにより、発生する可能性のある同時転送の数が削減されるため、エージェントの最大同時ワークロードも削減されます。

3. エージェント・キュー・マネージャーへの接続時に CLIENT トランスポートを使用するようにエージェントを構成します。これを行うには、以下のエージェント・プロパティを設定します。

- `agentQMgrHost`
- `agentQMgrPort`
- `agentQMgrChannel`

これらのプロパティに関する情報は、[MFT agent.properties ファイル](#)のトピックに記載されています。

これにより、エージェントとキュー・マネージャー間のすべての通信が、ネイティブ・コードではなく TCP/IP を介して行われるようになります。これにより、エージェントが使用するネイティブ・メモリーの量が削減されます。

重要: このアクションを実行すると、パフォーマンスも低下します。ネイティブ・コードではなく、ローカル・ホストへの TCP/IP 接続を使用すると、エージェントがキュー・マネージャーとの対話を必要とする場合、この構成はあまり効率的ではありません。

BFGSS0023E エラーとその回避方法

以前のバージョンの製品に戻すためにインストール済み環境からフィックスパックをアンインストールする場合に、インストール済み環境に関連しているエージェントがアンインストールの実行時に管理対象転送に関係していると、そのエージェントは開始できず、BFGSS0023E エラーが報告されます。このエラーは、エージェントの再始動時に BFGSS0023E メッセージが表示されないようにするためのいくつかの手順を実行することで回避できます。

エージェントが現在関係している未完了の管理対象転送ごとに、エージェントの `SYSTEM.FTE.STATE.agent_name` キュー上にメッセージがあります。このメッセージは、管理対象転送に関するチェックポイント情報を格納し、その管理対象転送がリカバリーされる際に使用されます。管理対象転送が完了すると、`SYSTEM.FTE.STATE.agent_name` キュー上のこれに対応するメッセージが削除されます。

各状態メッセージには、管理対象転送の実行中にエージェントが使用していた Managed File Transfer コンポーネントのバージョンを示す何らかの内部ヘッダー情報が含まれています。このバージョン情報は、特定のフィックスパックのレベルを示します。したがって、例えば IBM MQ 8.0.0 Fix Pack 5 エージェントが管理対象転送を実行していた場合、その管理対象転送の状態メッセージには、IBM MQ 8.0.0 Fix Pack 5 への参照が含まれることになります。

インストール済み環境からフィックスパックをアンインストールする場合に、そのインストール済み環境に関連しているエージェントで、そのエージェントに関連付けられている未完了の転送があると、エージェントは開始できず、次のエラーが報告されます。

BFGSS0023E: エージェントは、新しいバージョンを使用して作成されたデータを含む IBM MQ キューを使用するように構成されています。

製品の エージェントはこの構成では実行できないため、終了します。

例えば、IBM MQ 8.0.0 Fix Pack 5 エージェントが停止してから IBM MQ 8.0.0 Fix Pack 4 レベルにダウングレードされたときに、そのエージェントで未完了の転送がいくつか実行されている場合、エージェントが

次回に開始したときに、エージェントによってその SYSTEM.FTE.STATE.agent_name キュー上のメッセージが確認され、それらのメッセージが IBM MQ 8.0.0 Fix Pack 5 を使用していたときに書き込まれたことが分かります。現在は IBM MQ 8.0.0 Fix Pack 4 が使用されているため、前述のように、エージェントは BFGSS0023E エラーを報告してシャットダウンします。

一般に、いずれかの Managed File Transfer コンポーネントに対するフィックスパックを除去する場合は、以下の手順を実行することによって、エージェントの再始動時に BFGSS0023E メッセージが表示されなくなるはずですが。

1. すべてのエージェントで管理対象転送が完了していることを確認します。
2. エージェントを停止します。
3. フィックスパックを除去します。
4. エージェントを再始動します。

関連タスク

[155 ページの『MFT エージェントの開始』](#)

Managed File Transfer エージェントを使用してファイル転送を行うには、まずエージェントを開始する必要があります。

関連資料

[620 ページの『MFT エージェント・キュー設定』](#)

fteCreateAgent コマンドで生成される MQSC コマンド・スクリプトは、次の値に設定されたパラメーターでエージェント・キューを作成します。キューを作成するために指定された MQSC スクリプトを使用せず、キューを手動で作成する場合、指定された値に次のパラメーターを設定する必要があります。

関連情報

[BFGSS0001 - BFGSS9999](#)

[キュー・マネージャーを前のバージョンに戻す \(UNIX\)](#)

[キュー・マネージャーを前のバージョンに戻す \(Windows\)](#)

管理対象転送が BFGIO0341E エラーで失敗する場合の対処法

外部プロセスでモニターしている場所に管理対象転送でファイルを転送する場合は、その管理対象転送が「BFGIO0341E: 一時ファイル *destination_filename.part* は存在していないため、名前を *destination_filename* に変更できませんでした」というエラーで失敗する可能性があります。この原因は、管理対象転送の宛先エージェントが宛先ファイルを書き込むときに一時ファイルを使用する方法にあります。

宛先エージェントによる一時ファイルの使用方法

デフォルトでは、管理対象ファイル転送が行われる際に、宛先エージェントは以下のステップを実行します。

- *destination_filename.part* という一時ファイルを作成します。
- 一時ファイルをロックします。
- ソース・エージェントから受信したファイル・データを一時ファイルに書き込みます。
- ファイル・データをすべて受信して書き出した後、一時ファイルをアンロックします。
- 一時ファイルの名前を *destination_filename.part* から *destination_filename* に変更します。

管理された転送がリカバリーに入る場合、宛先エージェントは *destination_filename.partnumber* という一時ファイルを作成することができます。次に、宛先エージェントは、*destination_filename.part* という名前のファイルではなく、ファイル・データをこのファイルに書き込みます。

一時ファイル名 *destination_filename.partnumber* が既に存在する場合は、宛先エージェントは *destination_filename.part(number + 1)* という名前の新しい一時ファイルを作成しようとし、そのファイルが既に存在する場合、宛先エージェントは、*destination_filename.part(number + 2)* という名前の一時ファイルを作成しようとし、そのため、ファイルが正常に作成されるまでは、

そのファイルが正常に作成されます。一時ファイル `destination_ filename.part1000` を作成しようとして失敗した場合、エージェントは宛先ファイルに直接書き込み、一時ファイルを使用しません。

管理対象の転送が完了すると、宛先エージェントは、`destination_ filename.partnumber` という名前の一時ファイルをすべて削除します。これは、管理対象転送中にエージェントによって作成されたものであることを前提としています。

注：エージェント・プロパティ `doNotUseTempOutputFile` に値 `true` が設定されている場合、宛先エージェントは一時ファイルを使用しません。代わりに、宛先ファイルに直接書き込みます。

`doNotUseTempOutputFile` プロパティについて詳しくは、[MFT agent.properties](#) ファイルを参照してください。

この問題が発生する理由

BFGIO0341E エラーが生成されるのは、宛先エージェントが一時ファイルの名前を変更しようとしたが、そのファイルがもう存在しなくなっていたという場合です。この問題が起きる典型的なシナリオは、以下のとおりです。

- ターゲットのファイル・システム上にステージング・ディレクトリーがセットアップされている。
- 外部プロセスが、ステージング・ディレクトリーをモニターし、検出したすべてのファイルを新しい場所に移動するように構成されている。
- 宛先エージェントがステージング・ディレクトリー内の一時ファイル `destination_ filename.part` を作成し、ロックします。
- 宛先エージェントが、一時ファイルにファイル・データを書き込む。
- すべてのファイル・データが一時ファイルに書き込まれた後、宛先エージェントがそのファイルをアンロックする。
- 外部プロセスが、一時ファイルを検出して新しい場所に移動する。
- 宛先エージェントが、一時ファイルの名前を変更しようとするが、ファイルはもう存在しなくなっている。その結果、転送項目に BFGIO0341E エラーによる「失敗」のマークが付けられます。

この問題を回避する方法

BFGIO0341E エラーが起こらないようにするには、以下の 2 つの方法があります。

- 宛先エージェントが書き込む一時ファイルは、必ず、`.part` または `.partnumber` のいずれかの接尾部で終わります。これらのファイルを移動せずに無視するように外部プロセスを構成できれば、宛先エージェントが名前変更操作を行うときにターゲット・ディレクトリーにファイルがまだ存在するようになります。
- もう一つの方法としては、一時ファイルを使用せずに宛先ファイルに直接書き込むように宛先エージェントを構成します。すべてのファイル・データが宛先ファイルに書き込まれたときに初めて、宛先ファイルがアンロックされ、外部プロセスが宛先ファイルを取得できるようになります。

宛先ファイルに直接書き込むように宛先エージェントを構成するには、エージェント・プロパティ `doNotUseTempOutputFile=true` を設定します。このプロパティについて詳しくは、[MFT agent.properties](#) ファイルを参照してください。

MFT の戻りコード

Managed File Transfer コマンド、Ant タスク、およびログ・メッセージは、機能が正常に実行されたかどうかを示す戻りコードを提供します。

以下の表では、製品の戻りコードとその意味をリストしています。

戻りコード	短縮名	説明
0	成功	コマンドは正常終了しました。

表 24. 戻りコード (続き)

戻りコード	短縮名	説明
1	コマンド失敗	コマンドは正常に終了していません。
2	コマンドのタイムアウト	エージェントは、指定されたタイムアウト時間内にコマンドの状況を含む応答をしませんでした。デフォルトでは、管理対象の呼び出しおよび転送コマンドについては、このタイムアウトの制限はありません。例えば、 fteCreateTransfer コマンドに -w パラメーターを指定する場合などです。他のコマンドについては、デフォルトで、このタイムアウトは 5 秒です。
3	確認応答のタイムアウト	エージェントは、指定されたタイムアウト時間内にコマンドの受信確認をしませんでした。デフォルトで、このタイムアウトは 5 秒です。
4	誤ったエージェント	コマンドは誤ったエージェントに送信されました。コマンド XML で指定したエージェントは、メッセージが入れられたコマンド・キューを読み取っているエージェントではありません。
20	転送は一部正常終了	転送は、部分的に成功して完了し、一部のファイルが転送されました。
21	転送の停止	転送は、ユーザー出口の 1 つによって停止されました。
22	転送取り消しのタイムアウト	エージェントは、転送の取り消し要求を受け取りましたが、30 秒以内に取り消しを完了できませんでした。転送は取り消されませんでした。
26	取り消し ID がありません	エージェントは、転送の取り消し要求を受け取りましたが、転送が見つかりませんでした。この場合原因として、取り消し要求がエージェントによって処理される前に転送が完了したことが考えられます。また fteCancelTransfer コマンドに対して指定した転送 ID が正しくないことが原因の可能性もあります。取り消し要求は無視されました。

表 24. 戻りコード (続き)

戻りコード	短縮名	説明
27	取り消しが進行中	エージェントは、転送の取り消し要求を受け取りましたが、転送は既に取り消し処理中です。新しい転送の取り消し要求は無視されました。
40	失敗	転送は失敗し、指定したどのファイルも転送されませんでした。
41	取り消し済み	転送は取り消されました。
42	トリガー失敗	転送が条件付きであり、必要な条件が満たされなかったため、転送は実行されませんでした。
43	誤った形式の XML	XML メッセージの形式が正しくありません。
44	ソース・エージェントの容量を超過	ソース・エージェントに、転送を実行するための容量が十分に存在しません。
45	宛先エージェントの容量を超過	宛先エージェントに、転送を実行するための容量が十分に存在しません。
46	ソース・エージェントの最大ファイル数を超過	転送ファイル数が、ソース・エージェントの制限を超過しました。
47	宛先エージェントの最大ファイル数を超過	転送ファイル数が、宛先エージェントの制限を超過しました。
48	無効なログ・メッセージの属性	誤った形式のログ・メッセージ。このエラーは内部エラーです。この戻りコードを受け取った場合は、IBM サポートに支援を求めてください。
49	宛先が到達不能	ソース・エージェントは、IBM MQ の問題が原因で宛先エージェントにメッセージを送信できません。例えば、ソース・エージェントのキュー・マネージャーが、宛先エージェントのキュー・マネージャーと通信するように正しく構成されていません。
50	試用版の違反	試用版エージェントにより、試用版エージェントではないエージェントとの通信が試行されました。
51	ソース転送が許可されていない	maxSourceTransfers エージェント・プロパティが 0 に設定されました。このエージェントは、転送のソースにすることは許可されていません。

表 24. 戻りコード (続き)

戻りコード	短縮名	説明
52	宛先転送が許可されていない	maxDestinationTransfers エージェント・プロパティが 0 に設定されました。このエージェントが転送の宛先となることは許可されていません。
53	権限なし	ユーザーに操作の実行権限がありません。詳細については、付随するメッセージを参照してください。
54	権限レベルが一致しない	ソース・エージェントと宛先エージェントの authorityChecking エージェント・プロパティ値が一致しません。
55	トリガーがサポートされない	プロトコル・ブリッジ・エージェントでトリガーを使用して転送の作成が試行されました。この動作はサポートされていません。
56	ファイルからメッセージへの転送は宛先でサポートされない	宛先エージェントでは、宛先キューへのファイルの書き込みがサポートされていません。
57	ファイル・スペースがサポートされない	宛先エージェントはファイル・スペースをサポートしません。
58	ファイル・スペースが拒否された	宛先エージェントにより、ファイル・スペース転送が拒否されました。
59	ファイルへの宛先メッセージはサポートされない	宛先エージェントはメッセージからファイルへの転送をサポートしていません。
64	両方がキューであることは許可されない	転送のソースと宛先がどちらもキューです。
65	一般データ・キュー・エラー	Managed File Transfer Agent データ・キューにアクセスしたときにエラーが発生しました。
66	データ・キュー書き込み許可エラー	Managed File Transfer Agent データ・キューにアクセスしたときにエラーが発生しました。Advanced Message Security が使用可能ではありません。
67	データ・キュー書き込み AMS エラー	Managed File Transfer Agent データ・キューにアクセスしたときに許可エラーが発生しました。Advanced Message Security が使用可能になっています。

表 24. 戻りコード (続き)

戻りコード	短縮名	説明
V9.0.1 ▶ V9.0.1 69	転送リカバリー・タイムアウト	転送のリカバリーは、指定された transferRecoveryTimeout 値の経過後にタイムアウトになりました。
70	エージェントが異常終了しました	アプリケーションは、リカバリー不能な問題が発生したため、強制的に終了しています。
75	キュー・マネージャーが使用不可	アプリケーションにキュー・マネージャーを使用できないため、アプリケーションを続行できません。
78	始動構成に問題があります	始動構成データに問題があるため、アプリケーションを続行できません。
85	データベース・サーバーに問題があります	データベースに問題があるため、アプリケーションを続行できません (通常はログガーによってのみ返されます)
100	モニターの置換が無効	モニター・タスク XML スクリプト内の変数置換のフォーマットが正しくありません。
101	正しくないモニター・リソース	モニター・リソース定義の数が無効でした。
102	正しくないモニター・トリガー	モニター・トリガー定義の数が無効でした。
103	正しくないモニター・タスク	モニター・タスク定義の数が無効でした。
104	モニターの欠落	要求されたモニターがありません。
105	既に存在するモニター	要求されたモニターは既に存在しています。
106	モニターのユーザー出口エラー	リソース・モニターのポーリング中にモニターのユーザー出口でエラーが生成されました。
107	モニターのユーザー出口の取り消し	モニターのユーザー出口が、トランザクションの取り消しを要求しました。
108	モニター・タスクの失敗	モニター・タスクは、タスク処理のエラーのため、完了できませんでした。
109	モニター・リソースの失敗	モニター・リソース定義を指定のリソースに適用できません。

表 24. 戻りコード (続き)

戻りコード	短縮名	説明
110	モニター・タスクの変数置換の失敗	モニター・タスクで変数が指定されましたが、一致する名前がメタデータに見つかりませんでした。そのため、変数を値に置換できません。
111	モニター・タスクのソース・エージェントが無効	モニター転送タスクのソース・エージェントが、リソース・モニターのエージェントと一致しません。
112	モニター・タスクのソース・キュー・マネージャーが無効	モニター転送タスクのソース・エージェント・キュー・マネージャーが、リソース・モニターのエージェント・キュー・マネージャーと一致しません。
113	モニターがサポートされない	プロトコル・ブリッジ・エージェントでリソース・モニターの作成または削除が試行されました。この動作はサポートされていません。
114	モニター・リソースが拒否される	モニター・リソースによってスキャンされるディレクトリーでアクセスが拒否されます。
115	モニター・リソース・キューが使用中	モニター・リソース・キューが既に開いており、共有アクセスでの入力との互換性がありません。
116	モニター・リソース・キューが不明	モニター・リソース・キューが、モニターの関連キュー・マネージャーに存在しません。
118	モニター・リソース式が無効	XPath 式の評価中にエラーが発生しました。XPath 式は、メッセージのヘッダー内のユーザー定義プロパティにアクセスするものと評価されました。メッセージは、リソース・モニターによりモニターされるキューにあります。
119	モニター・タスクのソース・エージェント・キュー・マネージャーが欠落	モニター・タスク定義で、ソース・エージェント名またはソース・エージェント・キュー・マネージャー名が欠落しています。
120	モニター・キューが有効になっていない	モニター・リソース・キューが有効になっていません。
121	モニター・キューへのアクセス時の予期しないエラー	モニター・リソース・キューへのアクセス中に予期しないエラーが発生します。

戻りコード	短縮名	説明
122	モニター・コマンド・キューがコンテンツ ID に対して有効になっていない	モニター・エージェント・コマンド・キューが、設定されているコンテンツ ID に対して有効になっていません。

以下の表では、製品の間中応答コードとその意味をリストしています。

応答コード	短縮名	説明
-2	ACK	要求は受信しましたが、完了せずに保留中です。
-3	PROGRESS	要求は複数のファイルに対するものであり、一部がまだ完了せずに保留中です。

注：

応答コードが存在するのは、要求を生成したプロセスが応答キューを提供した場合のみです。これらは中間応答であり、Managed File Transfer コマンドは最終応答コードのみを返します。

関連資料

344 ページの『ファイル転送の戻りコード』

転送に含まれている個々のファイルには、それぞれ独自の結果コードがあります。それらの結果コードの意味は、コマンドから返される全体的な戻りコードとは別の意味があります。

ファイル転送の戻りコード

転送に含まれている個々のファイルには、それぞれ独自の結果コードがあります。それらの結果コードの意味は、コマンドから返される全体的な戻りコードとは別の意味があります。

<action> エレメントが値「経過」に設定されている転送ログ進行状況メッセージで、報告される各ファイルには結果コードという <status> エレメントが含まれています。以下に例を示します。

```
<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>
```

次の表に、resultCode として戻される値を示します。

結果コードの値	説明
0	正常終了しました。ファイルは正常に転送されました。
1	失敗しました。ファイルの転送は失敗しました。このエラーについて詳しくは、<supplement> エレメントを参照してください。

表 26. 転送に関するファイルの結果コード (続き)

結果コードの値	説明
2	警告。ファイルは転送されましたが、警告メッセージが報告されました。例えば、ソースの後処理として削除が設定されているのに、ソース・ファイルが削除されていない可能性があります。この警告について詳しくは、^{supplement} エlementを参照してください。

Connect:Direct ブリッジのトラブルシューティング

以下の参照情報および例を、Connect:Direct ブリッジから返されるエラーの診断に役立ててください。

- [345 ページの『Connect:Direct ブリッジのトレース』](#)
- [346 ページの『Connect:Direct ブリッジのログ情報』](#)
- [346 ページの『Connect:Direct ノードでの権限問題の解決』](#)
- [347 ページの『Connect:Direct ノードとの間のテキスト転送でデータが正しく変換されない場合の対策』](#)
-  [347 ページの『Connect:Direct ブリッジ経由で PDS または PDS メンバーに転送する操作が失敗する場合の対策』](#)
-  [348 ページの『ダブルスラッシュで指定される Connect:Direct ファイル・パス』](#)
- [348 ページの『Connect:Direct ブリッジの同時転送数を増やす』](#)
- [350 ページの『ファイル転送で呼び出される Connect:Direct プロセスのデバッグ』](#)

Connect:Direct ブリッジのトレース

Connect:Direct ブリッジの一部である Connect:Direct ノードからトレースを収集して、問題判別に役立てることができます。

このタスクについて

トレースを使用可能にするには、以下のステップを実行します。

手順

1. Connect:Direct ブリッジ・エージェントを停止します。
2. Connect:Direct ブリッジ・エージェントのプロパティ・ファイルを編集して、以下の行を組み込みます。

```
cdTrace=true
```

3. Connect:Direct ブリッジ・エージェントを開始します。

タスクの結果

トレース情報が、Connect:Direct ブリッジ・エージェントの構成ディレクトリーにある output0.log ファイルに書き込まれます。

関連資料

[550 ページの『MFT agent.properties ファイル』](#)

各 Managed File Transfer Agent には、独自のプロパティ・ファイル agent.properties があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。agent.properties ファイルには、エージェントの動作を変更するプロパティを含めることもできます。

Connect:Direct ブリッジのログ情報

Connect:Direct ブリッジ・エージェントを使用して、MFT エージェントと Connect:Direct ノードの間でファイルを転送できます。そのような転送にかかわっている Connect:Direct のノードとプロセスのログ情報は、IBM MQ Explorer・プラグインに表示され、ログ・データベースに格納されます。

Connect:Direct ブリッジ・エージェントは、IBM WebSphere MQ File Transfer Edition 7.0.4 以降でなければなりません。転送にかかわるその他のエージェントは、Managed File Transfer のどのバージョンでもかまいません。ただし、Connect:Direct のノードとプロセスに関する情報をログに記録するには、転送にかかわるすべての MFT エージェントが IBM WebSphere MQ File Transfer Edition 7.0.4 以降でなければなりません。その情報を IBM MQ Explorer・プラグインで表示するには、そのプラグインが IBM WebSphere MQ File Transfer Edition 7.0.4 以降でなければなりません。その情報をログ・データベースに格納するには、データベース・ロガーとデータベース・スキーマが IBM WebSphere MQ File Transfer Edition 7.0.4 以降でなければなりません。

ファイル転送にかかわる Connect:Direct ノードと Connect:Direct プロセスのログ情報は、調整キュー・マネージャーの SYSTEM.FTE トピックにパブリッシュされるログ・メッセージに組み込まれます。詳細については [748 ページの『ファイル転送ログ・メッセージ・フォーマット』](#) を参照してください。

パブリッシュされるメッセージには、以下の情報が組み込まれます。

- Connect:Direct ブリッジ・ノード名
- 1次ノード (PNODE) 名
- 2次ノード (SNODE) 名
- プロセス名
- プロセス ID 番号

Connect:Direct ブリッジ・ノードは、1次ノードまたは2次ノードのいずれかと同じノードになります。

Connect:Direct ブリッジ・ノード名の値は、MFT Connect:Direct ブリッジ・エージェントに認識されているブリッジ・ノード名です。1次ノード名と2次ノード名は、Connect:Direct ブリッジ・ノードのネットワーク・マップに含まれている各ノードを参照するために使用する名前です。

関連資料

[766 ページの『Connect:Direct ブリッジの転送ログ・メッセージの例』](#)

宛先エージェントまたはソース・エージェントが Connect:Direct ブリッジ・エージェントの場合は、destinationAgent エlement または sourceAgent エlement に追加の属性が含まれます。開始ログ・メッセージには、Connect:Direct 転送に関する情報のサブセットだけが含まれます。進行中と完了のログ・メッセージには、Connect:Direct 転送に関する情報全体が含まれます。

Connect:Direct ノードでの権限問題の解決

Managed File Transfer と Connect:Direct 間の転送が権限不十分のエラーを伴って失敗する場合、このトピックの情報を参照してください。

Connect:Direct ブリッジが関与する転送の場合、Connect:Direct ノードに接続するユーザー ID は、どの IBM MQ Message Descriptor (MQMD) ユーザー ID が転送要求に関連付けられているかによって決定されます。特定の MQMD ユーザー ID を特定の Connect:Direct ユーザー ID にマップすることができます。詳細については [146 ページの『Connect:Direct の資格情報のマップ』](#) を参照してください。

以下のエラーのいずれかを伴って転送が失敗する場合があります。

- ```
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.
```

これらのエラーのいずれかが表示される場合は、転送要求に使用された MQMD ユーザー ID にどの Connect:Direct ユーザー ID が関連付けられているかを判別してください。この Connect:Direct ユーザー ID には、Connect:Direct ブリッジで要求される Connect:Direct 操作を実行する権限が付与されている必要があります。必要な機能権限のリストおよびこれらの権限を付与する方法については、[146 ページの『ConnectDirectCredentials.xml ファイルを使用した Connect:Direct の資格情報のマップ』](#)を参照してください。

Connect:Direct ノードとの間のテキスト転送でデータが正しく変換されない場合の対策

MFT エージェントと Connect:Direct ノードの間でテキスト・モードでファイルを転送すると、コード・ページと行末文字の変換が実行されます。転送では、Connect:Direct ブリッジ・ノードのネットワーク・マップにあるオペレーティング・システム情報に基づいて、リモート・ノードの行末文字が判別されます。ネットワーク・マップにある情報が不正確であれば、行末文字の変換が正しく実行されない可能性があります。

Connect:Direct ブリッジ・ノードと、転送の宛先として使用する Connect:Direct ノードのネットワーク・マップに、正しいプラットフォームの記述を組み込むようにします。

- Connect:Direct ブリッジ・ノードが Windows システムにある場合は、ネットワーク・マップに含まれているリモート・ノードごとに、「オペレーティング・システム」リストから正しい値を選択するようにしてください。
 - リモート・ノードが Windows システムにある場合は、「Windows」を選択します。
 - リモート・ノードが UNIX または Linux のシステムにある場合は、「UNIX」を選択します。
 -  リモート・ノードが z/OS システムにある場合は、「OS/390」を選択します。

Connect:Direct ブリッジは、その他のオペレーティング・システムのリモート・ノードへの転送には対応していません。

- Connect:Direct ブリッジ・エージェントの構成ディレクトリーにある `ConnectDirectNodeProperties.xml` ファイルを使用して、ファイル転送にかかわるリモート・ノードごとに、リモート Connect:Direct ノードのオペレーティング・システム・タイプを指定します。詳細については、[145 ページの『ConnectDirectNodeProperties.xml ファイルを構成して、リモート Connect:Direct ノードに関する情報を組み込む』](#) および [588 ページの『Connect:Direct ノードのプロパティ・ファイルのフォーマット』](#)を参照してください。

関連資料

[645 ページの『Connect:Direct と MFT の間のテキスト・ファイル転送』](#)

テキストの転送には、あるコード・ページから別のコード・ページへの、ファイルのコード・ページの変換が伴います。テキスト・ファイルの転送には、システム間の CRLF (復帰改行) 文字の変換も伴います。このトピックでは、Managed File Transfer Agent と Connect:Direct ノードの間のテキスト・ファイル転送の動作をまとめます。

Connect:Direct ブリッジ経由で PDS または PDS メンバーに転送する操作が失敗する場合の対策

転送の宛先が z/OS 上の Connect:Direct ノードであり、PDS または PDS メンバーである場合、`-de` パラメーターに `overwrite` という値が指定されていなければ、転送は失敗します。

このタスクについて

`fteCreateTransfer` コマンドまたは `fteCreateTemplate` コマンドを使用して転送要求を送信した場合は、以下の手順を実行します。

手順

1. 送信したコマンドを変更して、`-de overwrite` を組み込みます。
2. コマンドを再実行依頼してください。

IBM MQ Explorer ・ プラグインの使用

このタスクについて

IBM MQ Explorer ・ プラグインを使用して転送要求を送信した場合は、以下の手順を実行します。

手順

1. 「ファイル転送管理の新規作成」ウィザードでソースと宛先の情報を指定します。
2. 「宛先ファイル・システム上の同じ名前のファイルを上書きする」を選択します。
3. コマンドを再実行依頼してください。

ダブルスラッシュで指定される Connect:Direct ファイル・パス

ファイル転送の一部として、Connect:Direct ノードにあるファイルを、ダブルスラッシュ (//) で始まるファイル・パスを使用して指定した場合、そのファイルはデータ・セットとして扱われます。

Connect:Direct ノード上のソースおよび宛先は、`cd_node_name:file_path` という形式で指定されます。`file_path` がダブルスラッシュ (//) で始まっている場合、ソースまたは宛先はデータ・セットとして扱われます。Connect:Direct ノードが z/OS 上にない場合でも同様です。ファイルがデータ・セットではないのにダブルスラッシュ (//) で始まるファイル・パスを誤って指定してしまい、転送が失敗するということがあります。

指定するファイルをデータ・セットとして扱わない場合は、ダブルスラッシュ (//) で始まる `file_path` を指定しないようにしてください。

関連概念

[345 ページの『Connect:Direct ブリッジのトラブルシューティング』](#)

以下の参照情報および例を、Connect:Direct ブリッジから返されるエラーの診断に役立ててください。

関連資料

[634 ページの『Connect:Direct ノードとの間のデータ・セット転送』](#)

Connect:Direct ブリッジを使用して、Managed File Transfer エージェントと IBM Sterling Connect:Direct ノードの間でデータ・セット転送を実行できます。転送のソースと宛先のいずれかまたは両方としてデータ・セットを指定できます。

Connect:Direct ブリッジの同時転送数を増やす

Connect:Direct ブリッジ・エージェントが処理できる同時転送数を増やすには、3つのエージェント・プロパティを変更する必要があります。Connect:Direct ノードが受け入れる接続の最大数も増やす必要があります。

Connect:Direct ブリッジ・エージェントが処理できる同時転送の最大数は、いくつかのエージェント・プロパティの値により決定されます。**maxSourceTransfers** エージェント・プロパティおよび **maxDestinationTransfers** エージェント・プロパティでは、Connect:Direct ブリッジ・エージェントの転送数のデフォルト値は5になっています。このデフォルト値は、他のタイプのエージェントの転送数のデフォルト値25よりも低いものです。エージェントが **maxSourceTransfers** および **maxDestinationTransfers** のデフォルト値で構成されている場合、Connect:Direct ブリッジで同時に処理できる転送は最大10個です。つまり、エージェントがソースの転送が5個とエージェントが宛先の転送が5個です。

これらのデフォルト値によって、Connect:Direct ブリッジ・エージェントが、Connect:Direct ノードへのAPI接続の最大数を超過しないようになっています。デフォルト構成のConnect:Direct ブリッジ・エージェントでは、Connect:Direct ノードへのAPI接続が最大10個が使用されます。UNIX上のConnect:Direct ノードによって受け入れられる接続の最大数は、**api.max.connects** Connect:Direct パラメーターによって制御されます。Windows上のConnect:Direct ノードの場合、対応するパラメーターは **max.api.connects** です。

Connect:Direct ブリッジが大量のファイル転送を実行するときに速度が不十分の場合は、Connect:Direct ブリッジ・エージェントが処理する同時転送の数を増やすことができます。Connect:Direct ブリッジ・エージェントの以下のエージェント・プロパティを変更してください。

maxSourceTransfers

このプロパティを 5 より大きい値、ただし 25 以下の値に設定します。25 より大きい値を選択した場合、エージェントが使用する JVM の使用可能なメモリー量を増やさない限り、エージェントがメモリー不足になるおそれがあります。

maxDestinationTransfers

このプロパティを 5 より大きい値、ただし 25 以下の値に設定します。25 より大きい値を選択した場合、エージェントが使用する JVM の使用可能なメモリー量を増やさない限り、エージェントがメモリー不足になるおそれがあります。

ioThreadPoolSize

ioThreadPoolSize のデフォルト値は 10 です。このプロパティは、Connect:Direct ブリッジ・エージェントがソース・エージェントである転送のための Connect:Direct ノード API 接続の数を制限します。Connect:Direct から Managed File Transfer への転送が、この場合の転送に該当します。以下の説明に従ってこのプロパティの値を設定してください。

- **maxSourceTransfers** の値が **maxDestinationTransfers** の値より小さい場合、**ioThreadPoolSize** を **maxSourceTransfers** の 2 倍の値かまたは 10 のどちらか大きい方に設定します。
- **maxSourceTransfers** の値が **maxDestinationTransfers** の値より大きい場合、**ioThreadPoolSize** を **maxSourceTransfers** と **maxDestinationTransfers** の合計に設定します。

これらのエージェント・プロパティに加えて、Connect:Direct ブリッジの一部である Connect:Direct ノードの同時 API 接続の最大数も変更する必要があります。この数を制御する Connect:Direct パラメーターは、ノードが UNIX 上にある場合は **api.max.connects**、ノードが Windows 上にある場合は **max.api.connects** になります。該当のパラメーターを以下のように変更してください。

api.max.connects (Connect:Direct ブリッジのノードが UNIX 上にある場合)

このパラメーターは、**maxSourceTransfers** と **maxDestinationTransfers** の合計より大きい値に設定してください。**api.max.connects** パラメーターのデフォルト値は 16 です。このパラメーターの設定方法について詳しくは、Connect:Direct の資料を参照してください。

max.api.connects (Connect:Direct ブリッジのノードが Windows 上にある場合)

このパラメーターは、**maxSourceTransfers** と **maxDestinationTransfers** の合計より大きい値に設定してください。**max.api.connects** パラメーターのデフォルト値は 10 です。このパラメーターの設定方法について詳しくは、Connect:Direct の資料を参照してください。

関連タスク

[143 ページの『Connect:Direct ブリッジの構成』](#)

Managed File Transfer ネットワークと Connect:Direct ネットワークの間でファイルを転送するために、Connect:Direct ブリッジを構成します。Connect:Direct ブリッジのコンポーネントは、Connect:Direct ノードと、そのノードと通信するための専用の Managed File Transfer エージェントです。このエージェントのことを Connect:Direct ブリッジ・エージェントといいます。

関連資料

[550 ページの『MFT agent.properties ファイル』](#)

各 Managed File Transfer Agent には、独自のプロパティ・ファイル **agent.properties** があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。**agent.properties** ファイルには、エージェントの動作を変更するプロパティを含めることもできます。

ファイル転送で呼び出される Connect:Direct プロセスのデバッグ

ファイル転送で呼び出される Connect:Direct プロセスのログ情報が Connect:Direct ブリッジ・エージェント構成ディレクトリーの output0.log ファイルに書き込まれるように、Connect:Direct ブリッジ・エージェントを構成できます。

このタスクについて

Connect:Direct プロセスのロギングを構成するには、以下のステップを実行します。

手順

1. Connect:Direct ブリッジ・エージェントを停止します。
2. `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` ディレクトリーの `agent.properties` ファイルを編集して、プロパティー `logCDProcess` が含まれるようにします。
`logCDProcess` プロパティーでは以下のいずれかの値を使用できます。
 - None - 情報は何も記録されません。これがデフォルトです。
 - Failures - 失敗した Connect:Direct プロセスについての情報が記録されます。
 - All - すべての Connect:Direct プロセスについての情報が記録されます。
3. Connect:Direct ブリッジ・エージェントを開始します。

タスクの結果

Connect:Direct プロセスについての情報が Connect:Direct ブリッジ・エージェントの output0.log ファイルに記録されます。記録される情報は、以下のとおりです。

- MFT 転送 ID
- Connect:Direct プロセス名
- Connect:Direct プロセス番号
- 生成されたプロセス定義
- プロセス・テンプレートのファイル名 (Connect:Direct プロセスがユーザー定義の場合)

関連概念

[345 ページの『Connect:Direct ブリッジのトラブルシューティング』](#)

以下の参照情報および例を、Connect:Direct ブリッジから返されるエラーの診断に役立ててください。

関連資料

[550 ページの『MFT agent.properties ファイル』](#)

各 Managed File Transfer Agent には、独自のプロパティー・ファイル `agent.properties` があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。agent.properties ファイルには、エージェントの動作を変更するプロパティーを含めることもできます。

MFT リファレンス

MFT セキュリティー・リファレンス

IBM MQ の MFT でのファイル・システムの権限

IBM MQ の Managed File Transfer コンポーネントをインストールして構成すると、以下のような権限を持つディレクトリー `configuration`、`installations`、および `logs` が作成されます。

UNIX および Linux



表 27. UNIX および Linux でのディレクトリーの権限の要約

ディレクトリー	権限
/var/mqm/mqft/config	<ul style="list-style-type: none"> • mqm グループによる書き込みが可能 • すべてのユーザーが読み取り可能 mqm グループのユーザーにはこれらのディレクトリーおよびファイルに対する書き込み権限があります。
/var/mqm/mqft/installations	<ul style="list-style-type: none"> • mqm グループによる書き込みが可能 • すべてのユーザーが読み取り可能
/var/mqm/mqft/logs	すべてのユーザーが読み取り可能および書き込み可能

Windows

Windows

表 28. Windows でのディレクトリーの権限の要約

ディレクトリー	権限
MQ_DATA_PATH\mqft\config	以下のユーザーには読み取りおよび書き込みの全アクセス権限があります。 <ul style="list-style-type: none"> • 管理者 • システム・アカウント • mqm グループ その他のユーザーには読み取り権限があります。
MQ_DATA_PATH\mqft\installations	以下のユーザーには読み取りおよび書き込みの全アクセス権限があります。 <ul style="list-style-type: none"> • 管理者 • システム・アカウント • mqm グループ その他のユーザーには読み取り権限があります。
MQ_DATA_PATH\mqft\logs	以下のユーザーには読み取りおよび書き込みの全アクセス権限があります。 <ul style="list-style-type: none"> • 管理者 • システム・アカウント • mqm グループ その他のユーザーには読み取り権限および書き込み権限があります。

z/OS

z/OS

表 29. z/OS でのディレクトリーの権限の要約

ディレクトリー	権限
DATA_PATH/mqft/config	<ul style="list-style-type: none"> mqm グループ、または環境変数 BFG_GROUP_NAME で名前を指定されたグループは書き込み可能 すべてのユーザーが読み取り可能 <p>mqm グループ内のユーザー、または環境変数 BFG_GROUP_NAME の値で指定されるグループは、これらのディレクトリーおよびファイルに対する書き込み権限があります。</p>
DATA_PATH/mqft/installations	<ul style="list-style-type: none"> mqm グループ、または環境変数 BFG_GROUP_NAME で名前を指定されたグループは書き込み可能 すべてのユーザーが読み取り可能
DATA_PATH/mqft/logs	すべてのユーザーが読み取り可能および書き込み可能

注: DATA_PATH は、環境変数 BFG_DATA から導出されます。

MFT 固有リソースの権限の管理

どのファイル転送要求でも、Managed File Transfer エージェント・プロセスでは、ローカル・ファイル・システムに対する一定レベルのアクセス権限が必要になります。さらに、エージェント・プロセスと関連付けられているユーザー ID と、ファイル転送操作を実行しているユーザーと関連付けられているユーザー ID の両方は、特定の IBM MQ オブジェクトを使用する権限を持つ必要があります。

コマンドは、ユーザーによって発行されます。ユーザーは操作ロールで、通常、ファイル転送を開始します。あるいは、管理ロールのユーザーであれば、エージェントの作成、開始、削除、またはクリア（つまり、すべてのエージェント・システム・キューからのメッセージ除去）などの追加管理が可能です。コマンド要求を含むメッセージは、ユーザーがコマンドを発行するときにエージェントの SYSTEM.FTE.COMMAND キューに置かれます。エージェント・プロセスは、コマンド要求を含むメッセージを SYSTEM.FTE.COMMAND キューから取り出します。また、エージェント・プロセスは、他に、以下の 4 つのシステム・キューも使用します。

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

コマンドを発行するユーザーは、上記のキューをエージェント・プロセスにおいてさまざまな方法で使用するため、ユーザー ID または各 ID に関連付けられたユーザー・グループには、異なる IBM MQ 権限を割り当てます。詳細については、[353 ページの『MFT 固有リソースのグループ権限の制限』](#)を参照してください。

さらに、エージェントは、特定のアクションの実行権限をユーザーに付与するために使用できるキューを持っています。権限キューの使用方法については、[『359 ページの『MFT エージェント・アクションのユーザー権限の制限』』](#)を参照してください。エージェントは、これらのキューに対してメッセージの PUT または GET を行いません。ただし、エージェント・プロセスの実行に使用されるユーザー ID と、特定のアクションの実行が許可されているユーザーと関連付けられているユーザー ID の両方に、キューに対する正しい IBM MQ 権限が割り当てられていることを確認する必要があります。権限キューは以下のとおりです。

- 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

7.0.2 より前のバージョンの Managed File Transfer から IBM WebSphere MQ 7.5 以降にマイグレーションし、既存のエージェント構成を保持する場合は、権限キューを手動で作成する必要があります。このキューを作成するには以下の MQSC コマンドを使用します。

```
DEFINE QLOCAL(authority_queue_name) DEFPRTY(0) DEFSOPT(SHARED) GET(ENABLED) MAXDEPTH(0) +
  MAXMSGL(0) MSGDLVSQ(PRIORITY) PUT(ENABLED) RETINTVL(99999999) SHARE NOTRIGGER +
  USAGE(NORMAL) REPLACE
```

さらに、エージェント・プロセスは、SYSTEM.FTE キューを使用して、調整キュー・マネージャーの SYSTEM.FTE トピックにメッセージをパブリッシュします。エージェント・プロセスが、ソース・エージェントまたは宛先エージェントのいずれのロールであるかにより、エージェント・プロセスでは、ファイルの読み取り、書き込み、更新、および削除権限が必要となる可能性があります。

IBM MQ Explorer を使用すると、IBM MQ オブジェクトの権限レコードを作成および変更できます。オブジェクトを右クリックして、次に「オブジェクト権限」>「権限レコードの管理」をクリックします。さらに、**setmqaut** コマンドを使用して、権限レコードを作成することもできます。これは『[setmqaut \(権限の付与または取り消し\) コマンド](#)』で説明されています。

関連資料

353 ページの『MFT 固有リソースのグループ権限の制限』

関連する可能性のあるさまざまなオブジェクトのすべてについて、個々のユーザーに権限を付与する代わりに、Managed File Transfer のアクセス制御を管理する目的で、2つのセキュリティ・グループ FTEUSER と FTEAGENT を構成します。これらのグループの作成と埋め込みは、IBM MQ 管理者が行う責任があります。管理者は、ここで説明される構成の提案を拡大または変更する選択ができます。

359 ページの『MFT エージェント・アクションのユーザー権限の制限』

グループを使用してリソースに対するアクセスを管理することに加えて、追加レベルのセキュリティを使用可能にして、ユーザーが実行できる Managed File Transfer エージェント・アクションを制限することができます。エージェントの権限キューに対する権限をユーザーに付与して、ユーザーが特定のエージェント・アクションを実行できるようにします。

362 ページの『MFT ロガーの権限』

ロガーを実行するオペレーティング・システム・ユーザーには、ロガー・キューと SYSTEM.FTE トピックに対する特定の IBM MQ 権限が必要です。

MFT 固有リソースのグループ権限の制限

関連する可能性のあるさまざまなオブジェクトのすべてについて、個々のユーザーに権限を付与する代わりに、Managed File Transfer のアクセス制御を管理する目的で、2つのセキュリティ・グループ FTEUSER と FTEAGENT を構成します。これらのグループの作成と埋め込みは、IBM MQ 管理者が行う責任があります。管理者は、ここで説明される構成の提案を拡大または変更する選択ができます。

キュー・マネージャーに接続する権限

操作権限を持つユーザー、管理権限を持つユーザー、および IBM MQ エクスプローラーによって実行されるコマンドは、コマンド・キュー・マネージャーおよび調整キュー・マネージャーに接続できなければなりません。エージェント・プロセス、およびエージェントの作成、変更、または削除のために実行されるコマンドは、エージェント・キュー・マネージャーに接続できなければなりません。

- コマンド・キュー・マネージャーおよび調整キュー・マネージャーに接続する権限を FTEUSER グループに付与します。以下に例を示します。

ULW UNIX、Linux、および Windows システムの場合:

```
setmqaut -m command_queue_manager -t qmgr -g FTEUSER +connect
setmqaut -m coordination_queue_manager -t qmgr -g FTEUSER +connect
```

IBM i IBM i の場合:

```
GRTMQMAUT OBJ('command_queue_manager') OBJTYPE(*MQM) USER(FTEUSER) AUT(*CONNECT)
GRTMQMAUT OBJ('coordination_queue_manager') OBJTYPE(*MQM) USER(FTEUSER) AUT(*CONNECT)
```

z/OS z/OS の場合:

```
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)
```

- エージェント・キュー・マネージャーに接続および照会する権限を FTEAGENT グループに付与します。以下に例を示します。

ULW UNIX、Linux、および Windows システムの場合:

```
setmqaut -m agent_queue_manager -t qmgr -g FTEAGENT +connect +inq +setid
```

IBM i IBM i の場合:

```
GRTMQMAUT OBJ('agent_queue_manager') OBJTYPE(*MQM) USER(FTEAGENT) AUT(*CONNECT)
```

z/OS z/OS の場合:

```
RDEFINE MQCONN agent_queue_manager.BATCH UACC(NONE)
PERMIT agent_queue_manager.BATCH CLASS(MQCONN) ID(FTEAGENT) ACCESS(READ)
```

キュー・マネージャーに直接接続するコマンドに関する詳細については、[『367 ページの『MFT のコマンドおよびプロセスと、その接続先のキュー・マネージャー』』](#)を参照してください。

エージェントに属する COMMAND キューにメッセージを PUT するための権限

エージェントのコマンド・キューは、エージェントによるアクションの実行を要求する権限が与えられた、すべてのユーザーが使用できる必要があります。この要件を満たすには、次のようにします。

- FTEUSER グループに SYSTEM.FTE.COMMAND.agent_name キューへの PUT 権限のみを付与します。以下に例を示します。

ULW UNIX、Linux、および Windows システムの場合:

```
setmqaut -m QM1 -n SYSTEM.FTE.COMMAND.agent_name -t queue -g FTEUSER +put
```

IBM i IBM i の場合:

```
GRTMQMAUT OBJ('SYSTEM.FTE.COMMAND.agent_name') OBJTYPE(*Q) USER(FTEUSER) AUT(*PUT)
MQMNAME('QM1')
```

z/OS z/OS の場合:

```
RDEFINE MQQUEUE QM1.SYSTEM.FTE.COMMAND.agent_name UACC(NONE)
PERMIT QM1.SYSTEM.FTE.COMMAND.agent_name CLASS(MQQUEUE) ID(FTEUSER) ACCESS(UPDATE)
```

- FTEAGENT グループに、SYSTEM.FTE.COMMAND.agent_name キューへの PUT、GET、および setid アクセス権限を付与します。以下に例を示します。

ULW**UNIX、Linux、および Windows システムの場合:**

```
setmqaut -m QM1 -n SYSTEM.FTE.COMMAND.agent_name -t queue -g FTEAGENT +browse +put +get +setid
```

IBM i**IBM i の場合:**

```
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**z/OS の場合:**

```
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)
```

エージェントには、他のエージェントのコマンド・キューにメッセージを書き込むためのアクセス権限が必要です。リモート・キュー・マネージャーに接続しているエージェントがあれば、チャンネルからそのキューにメッセージを書き込むために追加の権限を付与しなければならない場合があります。

エージェントに属する DATA、STATE、EVENT、および REPLY キューにメッセージを PUT するための権限

Managed File Transfer エージェントのみがこれらのシステム・キューを使用できるようにする必要があるため、グループ FTEAGENT に PUT、GET、および INQUIRE 権限を付与します。これらのシステム・キューの名前は以下のとおりです。

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

例えば、SYSTEM.FTE.DATA.agent_name キューに対して、次のようなコマンドを使用します。

ULW**UNIX、Linux、および Windows システムの場合:**

```
setmqaut -m QM1 -n SYSTEM.FTE.DATA.agent_name -t queue -g FTEAGENT +put +get +inq
```

IBM i**IBM i の場合:**

```
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**z/OS の場合:**

```
RDEFINE MQQUEUE QM1.SYSTEM.FTE.DATA.agent_name UACC(NONE)
PERMIT QM1.SYSTEM.FTE.DATA.agent_name CLASS(MQQUEUE) ID(FTEAGENT) ACCESS(UPDATE)
```

エージェントには、他のエージェントのデータ・キューと応答キューにメッセージを書き込むためのアクセス権限が必要です。リモート・キュー・マネージャーに接続しているエージェントがあれば、チャンネルからそれらのキューにメッセージを書き込むために追加の権限を付与しなければならない場合があります。

エージェント・プロセスが実行される権限

エージェント・プロセスが実行される権限は、エージェントがファイル・システムからの読み取りと書き込みができるファイルと、エージェントがアクセスできるキューおよびトピックに影響します。権限がどのように構成されるかは、システムに依存しています。エージェント・プロセスが実行されるユーザー ID を FTEAGENT グループに追加します。グループへのユーザー ID の追加の詳細については、「[セキュリティのセットアップ](#)」を参照し、ご使用のオペレーティング・システムに関する情報までナビゲートしてください。

コマンドと IBM MQ Explorer が実行される権限

fteStartAgent コマンドなどの管理コマンド、および IBM MQ Explorer の Managed File Transfer プラグインは、SYSTEM.FTE.COMMAND.agent_name キューに入れられ、そのキューからパブリッシュされた情報を取得します。コマンドまたは IBM MQ Explorer を実行する権限を付与されたユーザー ID を FTEUSER グループに追加します。この発信元ユーザー ID は、転送ログに記録されます。グループへのユーザー ID の追加の詳細については、「[セキュリティのセットアップ](#)」を参照し、ご使用のオペレーティング・システムに関する情報までナビゲートしてください。

SYSTEM.FTE キューおよび SYSTEM.FTE トピックにメッセージを PUT する権限

エージェント・プロセスのみが SYSTEM.FTE キューおよび SYSTEM.FTE トピックにメッセージを PUT できなければなりません。SYSTEM.FTE キューに対する PUT、GET、および INQUIRE 権限を FTEAGENT グループに付与し、SYSTEM.FTE トピックに対するパブリッシュおよびサブスクライブ権限を FTEAGENT グループに付与します。以下に例を示します。

ULW

UNIX、Linux、および Windows システムの場合:

```
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

IBM i の場合:

```
GRTRMQMAUT OBJ('SYSTEM.FTE') OBJTYPE(*Q) USER(FTEAGENT) AUT(*PUT) MQMNAME('QM1')
GRTRMQMAUT OBJ('SYSTEM.FTE') OBJTYPE(*Q) USER(FTEAGENT) AUT(*GET) MQMNAME('QM1')
GRTRMQMAUT OBJ('SYSTEM.FTE') OBJTYPE(*TOPIC) USER(FTEAGENT) AUT(*PUB) MQMNAME('QM1')
GRTRMQMAUT OBJ('SYSTEM.FTE') OBJTYPE(*TOPIC) USER(FTEAGENT) AUT(*SUB) MQMNAME('QM1')
```

z/OS

z/OS の場合:

```
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)
```

リモート・キュー・マネージャーに接続しているエージェントがあれば、チャンネルから SYSTEM.FTE キューにメッセージを書き込むために追加の権限を付与しなければならない場合もあります。

メッセージが SYSTEM.FTE トピックにパブリッシュされるには、SYSTEM.FTE トピックの権限レコードによって、メッセージのメッセージ記述子構造 (MQMD) に含まれるユーザー ID によるパブリケーションを許可する必要があります。[ログ・メッセージおよび状況メッセージをパブリッシュする権限](#)を参照してください。

z/OS 上の SYSTEM.FTE トピックにパブリッシュする許可をユーザーに与えるには、SYSTEM.FTE トピックにパブリッシュするアクセス権限をチャンネル・イニシエーターのユーザー ID に付与する必要があります。RESLEVEL セキュリティー・プロファイルに基づいて、チャンネル・イニシエーター接続に関して 2 つのユーザー ID が検査される場合、メッセージのメッセージ記述子構造 (MQMD) に含まれているユーザー ID にもアクセス権限を付与する必要があります。詳しくは、[RESLEVEL セキュリティー・プロファイル](#)を参照してください。

SYSTEM.FTE トピックでパブリケーションを受信する権限

ログ・メッセージ、進行メッセージ、および状況メッセージの転送は、一般的な用途を目的としているため、FTEUSER グループに SYSTEM.FTE トピックのサブスクライブ権限を付与します。以下に例を示します。

ULW UNIX、Linux、および Windows システムの場合:

```
setmqaut -m QM1 -n SYSTEM.FTE -t topic -g FTEUSER +sub
```

IBM i IBM i の場合:

```
GRTMQMAUT OBJ('SYSTEM.FTE') OBJTYPE(*TOPIC) USER(FTEUSER) AUT(*SUB) MQMNAME('QM1')
```

z/OS z/OS の場合:

```
RDEFINE MXTOPIC QM1.SUBSCRIBE.SYSTEM.FTE UACC(NONE)  
PERMIT QM1.SUBSCRIBE.SYSTEM.FTE CLASS(MXTOPIC) ID(FTEUSER) ACCESS(ALTER)
```

伝送キューを使用してリモート・キュー・マネージャーに接続する権限

複数キュー・マネージャーのトポロジーでは、エージェントは、リモート・キュー・マネージャーへの接続に使用する伝送キューに対して PUT 権限を必要とします。

ファイル転送のために一時応答キューを作成する権限

ファイル転送要求は、転送の完了を待機し、作成されて埋め込まれる一時応答キューに依存します。FTEUSER グループに、一時モデル・キュー定義に対する DISPLAY、PUT、GET、および BROWSE 権限を付与します。以下に例を示します。

ULW UNIX、Linux、および Windows システムの場合:

```
setmqaut -m QM1 -n SYSTEM.DEFAULT.MODEL.QUEUE -t queue -g FTEUSER +dsp +put +get +browse
```

IBM i IBM i の場合:

```
GRTMQMAUT OBJ('SYSTEM.DEFAULT.MODEL.QUEUE') OBJTYPE(*Q) USER(FTEUSER) AUT(*ADMDSP)  
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 z/OS の場合:

```
RDEFINE MQQUEUE QM1.SYSTEM.DEFAULT.MODEL.QUEUE UACC(NONE)  
PERMIT QM1.SYSTEM.DEFAULT.MODEL.QUEUE CLASS(MQQUEUE) ID(FTEUSER) ACCESS(UPDATE)
```

デフォルトでは、このキューは SYSTEM.DEFAULT.MODEL.QUEUE ですが、command.properties ファイルの 'modelQueueName' と 'dynamicQueuePrefix' プロパティに値を設定することにより、名前を構成することができます。

z/OS では、一時キューにアクセスする権限を FTEUSER に付与することも必要です。以下に例を示します。

```
RDEFINE MQQUEUE QM1.WMQFTE.** UACC(NONE)  
PERMIT QM1.WMQFTE.** CLASS(MQQUEUE) ID(FTEUSER) ACCESS(UPDATE)
```

デフォルトでは、z/OS 上の各一時キューの名前の先頭に WMQFTE が付きます。

以下の表は、セキュリティ・スキームで説明した FTEUSER と FTEAGENT のアクセス制御構成を要約しています。

表 30. FTEUSER および FTEAGENT のアクセス制御構成の要約

オブジェクト	オブジェクト・タイプ	FTEUSER	FTEAGENT
エージェント・キュー・マネージャー	キュー・マネージャー		CONNECT、INQ、および SETID。また、 <u>ユーザー権限検査</u> を使用可能にするために、ALT_USER が必要です。
調整キュー・マネージャー	キュー・マネージャー		
コマンド・キュー・マネージャー	キュー・マネージャー	CONNECT	CONNECT
SYSTEM.FTE	ローカル・キュー		GET および PUT
SYSTEM.FTE.COMMAND.agent_name	ローカル・キュー	PUT	BROWSE、GET、PUT、および SETID
SYSTEM.FTE.DATA.agent_name	ローカル・キュー		GET および PUT
SYSTEM.FTE.EVENT.agent_name	ローカル・キュー		BROWSE、GET および PUT
SYSTEM.FTE.REPLY.agent_name	ローカル・キュー		GET および PUT
SYSTEM.FTE.STATE.agent_name	ローカル・キュー		BROWSE、GET、INQ、および PUT
SYSTEM.FTE	ローカル・トピック	SUBSCRIBE	PUBLISH および SUBSCRIBE
SYSTEM.DEFAULT.MODEL.QUEUE (または一時応答キューを作成するために使用される Managed File Transfer で定義したモデル・キュー。)	モデル・キュー	BROWSE、DISPLAY、GET、および PUT	BROWSE、DISPLAY、GET、および PUT
リモート・キュー・マネージャーと通信するための伝送キュー	ローカル・キュー		PUT

IBM MQ Explorer を使用して転送を管理するための権限

このページで説明した状況にあるユーザーに MFT 権限を付与することに加え、IBM MQ Explorer を使用してすべての MFT 操作を管理して実行する MFT エージェントのユーザーには、さらに権限を付与する必要があります。IBM MQ Explorer ユーザーがファイル転送の作成、キャンセル、スケジュールや、リソース・モニターの作成や削除、さらに転送テンプレートの作成などのコマンドを実行するには、以下の権限が必要です。

- 調整キュー・マネージャー: connect、inquire、display
- コマンド・キュー・マネージャー: connect、inquire、display
- SYSTEM.FTE トピック: publish、subscribe
- SYSTEM.MQEXPLORER.REPLY.MODEL: display、inquire、get、browse、put
- SYSTEM.ADMIN.COMMAND.QUEUE: inquire、put、display
- SYSTEM.DEFAULT.MODEL.QUEUE: get、put、inquire、display、browse

キュー・マネージャーに直接接続するコマンドに関する詳細については、『367 ページの『MFT のコマンドおよびプロセスと、その接続先のキュー・マネージャー』』を参照してください。

関連資料

359 ページの『MFT エージェント・アクションのユーザー権限の制限』

グループを使用してリソースに対するアクセスを管理することに加えて、追加レベルのセキュリティーを使用可能にして、ユーザーが実行できる Managed File Transfer エージェント・アクションを制限することができます。エージェントの権限キューに対する権限をユーザーに付与して、ユーザーが特定のエージェント・アクションを実行できるようにします。

362 ページの『MFT ロガーの権限』

ロガーを実行するオペレーティング・システム・ユーザーには、ロガー・キューと SYSTEM.FTE トピックに対する特定の IBM MQ 権限が必要です。

MFT エージェント・アクションのユーザー権限の制限

グループを使用してリソースに対するアクセスを管理することに加えて、追加レベルのセキュリティーを使用可能にして、ユーザーが実行できる Managed File Transfer エージェント・アクションを制限することができます。エージェントの権限キューに対する権限をユーザーに付与して、ユーザーが特定のエージェント・アクションを実行できるようにします。

ユーザー権限管理の使用可能化

エージェント・アクションに対するユーザー権限検査を有効にするには、以下のステップを実行します。

1. `agent.properties` ファイルで、`authorityChecking` 値を `true` に設定します。詳しくは、[550 ページの『MFT agent.properties ファイル』](#)を参照してください。
2. エージェントを実行するユーザーが、エージェント・キュー・マネージャーに対して IBM MQ 代替ユーザー (`ALT_USER`) 権限を持っていることを確認してください。

 z/OS プラットフォームでは、エージェントを実行するユーザーに、エージェント・アクションを実行する許可を要求できるユーザー ID に対する `ALT_USER` 権限が付与されていないことにご注意ください。

転送に関与する両方のエージェントで同レベルのセキュリティーが有効になっている必要があります。つまり、両方のエージェントのプロパティ・ファイルの `authorityChecking` は同じ値に設定されている必要があります。`authorityChecking` プロパティが異なる値に設定されているエージェント間の転送は失敗します。

エージェント権限キュー

エージェントには、どのユーザーが特定のエージェント・アクションを実行するための権限を持つのかを管理する権限キューがあります。エージェントは、これらのキューに対してメッセージの `PUT` または `GET` を行いません。エージェント権限キューは以下のとおりです。

- `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`

エージェント・プロパティ `authorityChecking=true` を設定してユーザー権限管理を有効にすると、ユーザーがエージェント権限キューに対して持っている権限によって、ユーザーが実行権限を持つアクションが決まります。

重要:   IBM MQ 9.0.0 Fix Pack 3 および IBM MQ 9.0.4 より、すべてのエージェント権限キューで `inquire` が必須の権限になりました。

以下の表は、エージェントの権限キューに対して特定のアクションを実行するために `inquire` 権限に加えてユーザーまたはグループが必要とする IBM MQ アクセス権限を要約したものです。

表 31. ユーザーまたはグループがエージェントの権限キューに対して特定のアクションを実行する必要がある IBM MQ アクセス権限のレベル

ユーザー処置	Managed File Transfer のアクセス権限	権限キュー	IBM MQ のアクセス権限 (Multiplatforms)	RACF のアクセス・レベル (z/OS のみ)
-m オプションを指定した fteStopAgent コマンドによるエージェントのシャットダウン	管理	SYSTEM.FTE.AUTHADM1.agent_name	BROWSE	READ
このエージェントからのファイル転送の開始	転送ソース	SYSTEM.FTE.AUTHTRN1.source_agent_name	BROWSE	READ
このエージェントで管理対象呼び出しを実行します				
同じユーザーが開始したこのエージェントからのファイル転送の取り消し				
このエージェントへのファイル転送の開始	転送宛先	SYSTEM.FTE.AUTHTRN1.destination_agent_name	PUT	UPDATE
同じユーザーが開始したこのエージェントへのファイル転送の取り消し				
リソース・モニターの作成	モニター	SYSTEM.FTE.AUTHMON1.monitor_agent_name	BROWSE	READ
同じユーザーが作成したリソース・モニターの削除				
任意のユーザーが作成したリソース・モニターの削除	モニター操作	SYSTEM.FTE.AUTHOPS1.agent_name	SET	ALTER
スケジュール済み転送の作成	スケジュール	SYSTEM.FTE.AUTHSCH1.source_agent_name	BROWSE	READ
同じユーザーによって作成されたスケジュール済み転送を削除する				
任意のユーザーまたはグループによって作成されたスケジュール済み転送の削除	スケジュール操作	SYSTEM.FTE.AUTHOPS1.agent_name	PUT	UPDATE
転送を開始したのと同じユーザーやグループ、または別のユーザーやグループによって作成された転送のキャンセル	転送操作	SYSTEM.FTE.AUTHOPS1.source_agent_name SYSTEM.FTE.AUTHOPS1.destination_agent_name	BROWSE	READ

注: ユーザーまたはグループに、転送を開始するリソース・モニターまたはスケジュール済み転送をセットアップする権限を付与するには、モニター または スケジュール 権限と、転送 ソースおよび 転送 宛先の両方の権限が必要です。

IBM MQ 9.0.0 Fix Pack 3 および IBM MQ 9.0.4 より、エージェントがファイル転送のキャンセル要求を受け取った時に実行する権限検査の順序が変わり、キャンセルを要求したユーザーが転送を開始したユーザーと同じかどうかを最初に検査するようになりました。転送をキャンセルするユーザーと転送を要求したユーザーが同じでない場合、エージェントはキャンセルするユーザーに、SYSTEM.FTE.AUTHOPS1.agent_name キューの browse 権限があるかどうかを検査します。この検査順序の変更によって、ファイル転送を要求したユーザーとキャンセルを要求したユーザーが同じ場合に、エージェントとキュー・マネージャーのエラー・ログで予期しないエラーが発生するのを回避できるようになります。

ユーザーは1つのエージェントを開始でき、そのエージェントから別のエージェントとの対話を行うことができます。2つのエージェントの対話方法は、ユーザーがもう一方のエージェント権限キューに対して持っているアクセス権限のレベルに応じて異なります。

エージェント・アクション	Managed File Transfer のアクセス権限	権限キュー	IBM MQ のアクセス権限 (Multiplatforms)	RACF のアクセス・レベル (z/OS のみ)
転送の受信 source_agent	エージェント・ソース	SYSTEM.FTE.AUTHAGT1.source_agent_name	BROWSE	READ
転送の送信先 destination_agent	エージェント宛先	SYSTEM.FTE.AUTHAGT1.destination_agent_name	PUT	UPDATE

ユーザー権限管理の構成

ユーザーにエージェントに対するアクションの実行権限を与えるには、関係する権限キューに対する適切な権限をユーザーに付与します。ユーザーに権限を付与するには、以下のステップを実行します。

1. エージェント・キュー・マネージャーがあるシステムに、エージェント・アクションの実行権限を与えるユーザーと同じ名前のユーザーを作成します。このユーザーはアクティブでなくても構いません。
2. 関係する権限キューに対する適切な権限をユーザーに付与します。Linux、UNIX、または Windows を使用している場合は、`setmqaut` コマンドを使用できます。
3. キュー・マネージャーのセキュリティー構成をリフレッシュします。`REFRESH SECURITY MQSC` コマンドを使用できます。

例

z/OS `setmqaut` コマンドは、z/OS システムでは使用されません。z/OS では、代わりに RACF を使用してください。詳しくは、[z/OS でのセキュリティーのセットアップ](#)を参照してください。

IBM i `setmqaut` コマンドは、IBM i システムでは使用されません。IBM i の場合は、『[IBM MQ オブジェクトに対するアクセス権限](#)』を参照してください。このトピックでは、IBM MQ オブジェクトの許可を実行する方法を説明しています。IBM i では、**Grant MQ Object Authority (GRMQMAUT)**、**Revoke MQ Object Authority (RVKMQMAUT)**、および **Refresh MQ Authority (RFRMQMAUT)** の 3 つの関連 CL コマンドを使用できます。

requestor_group グループのメンバーであるユーザーが、ファイルを AGENT1 から AGENT2 に転送する AGENT1 にリソース・モニターをセットアップするとします。AGENT1 を実行するユーザーは user1 であり、user1_group グループのメンバーです。AGENT2 を実行するユーザーは user2 であり、user2_group グループのメンバーです。AGENT1 は QM1 に接続し、AGENT2 は QM2 に接続します。両方のエージェントで権限検査が有効になっています。これを実現するには、以下のステップを実行します。

1. `requestor` は、AGENT1 に対する **モニター** 権限を持つ必要があります。この権限を設定するには、QM1 が稼働中のシステムで次のコマンドを実行します。

```
setmqaut -m QM1 -t queue -n SYSTEM.FTE.AUTHMON1.AGENT1 -g requestor_group +browse
```

2. `requestor` は、AGENT1 に対する **転送ソース** 権限を持つ必要があります。この権限を設定するには、QM1 が稼働中のシステムで次のコマンドを実行します。

```
setmqaut -m QM1 -t queue -n SYSTEM.FTE.AUTHTRN1.AGENT1 -g requestor_group +browse
```

3. `requestor` は、AGENT2 に対する **転送宛先** 権限を持つ必要があります。この権限を設定するには、QM2 が稼働中のシステムで次のコマンドを実行します。

```
setmqaut -m QM2 -t queue -n SYSTEM.FTE.AUTHTRN1.AGENT2 -g requestor_group +put
```

4. `user2` は、AGENT1 に対する **エージェント・ソース** 権限を持つ必要があります。この権限を設定するには、QM1 が稼働中のシステムで次のコマンドを実行します。

```
setmqaut -m QM1 -t queue -n SYSTEM.FTE.AUTHAGT1.AGENT1 -g user2_group +browse
```

5. `user1` は、AGENT2 に対する **エージェント宛先** 権限を持つ必要があります。この権限を設定するには、QM2 が稼働中のシステムで次のコマンドを実行します。

```
setmqaut -m QM2 -t queue -n SYSTEM.FTE.AUTHAGT1.AGENT2 -g user1_group +put
```

ロギング

ユーザー権限検査が有効である場合、権限検査が失敗すると、非許可ログ・メッセージが調整キュー・マネージャーに対してパブリッシュされます。詳しくは、『[809 ページの『セキュリティーの MFT メッセージ・フォーマット』](#)』を参照してください。

ユーザー権限に関するメッセージはエージェント・イベント・ログへ書き込めます。エージェント・イベント・ログに書き込まれる情報量は、エージェント・プロパティ・ファイルで `logAuthorityChecks` プロパティを設定することにより構成できます。デフォルトでは、権限検査ロギングのレベルは `None` です。また、`logAuthorityChecks` の値は `Failures` に設定できます。この場合、失敗した許可検査のみ報告されることを示します。また、`All` に設定すると、失敗した許可検査と成功した許可検査が報告されることを示します。

詳しくは、[550 ページの『MFT agent.properties ファイル』](#)を参照してください。

関連資料

[353 ページの『MFT 固有リソースのグループ権限の制限』](#)

関連する可能性のあるさまざまなオブジェクトのすべてについて、個々のユーザーに権限を付与する代わりに、Managed File Transfer のアクセス制御を管理する目的で、2つのセキュリティー・グループ `FTEUSER` と `FTEAGENT` を構成します。これらのグループの作成と埋め込みは、IBM MQ 管理者が行う責任があります。管理者は、ここで説明される構成の提案を拡大または変更する選択ができます。

[362 ページの『MFT ロガーの権限』](#)

ロガーを実行するオペレーティング・システム・ユーザーには、ロガー・キューと `SYSTEM.FTE` トピックに対する特定の IBM MQ 権限が必要です。

[531 ページの『fteStopAgent』](#)

`fteStopAgent` コマンドは、Managed File Transfer のエージェントを制御された仕方で停止させるか、必要であれば `-i` パラメーターを使用してエージェントを即時に停止させるために使用します。

MFT ロガーの権限

ロガーを実行するオペレーティング・システム・ユーザーには、ロガー・キューと `SYSTEM.FTE` トピックに対する特定の IBM MQ 権限が必要です。

ロガーを実行するオペレーティング・システム・ユーザーには、以下の IBM MQ 権限が必要です。

- 調整キュー・マネージャーに対する CONNECT および INQUIRE 権限。
- SYSTEM.FTE トピックに対する SUBSCRIBE 権限。
- SYSTEM.FTE.LOG.RJCT.logger_name キューに対する PUT 権限。
- SYSTEM.FTE.LOG.CMD.logger_name キューに対する GET 権限。

関連資料

353 ページの『MFT 固有リソースのグループ権限の制限』

関連する可能性のあるさまざまなオブジェクトのすべてについて、個々のユーザーに権限を付与する代わりに、Managed File Transfer のアクセス制御を管理する目的で、2つのセキュリティー・グループ FTEUSER と FTEAGENT を構成します。これらのグループの作成と埋め込みは、IBM MQ 管理者が行う責任があります。管理者は、ここで説明される構成の提案を拡大または変更する選択ができます。

359 ページの『MFT エージェント・アクションのユーザー権限の制限』

グループを使用してリソースに対するアクセスを管理することに加えて、追加レベルのセキュリティーを使用可能にして、ユーザーが実行できる Managed File Transfer エージェント・アクションを制限することができます。エージェントの権限キューに対する権限をユーザーに付与して、ユーザーが特定のエージェント・アクションを実行できるようにします。

機密構成情報にアクセスするための MFT アクセス権

機密構成情報を保管するために使用するファイル (つまり IBM MQ 構成ツリーから参照されるファイル) に対する読み取り権限、書き込み権限、または削除権限 (可能な場合) は、システム全体に対して付与してはなりません。この制限は、トラストストア・ファイルと鍵ストア・ファイルにも当てはまります。

構成ファイルに機密情報が含まれていて、構成ファイルが鍵ストア・ファイルまたはトラストストア・ファイルであり、構成ファイルに対する読み取り権限、書き込み権限、または削除権限がシステム全体に付与されている状態を Managed File Transfer プロセスが検出すると、プロセスは以下のいずれかのアクションを実行します。

- 状態が起動時に検出された場合は、開始に失敗する。
- 状態が実行時に検出された場合は、警告メッセージを生成し、構成ファイルの内容を無視する。これは、プロセスの実行中に構成が変更された場合に構成を再ロードする、プロトコル・ブリッジおよび Connect:Direct ブリッジに関係します。

ファイル・システムが UNIX タイプのシステムの場合

ファイルに対して許容できないシステム全体にわたる権限が設定されているかどうかを判別するための基準は以下のとおりです。

- その他クラスに、ファイルに対する読み取り権限が付与されている
- その他クラスに、ファイルに対する書き込み権限が付与されている
- その他クラスに、ファイルが含まれているディレクトリーに対する書き込み権限が付与されている

Windows システムの場合:

ファイルに対して許容できないシステム全体にわたる権限が設定されているかどうかを判別するための基準は以下のとおりです。

- 全員グループ、ゲスト・グループ、またはユーザー・グループには以下の権限を付与できます。
 - ファイルに対するデータ読み取り権限
 - ファイルに対するデータ追加権限
 - ファイルに対するデータ書き込み権限
- 全員グループ、ゲスト・グループ、またはユーザー・グループには、ファイルが含まれているフォルダーに対するファイル作成権限が付与されており、以下の権限も付与できます。
 - ファイルが含まれているフォルダーに対するサブフォルダーおよびファイルの削除権限
 - ファイルに対する削除権限

MFT エージェント・ログ・メッセージおよび状況メッセージをパブリッシュする権限

Managed File Transfer エージェントは、調整キュー・マネージャーにパブリッシュされるさまざまな ログ・メッセージ、進行メッセージ、および状況メッセージを発行します。これらのメッセージのパブリケーションは、IBM MQ セキュリティー・モデルの影響を受け、場合によっては、パブリケーションを可能にするために、さらに構成が必要になることがあります。

IBM MQ のセキュリティの詳細については、[保護から始まるセクション](#)を参照してください。

Managed File Transfer エージェントは、調整キュー・マネージャー上の SYSTEM.FTE キューへのパブリケーションのためのメッセージを出します。個々のメッセージは、メッセージ記述子 (MQMD) にユーザー ID を含みます。メッセージは SYSTEM.FTE とも呼ばれるトピック・オブジェクトを使用してパブリッシュされます。指定されたメッセージのパブリケーションを実行するには、SYSTEM.FTE トピックの権限レコードが、メッセージの MQMD に含まれるユーザー ID によるパブリケーションを許可する必要があります。

z/OS では、チャンネル・イニシエーターのユーザー ID に、SYSTEM.FTE トピックにパブリッシュするためのアクセス権限が必要です。[RESLEVEL セキュリティー・プロファイル](#)により 2 つのユーザー ID でチャンネル・イニシエーター接続の検査が行われる場合、メッセージの MQMD に含まれるユーザー ID にもこのトピックにパブリッシュするためのアクセス権限が必要になります。

当初メッセージに含まれるユーザー ID は、エージェントが自身のキュー・マネージャーにどのように接続しているかにより異なります。バインディング接続エージェントからのメッセージには、エージェントが動作する際のユーザー ID が含まれます。クライアント接続エージェントからのメッセージには、IBM MQ の内部ユーザー ID が含まれます。

メッセージ内のユーザー ID は変更することができます。クライアント接続エージェントおよびバインディング接続エージェントの両方で、`publicationMDUser` プロパティ (`agent.properties` ファイル内) を使用して、そのエージェントからのすべてのログ・メッセージと状況メッセージで使用されるユーザー ID を指定することができます。この代替ユーザー ID を使用するには、自身のキュー・マネージャーによってエージェントに権限が付与されなければなりません。この権限を付与するには、エージェントが動作する際のユーザー ID に `setid` 権限を付与します。

さらに、エージェントが自身のキュー・マネージャーに接続するために使用するチャンネルで、`MCAUSER` プロパティを使用して、クライアント接続エージェントからのすべてのメッセージに含まれるユーザー ID を変更することもできます。

調整キュー・マネージャーにメッセージを組み込む受信側チャンネルなどのチャンネル出口を使用して、メッセージ内のユーザー ID を変更することができます。

IBM MQ トポロジーおよびポリシーによっては、IBM MQ 管理者がこのトピックの情報を使用して、状況メッセージとログ・メッセージのパブリケーションを確実に実行する方法が数多くあります。以下に 2 つの例を示します。

- エージェントで使用されるすべてのユーザー ID をネットワークで決定します。これらの ID にそれぞれ権限レコードを明示的に付与します。
- ログ・メッセージと状況メッセージをパブリッシュするための 1 つ以上の共通ユーザー名を作成します。調整キュー・マネージャーに、これらのユーザー名の権限レコードを作成します。各エージェントの `publicationMDUser` プロパティを共通ユーザー名に設定します。各エージェントのキュー・マネージャーで、`publicationMDUser` プロパティを受け入れることができるように、エージェント実行時に使用するユーザー ID に `setid` 権限を付与します。

ファイル・システムにアクセスするための MFT 権限

どのファイル転送要求でも、Managed File Transfer エージェント・プロセスでは、ローカル・ファイル・システムに対する一定レベルのアクセス権限が必要になります。

- ソース・ファイルから転送するには、ソース・エージェントが動作する際のユーザー ID に、ソース・ファイルへの読み取り権限が必要です。さらに、ソースの後処理属性によっては、ソース・エージェントに削除または書き込み権限を付与する必要がある場合もあります。

- ファイルまたはディレクトリーへの転送では、宛先エージェントを実行するユーザー ID に、指定のパスへの書き込み権限が必要です。さらに、宛先出口属性によっては、宛先エージェントに更新権限を付与する必要がある場合もあります。
- エージェント・プロセスに付与するファイル・アクセス権限に加えて、サンドボックスを使用して、制限付きファイル・パス領域を指定してこれを実施することも可能です。詳細内容は [を参照してください](#)。
- 転送先または転送元のファイルが、エージェントからアクセスできる場所がない場合 (例えば、 VSAM データ・セットである場合や、サンドボックス機能によって制限されている場所にある場合) は、Managed File Transfer のユーザー出口を使用して、エージェントがアクセスできる場所にそのファイルを移動できます。詳細内容は [を参照してください](#)。

commandPath MFT プロパティ

commandPath プロパティを使用して、Managed File Transfer がコマンドを実行できる場所を指定します。指定された commandPath にある任意のコマンドは、エージェントにコマンドを送信できるリモート・クライアント・システムから事実上呼び出すことができるため、このプロパティを設定するときは十分な注意を払ってください。

エージェントが実行されているシステム上で、Managed File Transfer の管理対象転送および管理対象呼び出し機能から実行するコマンドを指定できます。詳しくは、[プログラム呼び出し](#)を参照してください。ただし、コマンドは、commandPath エージェント・プロパティで参照されるパス上にある必要があります。

指定されたコマンドが完全修飾されていない場合、Managed File Transfer はコマンド・パス上で一致するコマンドを見つけようとします。コマンド・パス上に一致するコマンドが2つ以上ある場合、最初の一致が使用されます。

デフォルトでは、commandPath プロパティは空であるため、エージェントはどのコマンドも呼び出すことができません。

commandPath エージェント・プロパティを以下のように指定します。

```
commandPath=command_directory_name
separator...command_directory_name
```

 また、z/OS の場合のみ、次のように指定します。

```
commandPath=command_directory_name_or_data_set_name_prefix
separator...command_directory_name_or_data_set_name_prefix
```

ここで、

- `command_directory_name` は、実行できるコマンドのディレクトリー・パスです。
-  `command_directory_name_or_data_set_name_prefix` は、実行できるコマンドの z/OS UNIX システム・サービス・ディレクトリー・パス、またはデータ・セット名接頭部です (先頭に // を付けます)。データ・セット名接頭部では、完全修飾の形式を使用するか、非修飾の形式を使用するか (つまり、// 'HLQ...' という形式にするか、//HLQ... という形式にするか) を選択できます。区分データ・セットを // 'HLQ()...' または //HLQ()... の形式で指定します。データ・セットを使用するのは、JCL スクリプト・コマンドを指定する場合のみです。
- `separator` は、プラットフォーム固有の分離文字です。

 例えば、UNIX システムで、ディレクトリー /home/user/cmds1 および /home/user/cmds2 にあるコマンドを実行する場合、commandPath エージェント・プロパティを次のように設定します。

```
commandPath=/home/user/cmds1:/home/user/cmds2
```

Windows 例えば、Windows システムで、ディレクトリー C:\File Transfer\commands および C:\File Transfer\agent commands にあるコマンドを実行する場合、commandPath エージェント・プロパティを次のように設定します。

```
commandPath=C:\\File Transfer\\commands;C:\\File Transfer\\agent commands
```

Windows システム上では、区切り文字のバックスラッシュ (\) をエスケープして、ダブルバックスラッシュ (\\) として入力する必要があります。バックスラッシュはスラッシュ (/) で置き換えることもできます。

z/OS 例えば、z/OS で、次のようなコマンドを実行する場合、

- ディレクトリー /home/user/cmds1 および /home/user/cmds2 にある
- //'USER.CMD1', //CMD2, で始まるデータ・セットにある
- //'USER.CMDS' という完全修飾された PDS のメンバーである

次のようにして、commandPath エージェント・プロパティを設定します。

```
commandPath=/home/user/cmds1:/home/user/cmds2:/'USER.CMD1':/CMD2:/'USER.CMDS()'
```

重要: このプロパティを設定するときは細心の注意を払う必要があります。指定されたいずれかの commandPath にあるコマンドはどれも、エージェントにコマンドを送信できるリモート・クライアント・システムから呼び出し可能になるからです。このため、デフォルトでは、commandPath を指定すると、すべての commandPath ディレクトリー (およびそれらのサブディレクトリー) における転送のためのアクセスが自動拒否されるように、サンドボックスが構成されます。

- エージェント・サンドボックスを使用するようにエージェントが構成されている場合、エージェントの開始時に、commandPath ディレクトリーが拒否ディレクトリーのリストに自動的に追加されます。
- 1 つ以上のユーザー・サンドボックスを使用するようにエージェントが構成されている場合、エージェントの開始時に、各ユーザー・サンドボックスの <read> エlement および <write> エlement に commandPath ディレクトリーが <exclude> エlement として追加されます。
- エージェントがエージェント・サンドボックスとユーザー・サンドボックスのいずれかを使用するようには構成されていない場合、エージェントの開始時に、新しいエージェント・サンドボックスが作成され、その中で commandPath ディレクトリーが拒否ディレクトリーとして指定されます。

V9.0.0.1 以下のリリースとの互換性が必要な場合は、この動作をオーバーライドできます。

- IBM WebSphere MQ File Transfer Edition.
- IBM WebSphere MQ 7.5.0 Fix Pack 1 Managed File Transfer コンポーネント (またはそれ以前)。
- インストール・プロパティ enableFunctionalFixPack=7502 が設定されていないインストール済み環境での IBM WebSphere MQ 7.5.0 Fix Pack 2 Managed File Transfer コンポーネント (またはそれ以降)。

この動作をオーバーライドするには、agent.properties ファイルに次のプロパティを追加します。

```
addCommandPathToSandbox=false
```

V9.0.0.1 addCommandPathToSandbox プロパティが存在して false に設定されていると、以下のような動作になります。

- エージェント・サンドボックスを使用するようにエージェントが構成されていて、サンドボックスに許可ディレクトリーが指定されていない場合、エージェントが開始すると、commandPath ディレクトリーが拒否ディレクトリーのリストに自動的に追加されます。
- エージェント・サンドボックスを使用するようにエージェントが構成されていて、サンドボックスに許可ディレクトリーが 1 つ以上指定されている場合、エージェントが開始しても、commandPath ディレクトリーは拒否ディレクトリーのリストに追加されません。

- エージェントが1つ以上のユーザー・サンドボックスを使用して構成されている場合、ユーザー・サンドボックスは変更されず、commandPath ディレクトリーは、<exclude>エレメントとして、各ユーザー・サンドボックスの<read>エレメントおよび<write>エレメントに追加されません。
- エージェントがエージェント・サンドボックスとユーザー・サンドボックスのいずれかを使用するには構成されていない場合、エージェントの開始時に、新しいエージェント・サンドボックスが作成され、その中で commandPath ディレクトリーが拒否ディレクトリーとして指定されます。

関連資料

550 ページの『MFT agent.properties ファイル』

各 Managed File Transfer Agent には、独自のプロパティー・ファイル agent.properties があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。agent.properties ファイルには、エージェントの動作を変更するプロパティーを含めることもできます。

MFT のコマンドおよびプロセスと、その接続先のキュー・マネージャー

Managed File Transfer トポロジーは、さまざまなコンポーネントによって構成されます。

それらのコンポーネントは、以下のとおりです。

- 1つ以上のエージェントと、それに関連付けられたエージェント・キュー・マネージャー
- 調整キュー・マネージャー
- コマンド・キュー・マネージャー
- トポロジーの管理やマネージド転送の送信に使用されるいくつかのコマンド。
- トポロジー内のエージェントによって実行されるマネージド転送に関する情報を収集する、オプションのロガー。
- IBM MQ Explorer Managed File Transfer プラグイン。一部の管理用タスクを実行し、マネージド転送に関する情報を表示するために使用できます

エージェント、ロガー、コマンド、および IBM MQ Explorer Managed File Transfer プラグインは、実行時に1つ以上のキュー・マネージャーに接続します。

以下の表は、実行時にエージェント、ロガー、コマンド、および IBM MQ Explorer Managed File Transfer プラグインがどのキュー・マネージャーに接続するかを示しています。

表内の「X」印がないコマンドまたはプロセスは、実行されたときに、どのキュー・マネージャーまたはプロセスにも接続しません。

コマンド名	エージェント・キュー・マネージャー	コマンド・キュー・マネージャー	調整キュー・マネージャー	ロガー・キュー・マネージャー
fteAnt				
fteCancelTransfer		X		
fteChangeDefaultConfigurationOptions				
fteCleanAgent	X			
fteCreateAgent	X			
fteCreateBridgeAgent	X			
fteCreateCDAgent	X			
V9.0.1 fteCreateEnvironment				
fteCreateLogger				
fteCreateMonitor		X		
fteCreateTemplate			X	
fteCreateTransfer		X		

表 33. Managed File Transfer コマンドとその接続先のキュー・マネージャーの要約 (続き)				
コマンド名	エージェント・キュー・マネージャー	コマンド・キュー・マネージャー	調整キュー・マネージャー	ロガー・キュー・マネージャー
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				
 fteSetAgentLogLevel				
fteSetAgentTraceLevel				
fteSetLoggerTraceLevel				
 fteSetProductID				
fteSetupCommands				
fteSetupCoordination				
fteShowAgentDetails			X	
fteShowLoggerDetails				
fteStartAgent				
fteStartLogger				
fteStopAgent		X		
fteStopLogger		X		

表 34. Managed File Transfer プロセスとその接続先のキュー・マネージャーの要約				
Processes	エージェント・キュー・マネージャー	コマンド・キュー・マネージャー	調整キュー・マネージャー	ロガー・キュー・マネージャー
Managed File Transfer エージェント	X			
IBM MQ Explorer に対応した Managed File Transfer プラグイン		X	X	

表 34. Managed File Transfer プロセスとその接続先のキュー・マネージャーの要約 (続き)

Processes	エージェント・キュー・マネージャー	コマンド・キュー・マネージャー	調整キュー・マネージャー	ロガー・キュー・マネージャー
Managed File Transfer ロガー			X	X

各タイプのキュー・マネージャー (つまり、エージェント・キュー・マネージャー、コマンド・キュー・マネージャー、調整キュー・マネージャー) に接続するのに必要な資格情報が入っているファイルは、関連するプロパティ・ファイルで指定できます。例えば、調整キュー・マネージャーには `coordination.properties` ファイルがあります。そのファイルで、資格情報ファイルを参照するように `coordinationQMgrAuthenticationCredentialsFile` プロパティを設定できます。

調整キュー・マネージャーに接続するコマンドは、そのファイルで指定されている資格情報を使用します。キュー・マネージャーでセキュリティが有効になっていて、このプロパティが正しく設定されていない場合は、MFT コマンドが正常に完了しません。詳しくは、[MFT と IBM MQ の接続認証](#) を参照してください。

関連情報

[インストール済みの MFT コマンド・セット](#)

MFT コマンドとその接続先のキュー・マネージャーの詳細

Managed File Transfer コマンドとその接続先のキュー・マネージャーの詳細情報。

このトピックでは、[367 ページの『MFT のコマンドおよびプロセスと、その接続先のキュー・マネージャー』](#) の情報について、いくつかの図を使用してさらに詳しく説明します。

調整キュー・マネージャーに接続するコマンド

以下のコマンドは、調整キュー・マネージャーに接続します。

- [fteCreateTemplate](#)
- [fteDeleteAgent](#)
- [fteDeleteTemplates](#)
- [fteListAgents](#)
- [fteListMonitors](#)
- [fteListScheduledTransfers](#)
- [fteListTemplates](#)
- [fteShowAgentDetails](#)

Managed File Transfer トポロジーの調整キュー・マネージャーは、トポロジー全体に関する情報を持つ中央のハブです。調整キュー・マネージャーは、送信側チャンネルと受信側チャンネルを介して、トポロジー内のすべてのエージェント・キュー・マネージャーに接続されます。エージェントは定期的にステータス情報を調整キュー・マネージャーに公開し、そこに転送テンプレートを保管します。

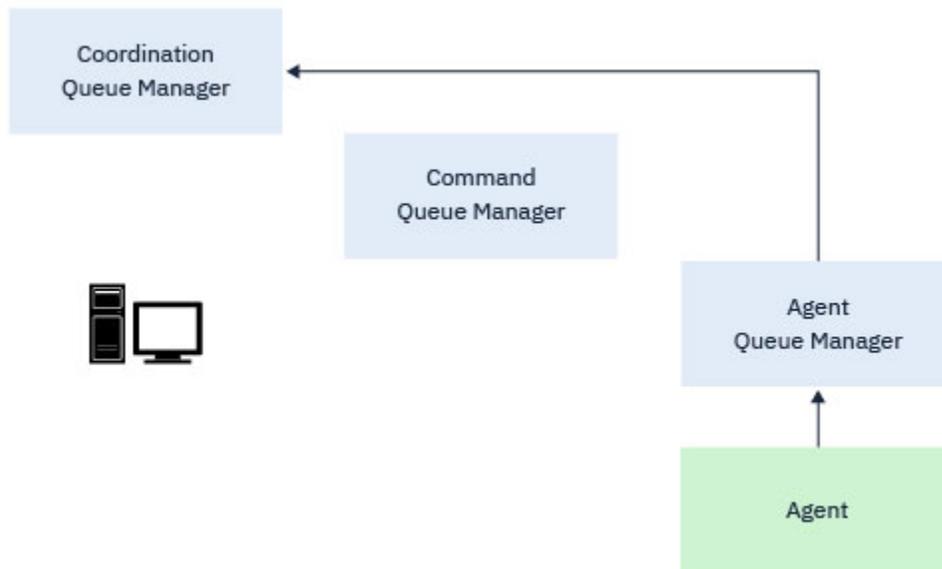


図 9. エージェントは、調整キュー・マネージャーで状況情報を公開したり、転送テンプレートを保管したりします。

調整キュー・マネージャーに接続する前述のコマンドのいずれかが実行されると、それらは調整キュー・マネージャーに直接接続され、以下のいずれかを実行します。

- 転送テンプレートを作成または削除します。
- エージェント、モニター、またはスケジュール済み転送に関する状態情報を照会し、その情報をユーザーに表示します。

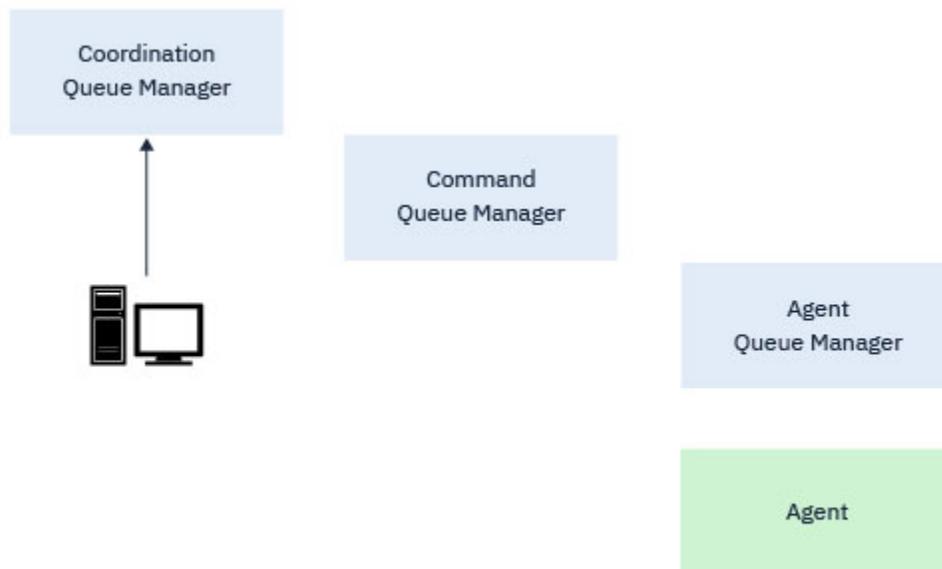


図 10. コマンドは、調整キュー・マネージャーに直接接続して、適切な状況情報を取得するか、転送テンプレートを処理します。

コマンド・キュー・マネージャーに接続するコマンド

以下のコマンドは、コマンド・キュー・マネージャーに接続します。

- [fteCancelTransfer](#)

- [fteCreateMonitor](#)
- [fteCreateTransfer](#)
- [fteDeleteMonitor](#)
- [fteDeleteScheduledTransfer](#)
- [ftePingAgent](#)
- [fteStopAgent](#)

コマンド・キュー・マネージャーは、Managed File Transfer トポロジーへのゲートウェイの一形態として考えることができます。送信側チャンネルと受信側チャンネルを使用してエージェント・キュー・マネージャーに接続されます。

キュー・マネージャーに接続する前述のコマンドのいずれかが実行されると、コマンドは以下を実行します。

- コマンド・キュー・マネージャーに接続します。
- 一時応答キューを作成します。
- コマンドの詳細を含むメッセージを適切なエージェントに送信します。

メッセージは IBM MQ ネットワークを経由してエージェント・キュー・マネージャーに経路指定されます。そこでメッセージはエージェントによって取り出され、処理されます。

エージェントがコマンドを処理した後、エージェントはコマンド・キュー・マネージャーに応答を送り返します。ここで、応答はコマンドによって取得されます。

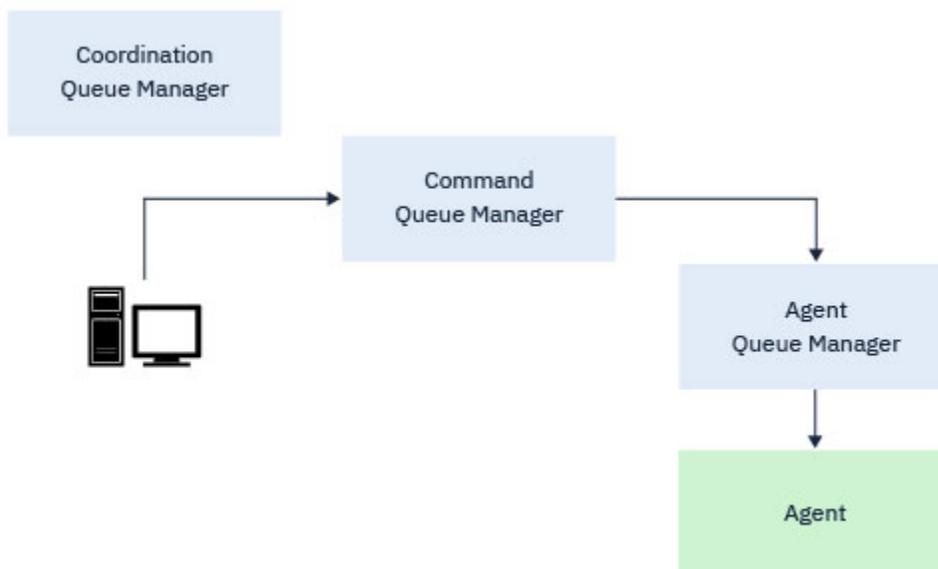


図 11. コマンドがコマンド・キュー・マネージャーに接続します。コマンドを含むメッセージが、IBM MQ ネットワークを介して正しいエージェント・キュー・マネージャーに経路指定されます。そこでメッセージがエージェントによって取り出されます。

エージェント・キュー・マネージャーに接続するコマンド

以下のコマンドは、エージェント・キュー・マネージャーに接続します。

- [fteCleanAgent](#)
- [fteCreateAgent](#)
- [fteCreateBridgeAgent](#)
- [fteCreateCDAgent](#)
- [fteDeleteAgent](#)

各エージェントには、独自のエージェント・キュー・マネージャーがあります。エージェントは、このキュー・マネージャーでホストされているシステム・キューを使用して、状態情報を維持し、作業要求を受け取ります。

1つのキュー・マネージャーが、複数のエージェントのエージェント・キュー・マネージャーとして機能できます。エージェント・キュー・マネージャーは、送信側チャンネルと受信側チャンネルを使用して調整キュー・マネージャー、コマンド・キュー・マネージャー、および他のエージェント・キュー・マネージャーに接続されます。

fteCreateAgent、**fteCreateBridgeAgent**、および **fteCreateCDAgent** の各コマンドは、エージェント・キュー・マネージャー名を引数として使用します。

これらのコマンドが実行されると、そのコマンドは指定されたキュー・マネージャーに接続し、エージェントが Managed File Transfer トポロジーに追加されたことを示すメッセージを調整キュー・マネージャーに送信します。

同様に、**fteDeleteAgent** が実行されると、エージェント・キュー・マネージャーに接続し、調整キュー・マネージャーにメッセージを送信し、エージェントが Managed File Transfer トポロジーから除去されたことを通知します。

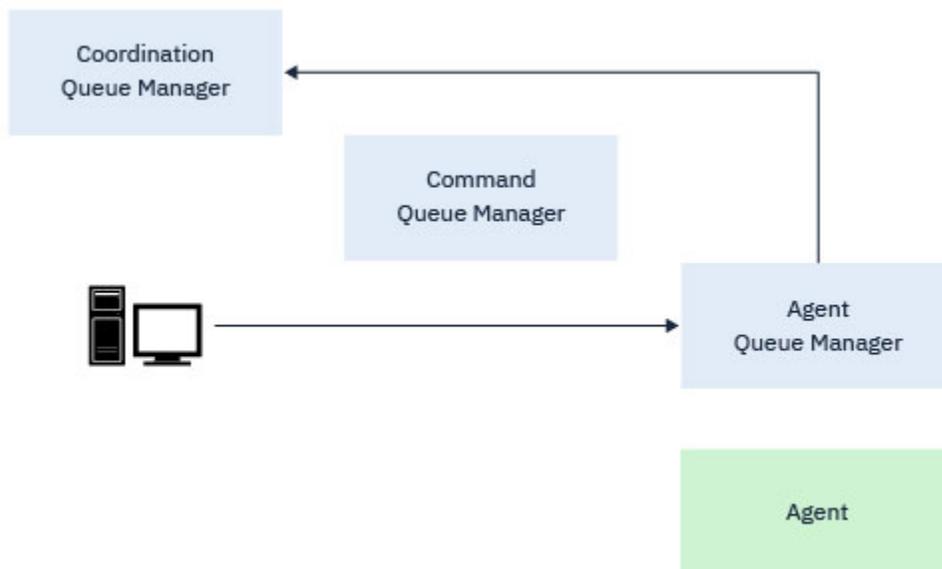


図 12. さまざまな **fteCreate** コマンドと **fteDeleteAgent** コマンドが、エージェント・キュー・マネージャーに接続し、調整キュー・マネージャーにメッセージを送信して、エージェントをトポロジーに登録またはトポロジーから登録解除します。

重要: **fteCleanAgent** はエージェント・キュー・マネージャーに接続し、そのエージェントの状態情報をそのシステム・キューから除去します。

このコマンドを実行すると、トポロジー全体に影響が及ぶ可能性があります。そのため、このコマンドは IBM の指示のもとでのみ行ってください。

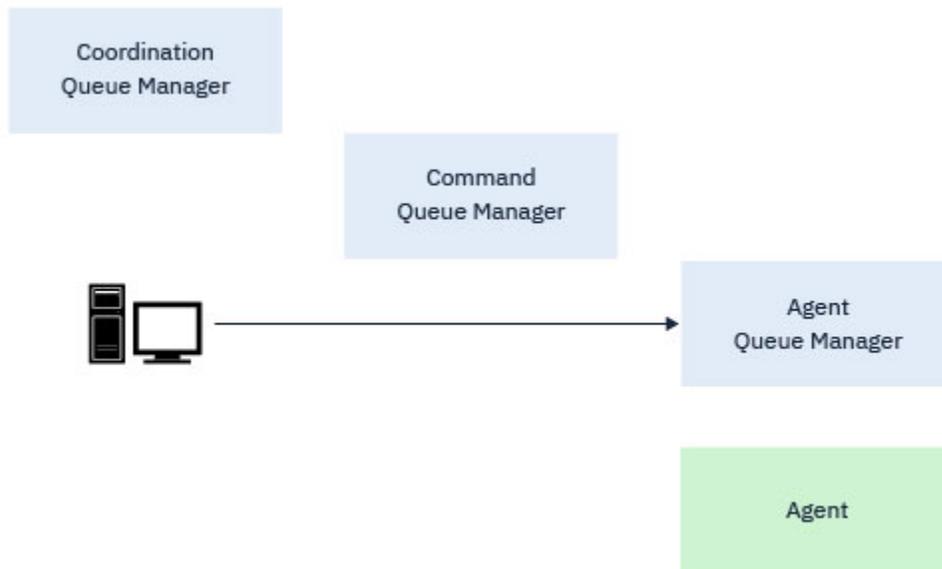


図 13. **fteCleanAgent** コマンドは、エージェント・キュー・マネージャーに直接接続し、エージェントのシステム・キューから状態情報を除去します。

関連情報

[インストール済みの MFT コマンド・セット](#)

MFT コマンド

Managed File Transfer のすべてのコマンドがリストされており、その詳しい説明へのリンクも掲載されています。

表 35. Managed File Transfer コマンドとその目的	
コマンド名	目的
マイグレーション用のコマンド:	
fteMigrateAgent	IBM WebSphere MQ File Transfer Edition 7.0.4 から Managed File Transfer 7.5 以降へのエージェントとその構成のマイグレーション
fteMigrateConfigurationOptions	IBM WebSphere MQ File Transfer Edition 7.0 構成を IBM WebSphere MQ 7.5 以降の Managed File Transfer にマイグレーションします。
fteMigrateLogger	スタンドアロン・データベース・ロガーの構成を IBM WebSphere MQ File Transfer Edition 7.0.1 以降から Managed File Transfer 7.5 以降にマイグレーションします。
構成用のコマンド:	
fteChangeDefaultConfigurationOptions	Managed File Transfer で使用したいデフォルトの構成オプションを変更する
fteCreateAgent	Managed File Transfer Agent を作成する
fteCreateBridgeAgent	Managed File Transfer プロトコル・ブリッジ・エージェントを作成する
fteCreateCDAgent	Managed File Transfer Connect:Direct ブリッジ・エージェントを作成する
V 9.0.1 fteCreateEnvironment	Redistributable Managed File Transfer Agent のファイルの構成および転送のための環境変数を設定します。
fteCreateLogger	Managed File Transfer ロガーを作成する

表 35. Managed File Transfer コマンドとその目的 (続き)	
コマンド名	目的
fteDefine	指定されたオブジェクトを定義するために必要な構成スクリプトを生成します。
fteDelete	指定されたオブジェクトを削除するために必要な構成スクリプトを生成します。
fteDeleteAgent	特定の Managed File Transfer Agent を削除する
fteDeleteLogger	Managed File Transfer ロガーを削除する
fteModifyAgent	Windows のみ。エージェント、Connect:Direct ブリッジ・エージェント、またはプロトコル・ブリッジ・エージェントを変更して、Windows サービスとして実行されるようにします。
fteModifyLogger	Windows のみ。ロガーを変更して、Windows サービスとして実行されるようにします。
fteSetupCommands	コマンドの発行時に IBM MQ ネットワークに接続するキュー・マネージャーの詳細を指定する
fteSetupCoordination	Managed File Transfer の調整キュー・マネージャーを構成する
管理用のコマンド:	
fteAnt	ファイル転送 Ant タスクが使用可能な環境で Ant スクリプトを実行する
fteCancelTransfer	ファイル転送を取り消す
fteCleanAgent	エージェントが使用するキューをクリーンアップする
fteCreateMonitor	新規リソース・モニターを作成および開始する
fteCreateTemplate	将来の使用のために転送テンプレートを作成する
fteCreateTransfer	新規ファイル転送を作成および開始する
fteDeleteMonitor	既存リソース・モニターを停止および除去する
fteDeleteScheduledTransfer	以前にスケジュールに入れた特定のファイル転送を削除する
fteDeleteTemplates	既存のファイル転送テンプレートを削除する
fteListAgents	特定の調整キュー・マネージャーに対して登録されたすべてのエージェントをリストする
fteListMonitors	特定の調整キュー・マネージャーに対して登録されたすべてのリソース・モニターをリストする
fteListScheduledTransfers	コマンド行または IBM MQ Explorer を使用して以前に作成した Managed File Transfer のすべての転送をリストする。
fteListTemplates	調整キュー・マネージャーのすべてのファイル転送テンプレートのリストを表示する
ftePingAgent	エージェントを ping して、そのエージェントがアクティブで、転送を処理できるかどうかを判別する
 fteSetProductID	z/OS SCRT レコーディング製品 ID を設定する
 fteShowAgentDetails	特定のエージェントの詳細を表示する
fteShowLoggerDetails	特定のロガーの詳細を表示する
fteStartAgent	ファイル転送に使用する前に特定のエージェントを開始する
fteStartLogger	ロガーを開始する
fteStopAgent	特定のエージェントを停止する
fteStopLogger	ロガーを停止する
セキュリティ用のコマンド:	
fteObfuscate	資格情報ファイル内の機密データを暗号化する

表 35. Managed File Transfer コマンドとその目的 (続き)	
コマンド名	目的
トラブルシューティング用のコマンド:	
<code>fteDisplayVersion</code>	製品のバージョンを表示する
<code>V9.0.0</code> <code>fteSetAgentLogLevel</code>	Managed File Transfer プロトコル・ブリッジ・エージェントと FTP/SFTP/FTPS ファイル・サーバーの間のファイル転送に関する診断情報のログギングを有効または無効にします。
<code>fteSetAgentTraceLevel</code>	実行するエージェント・トレースのレベルを設定する
<code>fteSetLoggerTraceLevel</code>	実行するロガー・トレースのレベルを設定する
<code>fteRAS</code>	RAS 収集ツールを実行する

インストールされる MFT コマンド・セットを参照し、Managed File Transfer の各オファリングでインストールされるコマンドを示す表を確認してください。

各コマンドとそのパラメーターの構文は、路線図と呼ばれる構文図の形式で示されています。路線図の解釈方法については、「[路線図の読み方](#)」を参照してください。

MFT コマンドの使用権限

Managed File Transfer コマンドを発行する場合には、ご使用のユーザー ID が mqm グループのメンバーでなければなりません (mqm グループに含まれないユーザーがコマンドを発行できるように既に IBM MQ を構成してある場合を除く)。

z/OS z/OS 上の mqm に代替グループを定義する方法について詳しくは、[IBM MQ for z/OS へのコマンドの発行](#)を参照してください。

権限について詳しくは、[IBM MQ を管理する権限](#)を参照してください。**IBM i** IBM i を使用している場合は、最初にトピック [IBM MQ 権限](#)を参照してください。

Managed File Transfer コマンドのサブセットは、IBM MQ エクスプローラーを使用して発行できます。

Windows および UNIX システムからのコマンドの発行

コマンドを発行する場合には、以下の環境固有の情報に注意してください。

Managed File Transfer for Windows

すべてのコマンドはコマンド行から実行できます。コマンド名には大/小文字の区別がありません。それらは大文字、小文字、または大文字と小文字の組み合わせのいずれでも入力できます。ただし、制御コマンドの引数 (キュー名など) およびパラメーター (キュー・マネージャー名の **-m** など) には、大/小文字の区別があります。

構文記述では、ハイフン (-) はフラグ標識として使用されます。

Managed File Transfer for UNIX システム

Managed File Transfer のすべてのコマンドは、シェルから発行できます。すべてのコマンドは、大/小文字が区別されます。

z/OS システムからのコマンドの発行

z/OS

Managed File Transfer コマンドは、製品のインストール時に選択された bin サブディレクトリーにインストールされます。コマンドは、以下のいずれかのオプションから実行できます。

- USS 環境から直接実行するには、コマンドにパスを指定するか、ユーザー・コマンド・パスに bin サブディレクトリーを含めます。
- 特定のエージェントまたはロガー用に PDSE コマンド・テンプレート・ライブラリーから構成した、コマンドの PDSE データ・セットから実行します。詳しくは、[63 ページの『MFT エージェントまたはロガーのコマンド・データ・セットの作成』](#)を参照してください。

IBM i プラットフォームからのコマンドの発行

IBM i

IBM i でコマンドを発行する場合には、以下の環境固有の情報に注意してください。

- Qshell インタープリターを使用して Managed File Transfer コマンドを開始できます。Qshell インタープリターを始動するには、IBM i システム・コマンド行から **STRQSH** コマンドを発行します。
- Qshell 環境でコマンドを実行する場合、コマンド名には大/小文字の区別がありません。それらは大文字、小文字、または大文字と小文字の組み合わせのいずれでも入力できます。ただし、制御コマンドの引数（キュー名など）およびパラメーター（キュー・マネージャー名の **-m** など）には、大/小文字の区別があります。

関連資料

338 ページの『MFT の戻りコード』

Managed File Transfer コマンド、Ant タスク、およびログ・メッセージは、機能が正常に実行されたかどうかを示す戻りコードを提供します。

路線 (構文) 図の読み方

各路線図は、右向き二重矢印で始まり、右向きと左向きの一対の矢印で終わります。単一の右矢印で始まる線は継続線です。路線図は、矢印の方向に従って、左から右へ、上から下へと読みます。

路線図で使用されるその他の規則は、以下のとおりです。

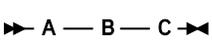
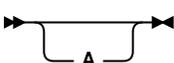
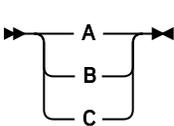
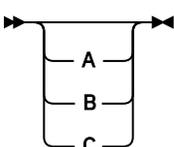
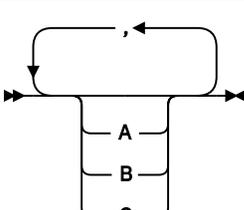
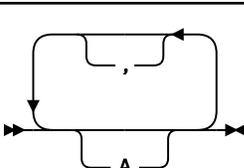
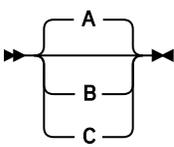
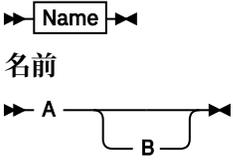
規則	意味
	値 A、B、および C を指定する必要があります。必須の値は、路線図の主線上に示されます。
	値 A を指定することができます。オプションの値は、路線図の主線の下に示されます。
	値 A、B、および C は選択肢であり、その 1 つを指定する必要があります。
	値 A、B、および C は選択肢であり、その 1 つを指定することができます。
	1 つ以上の値 A、B、および C を指定できます。複数の値を指定する場合、または同じ値を繰り返し指定する場合に必要な区切り記号（この例ではコンマ (,)）は、矢印の上を示されます。
	値 A を複数回指定できます。この例の区切り記号はオプションです。

表 36. 路線図の読み方 (続き)

規則	意味
	<p>値 A、B、および C は選択肢であり、その 1 つを指定することができます。示される値のどれも指定しない場合は、デフォルトの A (主線の上に表示されている値) が使用されます。</p>
 <p>名前</p>	<p>路線フラグメント Name は、主路線図とは別に示されます。</p>
<p>句読点および大文字の値</p>	<p>示されているとおりに指定します。</p>
<p>小文字の値 (例えば name)</p>	<p>name 変数の代わりにユーザー独自のテキストを入力します。</p>

MFT コマンドのトレース

コマンド行での問題判別に役立つために、Managed File Transfer のあらゆるコマンドをトレースできます。

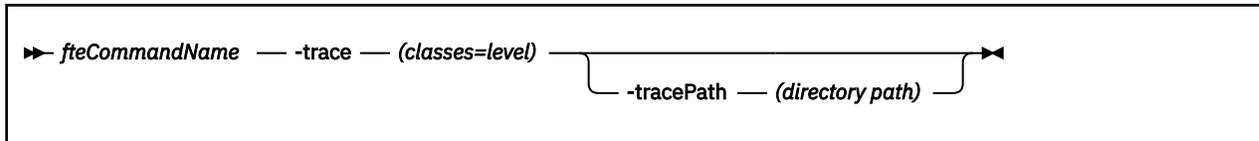
目的

指定されたレベルでトレースを有効にするには、任意の MFT コマンドに **-trace** パラメーターを使用します。生成されるトレース・ファイルは、別のディレクトリーを識別するために **-tracePath** パラメーターが含まれていない限り、現行作業ディレクトリーに配置されます。

トレースを実行すると、パフォーマンスに重大な影響を及ぼすおそれがあり、また大量のトレース・データが生成される可能性もあるため、トレースの実行は注意して、必要な場合にのみ実施してください。通常は、IBM サービス担当員からトレースを依頼された場合に限り、トレースを有効にします。

トレース・ファイルのサイズや保持するトレース・ファイルの数などの他のトレース・プロパティーを `agent.properties` ファイルに設定することができます。これらのプロパティーについては、『[拡張エージェント・プロパティー](#)』で説明されています。

構文



Parameters

-trace (classes=level)

必須。トレースを設定するレベル、およびトレースを適用するクラス。次のフォーマットで指定します。

```
classes=level
```

以下に例を示します。

```
com.ibm.wmqfte=all
```

この指定は Managed File Transfer のすべてのクラスをトレースします。

このレベルのトレースを適用する対象となる、コロンで区切られたクラス仕様のリストを指定します。このパラメーターが指定されない場合、トレース・レベルはすべてのエージェント・クラスに適用されます。

(classes) が正符号 (+) で始まる場合、現在トレースされている既存のトレース・クラスすべてに、正符号に続くトレース・クラスのリストが追加されます。

有効なトレース・レベル・オプションは以下のとおりです。トレース・ファイルのサイズと詳細度の昇順にリストしています。

off

エージェント・トレースをオフに切り替えますが、ログ・ファイルへの情報の書き込みは続行されます。これはデフォルト・オプションです。

flow

エージェント内でプロセッシング・フローに関連付けられたトレース・ポイントのデータを収集します。

moderate

トレースであまり多くない量の診断情報を収集します。

詳細

トレースで詳細な診断情報を収集します。

all

すべてのエージェント・クラスでエージェント・トレースを実行するように設定します。

-tracePath (directory path)

オプション。トレースの書き込み先にするディレクトリーを指定します。例えば、c:\temp などです。

 このパラメーターが指定されない場合、値はコマンド実行元のディレクトリーになります。例えば、z/OS では以下のとおりです。

 /u/smith/fte/wmqmft/mqft/logs/MQPV/loggers/BFGLG1/logs/

このパラメーターは、**-trace** パラメーターが指定されている場合にのみ有効です。

例

この例では、トレース・レベルを all に設定しています。これは、AGENT.NAME に属するすべてのクラスが **fteStartAgent** コマンドでトレースされることを意味します。

注：エージェントを開始すると、トレースが `mft_config/logscoordination_qmgr/agents/agent` に書き込まれます。

```
fteStartAgent -trace com.ibm.wmqfte=all -tracePath /u/mft/trace AGENT.NAME
```

この例では、エージェント AGENT.NAME の com.ibm.wmqfte.common クラスのトレース・レベルを moderate に設定します。 **ftePingAgent** コマンドで、普通程度の量のトレースが取り込まれることになります。

```
ftePingAgent -trace com.ibm.wmqfte.common=moderate AGENT.NAME
```

この例では、エージェント AGENT.NAME の com.ibm.wmqfte.common クラスのトレース・レベルを moderate に設定され、トレースは c:\\$user ディレクトリーに書き込まれます。 **ftePingAgent** コマンドで、普通程度の量のトレースが取り込まれることになります。

```
ftePingAgent -trace com.ibm.wmqfte.common=moderate -tracePath c:\$user AGENT.NAME
```

関連資料

270 ページの『[トラブルシューティング MFT](#)』

以下の参照情報を、Managed File Transfer のエラーの診断に役立ててください。

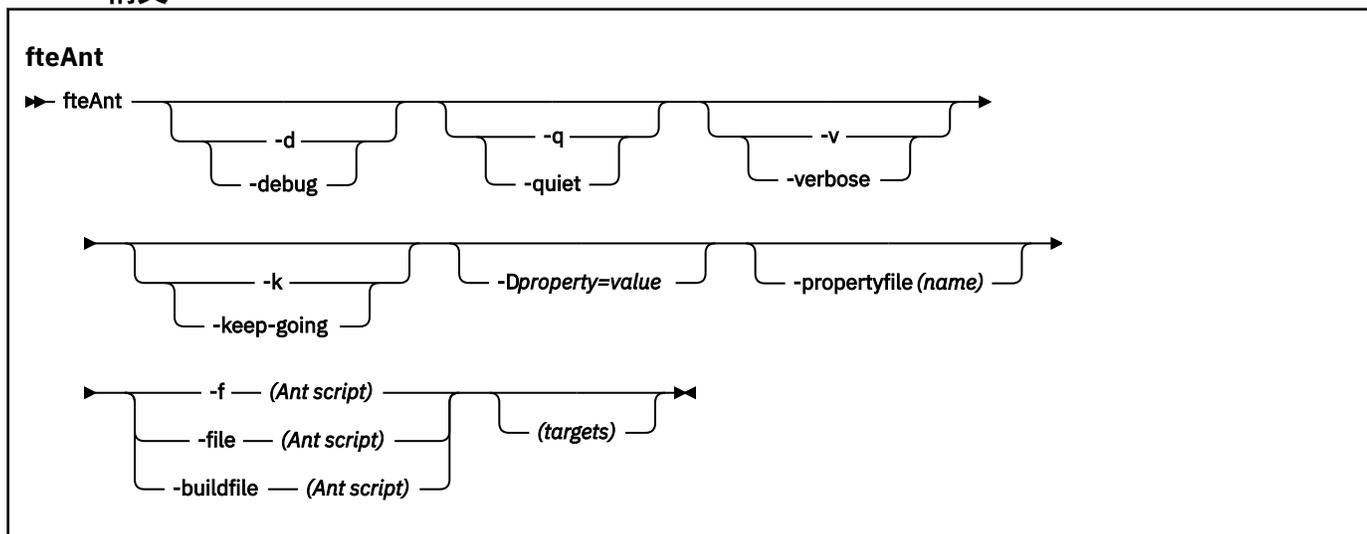
fteAnt: MFT で Ant タスクを実行します。

fteAnt コマンドは、Managed File Transfer の Ant タスクが使用可能な環境で Ant スクリプトを実行します。

目的

fteAnt コマンドは、Managed File Transfer がインストールされた環境で Ant スクリプトを実行する場合に使用します。標準の **ant** コマンドとは異なり、**fteAnt** ではスクリプト・ファイルを定義する必要があります。

構文



Parameters

-debug または **-d**

オプション。デバッグ出力を生成します。

-quiet または **-q**

オプション。最小出力を生成します。

-verbose または **-v**

オプション。詳細出力を生成します。

-keep-going または **-k**

オプション。失敗したターゲットに依存しないすべてのターゲットを実行します。

-D property=value

オプション。一定のプロパティの値を使用します。 **-D** で設定されたプロパティは、プロパティ・ファイル内に設定されたプロパティに優先します。

com.ibm.wmqfte.propertyset プロパティは、Ant タスクに使用される構成オプションのセットを指定するために使用します。このプロパティの値には、デフォルトでない調整キュー・マネージャーの名前を使用します。そうすることで、Ant タスクは、このデフォルトではない調整キュー・マネージャーに関連付けられた構成オプションのセットを使用します。このプロパティを指定しない場合、デフォルトの調整キュー・マネージャーに基づいたデフォルトの構成オプションのセットが使用されます。Ant タスクに対して **cmdqm** 属性を指定すると、**fteAnt** コマンド用に指定された構成オプションのセットよりもこの属性が優先されます。この振る舞いは、デフォルトの構成オプションのセットを使用するか、または **com.ibm.wmqfte.propertyset** プロパティでセットを指定するかにかかわらず適用されます。

-propertyfile (name)

オプション。すべてのプロパティを、優先順位を持つ **-D** プロパティが指定されたファイルからロードします。

-f (Ant スクリプト)、-file (Ant スクリプト)、または -buildfile (Ant スクリプト)

必須。実行する Ant スクリプトの名前を指定します。

targets

オプション。Ant スクリプトの実行元の 1 つ以上のターゲットの名前。このパラメーターに値を指定しない場合は、スクリプトのデフォルト・ターゲットが実行されます。

-バージョン

オプション。Managed File Transfer コマンドおよび Ant のバージョンを表示します。

-? または -h

オプション。コマンド構文を表示します。

例

この例では、Ant スクリプト `fte_script.xml` 内のターゲット **copy** が実行され、このコマンドにより、デバッグ出力が標準出力に書き込まれます。

```
fteAnt -d -f fte_script.xml copy
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

他の状況戻りコードは、Ant スクリプトからも、例えば、Ant fail タスクを使用するなどして指定できます。詳しくは、[失敗](#)を参照してください。

z/OS fteBatch、fteCommon および ftePlatform

z/OS fteBatch、fteCommon および ftePlatform は、Managed File Transfer によって `MQ_INSTALLATION_PATH/bin` ディレクトリーにヘルパー・スクリプトとして提供されるスクリプトです。どのプラットフォームでも、これらのスクリプトがすべて存在するわけではありません。

fteBatch スクリプト (z/OS のみ)

z/OS

fteBatch は、Managed File Transfer を JZOS バッチ・ランチャーから実行するためのヘルパー・スクリプトです。fteBatch は z/OS にのみインストールされます。通常、Managed File Transfer は提供されたコマンド・シェル・スクリプトを使用して開始されます。このスクリプトは、いくつかの環境構成を実行して

から、その機能に適した Java クラスを開始します。Managed File Transfer が JZOS バッチ・ランチャーを使用して開始されると、Java クラスがランチャーから直接開始されます。fteBatch は、必要なクラス名を環境変数に配置するためのランチャー・セットアップの一部として呼び出すことができ、Java を開始する前に通常のコマンド・シェル・スクリプトが実行するセットアップ作業を行います。これは、実行中のジョブと、Managed File Transfer によって使用される内部クラス名の間に分離レベルを設けます。

fteBatch コマンドは、Managed File Transfer 8.0 では推奨されません。新しい PDSE データ・セットのコマンドを使用して Managed File Transfer を実行することができます。詳しくは、[63 ページの『MFT エージェントまたはロガーのコマンド・データ・セットの作成』](#)を参照してください。

fteCommon

fteCommon は、Java を開始する前に共通のセットアップ処理を実行するために、その他の Managed File Transfer コマンド・スクリプトによって開始されるヘルパー・スクリプトです。

ftePlatform

ftePlatform は、fteCommon スクリプトによって開始されるヘルパー・スクリプトであり、プラットフォーム固有のセットアップ処理を実行します。

fteCancelTransfer

fteCancelTransfer コマンドは、Managed File Transfer 転送を取り消すために使用します。このコマンドは、転送のソース・エージェントまたは宛先エージェントに対して発行できます。

目的

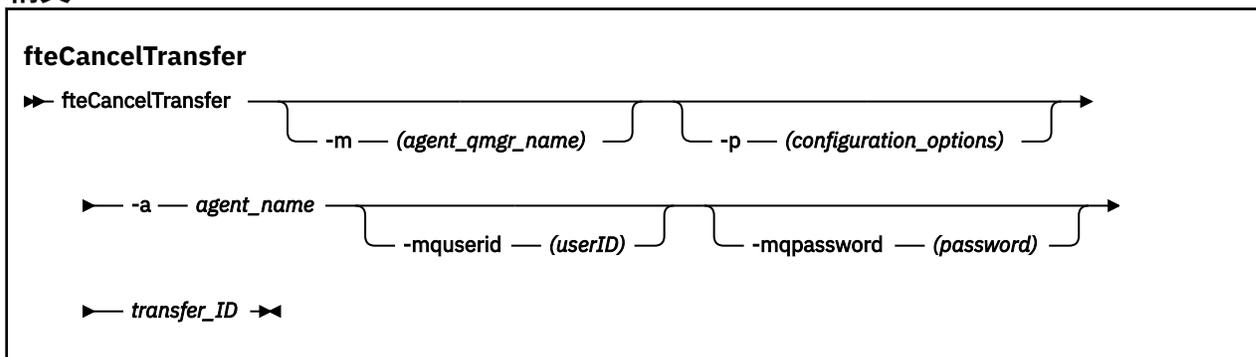
転送の進行中に **fteCancelTransfer** コマンドを発行した場合、その転送の一部として既に転送されたファイルは宛先システムに残り、削除されません。その転送の一部として部分的に転送されたファイルは、宛先システムからすべて削除されます。転送の宛先側では、その転送が「"取り消し済み"」としてログに記録されます。

Connect:Direct ノードへの転送が取り消されると、取り消された転送の一部として部分的に転送されたファイルは宛先システムに残り、削除されません。

fteCancelTransfer コマンドは、IBM MQ ネットワークに接続してからエージェントのキュー・マネージャーに経路指定のできる、任意のシステムから実行できます。特に、このコマンドを実行するには、このシステムに Managed File Transfer がインストール済みであること、および IBM MQ ネットワークと通信するために、このシステムで Managed File Transfer が構成済みであることが必要です。接続の詳細を使用できない場合、接続にそのエージェントのキュー・マネージャーの詳細が使用されます (詳細が使用可能な場合)。

オプションの **-p** パラメーターは、デフォルト・セットと異なる構成オプションのセットを使用する場合にのみこのコマンドに指定します。詳しくは、[構成オプション](#)を参照してください。

構文



Parameters

-m (agent_qmgr_name)

オプション。エージェントのキュー・マネージャーの名前。このエージェントは、取り消す転送のソース・エージェントまたは宛先エージェントのいずれかでなければなりません。このパラメーターを指定しない場合、使用している構成オプションのセットで識別されるキュー・マネージャーに取り消し要求が送信されます。

-p (configuration_options)

オプション。このパラメーターは、転送を取り消すために使用する構成オプションのセットを決定します。慣例として、このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティ・ファイルのセットを使用します。

このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。

-a (agent_name)

必須。取り消す転送のソース・エージェントまたは宛先エージェントいずれかの名前。

-mquserid (userID)

オプション。コマンド・キュー・マネージャーで認証するユーザー ID を指定します。

-mqpassword (password)

オプション。コマンド・キュー・マネージャーで認証するパスワードを指定します。-mquserid パラメーターも一緒に指定する必要があります。-mquserid を指定した場合に -mqpassword を指定しないと、関連付けるパスワードを入力するというプロンプトが出されます。パスワードは表示されません。

transfer_ID

必須。取り消す転送の ID。fteCreateTransfer コマンドが発行されると、転送 ID (要求 ID でもある) がコマンド行に表示されます。転送 ID は、ファイル転送ログ・メッセージにも組み込まれます。あるいは、IBM MQ エクスプローラーの「転送ログ」パネルにも表示されます。

-? または -h

オプション。コマンド構文を表示します。

例

この例では、AGENT1 が取り消す転送のソース・エージェントです。

```
fteCancelTransfer -a AGENT1 414d5120514d5f4c4d343336303920201159c54820027102
```

戻りコード

0

コマンドが正常に完了したか、あるいはエージェントには指定された転送 ID が不明であるかのいずれかです。転送 ID がエージェントに不明の場合、最も可能性の高い原因は、転送が既に完了済みであるか、またはキャンセル済みであることです。

1

コマンドは失敗しました。

関連資料

437 ページの『fteCreateTransfer: 新規ファイル転送の開始』

fteCreateTransfer コマンドは、コマンド行から新規のファイル転送を作成して開始します。このコマンドは、ファイル転送の即時開始、将来の日時におけるファイル転送のスケジュール、ファイル転送の 1 回以上の繰り返し、および特定の条件に基づくファイル転送の起動を実行できます。

fteChangeDefaultConfigurationOptions

fteChangeDefaultConfigurationOptions コマンドは、Managed File Transfer で使用するデフォルト構成オプションを変更するために使用します。構成オプションの値によって、Managed File Transfer で使用するプロパティ・ファイルのグループを定義します。

重要: **ULW** IBM MQ for UNIX, Linux, and Windows では、IBM MQ 管理者 (および mqm グループのメンバー) であるユーザーのみが、このコマンドを実行できます。IBM MQ 管理者以外のユーザーとしてこのコマンドを実行しようとすると、エラー・メッセージ BFGCL0502E: 要求された操作を実行する権限がありません。を受け取ります。コマンドは実行されません。

z/OS z/OS システムでは、マイグレーション・コマンドを実行するためには、ユーザーは以下の条件の (少なくとも) 1 つを満たす必要があります。

- mqm グループのメンバーである (システムに mqm グループが定義されている場合)。
- **V9.0.0.1** BFG_GROUP_NAME 環境変数で名前が指定されたグループのメンバーである (名前が指定されている場合)。
- **V9.0.0.1** コマンド実行時に BFG_GROUP_NAME 環境変数に値が設定されていない。

目的

デフォルトの Managed File Transfer 構成オプションは、初めて **fteSetupCoordination** コマンドを使用して調整キュー・マネージャーとしてキュー・マネージャーを構成するときに設定されます。MFT 製品のインストール時に、mqft ディレクトリーが (まだ存在しない場合) <MQ_DATA_PATH> の下に作成されます。また、構成、インストール、ログの各ディレクトリーも (まだ存在しない場合) mqft ディレクトリーの下に作成されます。

fteChangeDefaultConfigurationOptions コマンドを使用することにより、**installation.properties** ファイルで定義されたデフォルトの調整キュー・マネージャーを変更できます。この調整キュー・マネージャーを変更すると、Managed File Transfer は、デフォルトで **configuration_options** の入力として使用したディレクトリーに含まれる一連の構造化されたディレクトリーおよびプロパティ・ファイルによって指定される構成オプションを使用します。このディレクトリー名は、この構成の下でエージェントによって使用される調整キュー・マネージャーと同じです。

installation.properties ファイルについて詳しくは、57 ページの『MFT 構成オプション (Multiplatforms)』を参照してください。

構文

fteChangeDefaultConfigurationOptions

▶ **fteChangeDefaultConfigurationOptions** — *configuration_options* ◀

Parameters

configuration_options

必須。このパラメーターは、変更後のデフォルト構成オプションを指定します。このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。

-? または -h

オプション。コマンド構文を表示します。

例

次の例では、デフォルト構成オプションは QM_COORD2 に変更されます。

```
fteChangeDefaultConfigurationOptions QM_COORD2
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

関連概念

57 ページの『MFT 構成オプション (Multiplatforms)』

Managed File Transfer は、セットアップに関する重要な情報が含まれた、操作に必要ないくつかのプロパティ・ファイルを提供します。これらのプロパティ・ファイルは、製品のインストール時に定義される構成ディレクトリーにあります。

fteCleanAgent: MFT エージェントのクリーンアップ

fteCleanAgent コマンドを使用して、Managed File Transfer Agent が使用する永続的および非永続的キューからメッセージを削除することにより、エージェントが使用するキューをクリーンアップします。エージェントの開始に問題があり、エージェントが使用するキューに情報が残っていることが原因である可能性がある場合、**fteCleanAgent** コマンドを使用します。

目的

fteCleanAgent コマンドを使用して、エージェントが使用する永続的および非永続的キューからメッセージを削除します。具体的には、このコマンドは以下のアクションを実行できます。

- 転送が停止される前に進行中であったこのエージェントへのすべての転送またはこのエージェントからのすべての転送が除去されます。これらの転送は、エージェントが再始動しても再開されません
- エージェントにすでに実行依頼されているが、まだ実行されていないコマンドを除去します
- エージェント上に保管されているすべてのリソース・モニターが削除されます
- エージェント上に保管されているすべてのスケジュール済み転送が削除されます
- エージェント上に保管されているすべての無効なメッセージが削除されます

エージェントが Connect:Direct ブリッジ・エージェントの場合、パラメーター **-ms**、**-ss**、および **-ims** は無効です。Connect:Direct ブリッジ・エージェントの場合、このコマンドはさらに以下のアクションも実行します。

- Connect:Direct ブリッジ・エージェントがファイルの転送時にファイルを一時的に格納するディレクトリーからすべてのファイルを削除します。このディレクトリーの場所は、**cdTmpDir** パラメーターで定義します。
- 実行中の転送に関連した Connect:Direct プロセスに関する情報を表示します。

IBM MQ 9.0.3 および IBM MQ 9.0.0 Fix Pack 2 より前では、**agent_name** パラメーターのみを指定して **fteCleanAgent** コマンドを実行すると、デフォルトでは、**-all** パラメーターが指定されているかのようにコマンドが実行されます。これにより、エージェント上のスケジュールされた転送、リソース・モニター、およびスケジュールされた転送の定義がすべてクリアされます。

V 9.0.0.2 **V 9.0.3** IBM MQ 9.0.3 および IBM MQ 9.0.0 Fix Pack 2 以降では、エージェント名を指定するだけでなく、適切なパラメーターを **fteCleanAgent** コマンドに渡すことによって、クリアする Managed File Transfer 状態をデフォルトで指定する必要があります。これは、**fteCleanAgent** は、指定されたエージェントの進行中および保留中の転送、リソース・モニター定義、およびスケジュール済み転送定義を、デフォルトではすべて消去しないことを意味します。command.properties ファイルの **failCleanAgentWithNoArguments** プロパティを適切な値に設定することで、この動作を有効または無効にすることができます。

- デフォルトでは、**failCleanAgentWithNoArguments** の値は **true** に設定されているため、**agent_name** パラメーターのみが指定されている場合は、**fteCleanAgent** コマンドの実行は失敗します。

- `failCleanAgentWithNoArguments` が `false` に設定され、`agent_name` パラメーターのみが指定されている場合は、`fteCleanAgent` コマンドは、`-all` パラメーターを指定したときと同様に動作します。

`fteCleanAgent` コマンドは、停止しているエージェントで実行する必要があります。コマンドを現在実行中のエージェントに対して実行しようとする、エラーを受け取ります。このコマンドはエージェントを開始しません。`fteCleanAgent` コマンドによるクリーンアップの対象になるのは、コマンドを実行するシステムにあるエージェントです。リモート・システムにあるエージェントをクリーンアップすることはできません。`fteCleanAgent` コマンドを実行するには、

`MQ_DATA_PATH\mqft\logs\coordination_QMgr_name\agents\agent_name\agent.lck` にあるエージェント・ロック・ファイルへの書き込み権限が必要です。

FTEAGENT グループが `fteCleanAgent` を正常に実行するためには、以下のキューに対して GET 権限および BROWSE 権限を持っている必要があります。

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

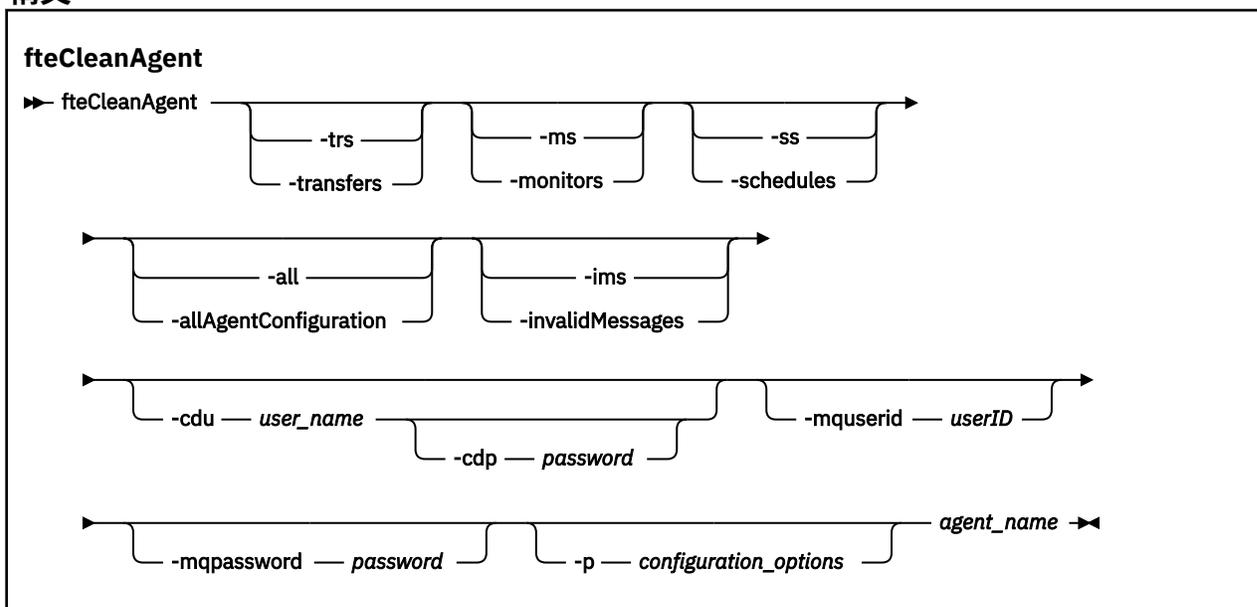
FTEAGENT グループおよびグループ権限の制限について詳しくは、353 ページの『MFT 固有リソースのグループ権限の制限』を参照してください。

キュー・マネージャーにバインディング・モードで接続されているエージェントに対して `fteCleanAgent` コマンドを実行している場合、エージェントが実行を停止すると、`fteCleanAgent` コマンドはメッセージングの問題 MQRC 2042 を報告する場合があります。この MQRC は、エージェントのキュー・ハンドルがまだキュー・マネージャーに存在するために発生します。短い遅延の後キュー・マネージャーはこのハンドルを除去するので、`fteCleanAgent` を再発行できます。

オプションの `-p` パラメーターは、デフォルト・セットと異なる構成オプションのセットを使用する場合にのみこのコマンドに指定します。詳しくは、[構成オプション](#)を参照してください。

注：Connect:Direct ブリッジ・エージェントをクリーンアップする場合は、`fteCleanAgent` コマンドを実行するときに、Connect:Direct ブリッジ・エージェントの一時ディレクトリーに対する読み取り権限と書き込み権限のあるユーザー ID を使用する必要があります。

構文



Parameters

fteCleanAgent コマンドを使用して、特定の成果物を削除できます。例えば、**-trs** コマンドを指定して、保留中の転送は削除するが、リソース・モニターおよびスケジュール済み転送の変更は行わないようにすることができます。

-trs または **-transfers**

オプション。進行中の転送と保留中の転送をエージェントから削除するように指定します。このパラメーターは、**-all** パラメーターまたは **-ims** パラメーターと一緒に指定できません。

-ms または **-monitors**

オプション。すべてのリソース・モニター定義をエージェントから削除するように指定します。このパラメーターは、**-all** パラメーターまたは **-ims** パラメーターと一緒に指定できません。

-ss または **-schedules**

オプション。スケジュール済み転送のすべての定義をエージェントから削除するように指定します。このパラメーターは、**-all** パラメーターまたは **-ims** パラメーターと一緒に指定できません。

-all または **-allAgentConfiguration**

オプション。すべての転送、リソース・モニター定義、およびスケジュール済み転送定義をエージェントから削除するように指定します。このパラメーターは、パラメーター **-trs**、**-ss**、**-ms**、または **-ims** と一緒に指定できません。



重要 : **all** パラメーターは、他のオプションを使用できない場合にのみ使用してください。転送、リソース・モニターの定義、スケジュール済み転送定義を削除するアクションは、企業に大きな影響を及ぼす場合があります。

-ims または **-invalidMessages**

オプション。すべての無効なメッセージをエージェントから削除するように指定します。このパラメーターは、パラメーター **-trs**、**-ss**、**-ms**、または **-all** と一緒に指定できません。

-cdu user_name

オプション。クリーンアップの対象が Connect:Direct ブリッジ・エージェントの場合にのみ有効です。このコマンドを実行するときにこのパラメーターを指定すると、指定したユーザー名に基づいて、Connect:Direct ブリッジ・ノードへの接続が確立され、既存の Connect:Direct プロセスに関する追加情報が取り込まれます。このパラメーターを指定しない場合、エージェントのクリーンアップは実行されますが、Connect:Direct プロセスに関する情報は表示されません。

-cdp password

オプション。クリーンアップの対象が Connect:Direct ブリッジ・エージェントであり、**-cdu** パラメーターが指定されている場合にのみ有効です。**-cdp** パラメーターを指定すると、コマンドは指定されたパスワードを使用して、Connect:Direct ブリッジ・ノードに接続し、既存の Connect:Direct プロセスに関する追加情報を取得します。**-cdu** パラメーターを指定した場合にこのパラメーターを指定しないと、パスワードを対話式に入力することが必要になります。

-mquserid (userID)

オプション。エージェント・キュー・マネージャーで認証するユーザー ID を指定します。

-mqpassword (password)

オプション。エージェント・キュー・マネージャーで認証するパスワードを指定します。**-mquserid** パラメーターも一緒に指定する必要があります。**-mquserid** を指定した場合に **-mqpassword** を指定しないと、関連付けるパスワードを入力するというプロンプトが出されます。パスワードは表示されません。

-p (configuration_options)

オプション。このパラメーターは、エージェントをクリーンアップするために使用される構成オプションのセットを決定します。慣例として、このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティ・ファイルのセットを使用します。

このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。

agent_name

必須。クリーンアップする Managed File Transfer エージェントの名前。

-? または -h

オプション。コマンド構文を表示します。

例

この基本的な例では、AGENT2 が使用しているすべてのキューをクリーンアップします。

```
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.
```

この例では、AGENT2 が使用している無効なメッセージ・キューをクリーンアップします。

```
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.
```

この例では、Connect:Direct ブリッジ・エージェント AGENT_CD_BRIDGE が使用している転送キューをクリーンアップします。

```
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:          414d5120716d312020202020202020202786de4d2048a703
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.

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

関連資料

[531 ページの『fteStopAgent』](#)

fteStopAgent コマンドは、Managed File Transferのエージェントを制御された仕方で停止させるか、必要であれば **-i** パラメーターを使用してエージェントを即時に停止させるために使用します。

[462 ページの『fteDeleteAgent』](#)

fteDeleteAgent コマンドは、Managed File Transfer Agent とその構成を削除します。エージェントがプロトコル・ブリッジ・エージェントの場合は、ユーザー資格情報ファイルがファイル・システムに残ります。

[546 ページの『MFT command.properties ファイル』](#)

`command.properties` ファイルは、コマンド発行時に接続先となるコマンド・キュー・マネージャーと、そのキュー・マネージャーに接続するために Managed File Transfer が必要とする情報を指定します。

fteCreateAgent (MFT エージェントの作成)

fteCreateAgent コマンドは、Managed File Transfer Agent とその関連構成を作成します。

エージェントへのアクセスを制御することができます。詳しくは、[MFT エージェント・アクションのユーザー権限の制限](#)を参照してください。 **-ac** パラメーターを使用し、いくつかのキューにアクセスするための権限を付与する必要があります。

重要: **ULW** IBM MQ for UNIX, Linux, and Windows では、IBM MQ 管理者 (および `mqm` グループのメンバー) であるユーザーのみが、このコマンドを実行できます。IBM MQ 管理者以外のユーザーとしてこのコマンドを実行しようとすると、エラー・メッセージ BFGCL0502E: 要求された操作を実行する権限がありません。を受け取ります。コマンドは実行されません。

z/OS z/OS システムでは、マイグレーション・コマンドを実行するためには、ユーザーは以下の条件の (少なくとも) 1 つを満たす必要があります。

- `mqm` グループのメンバーである (システムに `mqm` グループが定義されている場合)。
- **V9.0.0.1** `BFG_GROUP_NAME` 環境変数で名前が指定されたグループのメンバーである (名前が指定されている場合)。
- **V9.0.0.1** コマンド実行時に `BFG_GROUP_NAME` 環境変数に値が設定されていない。

目的

ftCreateAgent コマンドは、エージェントを作成するために使用します。このコマンドは MQSC コマンドを提供します。これをエージェントのキュー・マネージャーに対して実行し、以下のエージェント・キューを作成する必要があります。

- 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

これらのキューは、内部システム・キューであるため、エージェントを削除しない限り、メッセージの変更、削除、または読み取りを行うことはできません。実行する MQSC コマンドは、`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name\agent_name_create.mqsc`にあるファイルでも提供されます。

エージェントを後で削除する場合は、このコマンドはまた、エージェントが使用するキューをクリアし、続いて削除するために実行する必要がある MQSC コマンドも提供します。MQSC コマンドは、`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name\agent_name_delete.mqsc`にあるファイルに含まれています。

Managed File Transfer は、エージェントの構成に役立つ拡張エージェント・プロパティを備えています。これらのプロパティについては、『[agent.properties ファイル](#)』で説明しています。

エージェントを操作するために、`MQMFTCredentials.xml` 資格情報ファイルを作成しなければならない可能性があります。このファイルのサンプルは `MQ_INSTALLATION_PATH/mqft/samples/credentials/` 内にあります。詳細および例については、『[810 ページの『MFT 資格情報ファイル・フォーマット』](#)』を参照してください。

重要:

UNIX プラットフォームおよび Linux Managed File Transfer コマンドでは、ソケット・ファイルを使用して、同じホスト・マシン上で実行されているエージェント・プロセスと通信します。

これらのソケット・ファイルは、エージェントのログ・ディレクトリーに作成され、エージェントが停止すると削除されます。IBM MQ Managed File Transfer インストール済み環境では、このソケット・ファイルは `<MQ_DATA_PATH>/mqft/logs/<COORDINATION_QM_NAME>/agents/<AGENT_NAME>/logs/<AGENT_NAME>@<AGENT_QM_NAME>` というファイル・パスで作成されます。ここで、`MQ_DATA_PATH` はデフォルトで `/var/mqm` です。

再配布可能エージェントの場合、このソケット・ファイルはディレクトリー `<RE_DISTRIBUTABLE_DIRECTORY>/mqft/logs/<COORDINATION_QM_NAME>/agents/<AGENT_NAME>/logs/<AGENT_NAME>@<AGENT_QM_NAME>` の下に作成されます。

例えば、エージェント名が `SRCAGENT`、エージェント・キュー・マネージャー名が `SRCAGENTQM`、調整キュー・マネージャー名が `COORDQM`、および再配布可能エージェントがディレクトリー `/home/myuser/mqmft-redist` から実行されている場合、このソケット・ファイルの絶対パスは `/home/myuser/mqmft-redist/mqft/logs/COORDQM/agents/SRCAGENT/logs/SRCAGENT@SRCAGENTQM` になります。

これは、85 文字の合計ファイル・パス長です。

これらのオペレーティング・システムで許可されるソケット・ファイルの最大パス長は 107 文字です。したがって、エージェントの作成時には、ソケット・ファイル・パスが 107 文字を超えないように注意してください。これは、エージェントのログ・ディレクトリーを任意のディレクトリー・ロケーションに配置できる再配布可能エージェントでは特に重要です。構成ディレクトリーのセットアップについて詳しくは、**fteCreateEnvironment** コマンドを参照してください。

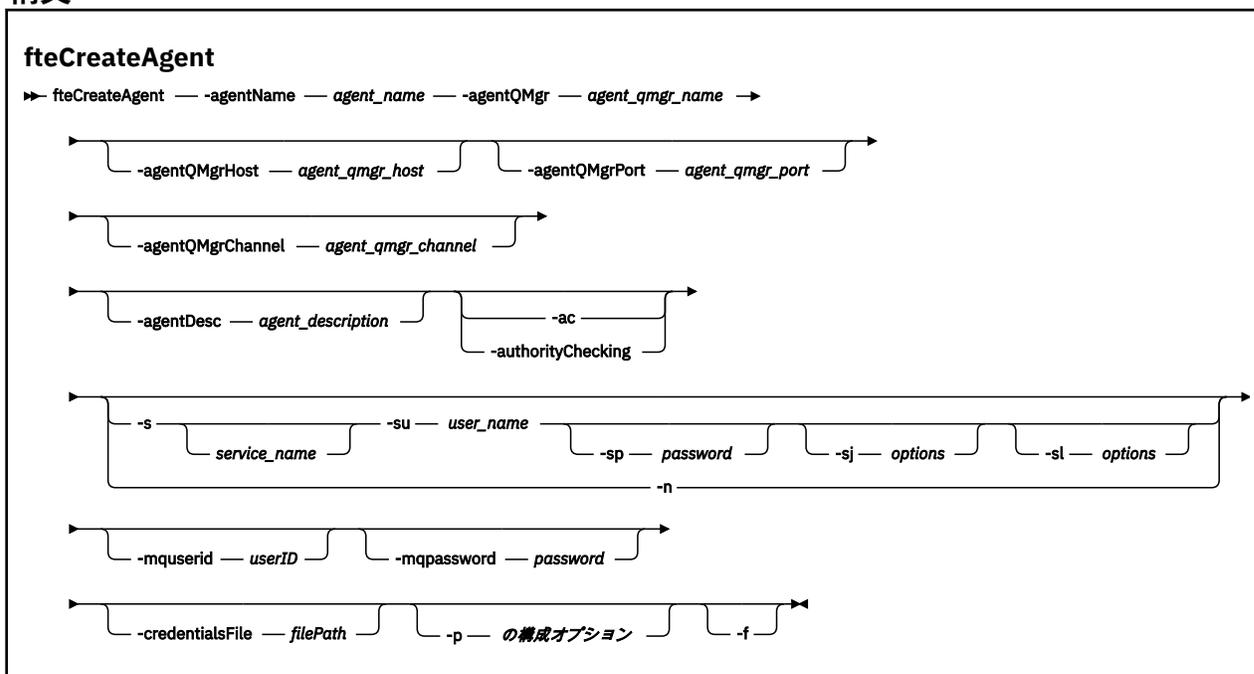
エージェントを開始した場合、またはエージェントに接続する他のコマンドを実行した場合に、パスの長さが 107 文字を超えると、以下のメッセージを受け取ります。

BFGNV0159E: FFDC を使用してソケット・ファイルにバインドしようとして失敗しました。

特殊文字

特殊文字が含まれているパラメーター値を使用する場合には、予期しない形で特殊文字がコマンド・シェルで解釈されないように注意してください。例えば、スペース、引用符 (単一または二重)、スラッシュ、バックスラッシュ文字などを含む完全修飾ファイル・パスや名前は、直接コマンド自体に渡されずに、コマンド・シェルで解釈されてしまう可能性があります。文字がコマンド・シェルによって解釈される動作を回避するには、パラメーター全体を二重/単一引用符で囲むか、またはコマンド・シェルのエスケープ・シーケンスを使用して特殊文字をエスケープします。

構文



Parameters

-agentName (agent_name)

必須。作成するエージェントの名前。エージェント名は、その調整キュー・マネージャーに対して固有である必要があります。

エージェントの命名については、[オブジェクトの命名規則](#)を参照してください。

-agentQMGr (agent_qmgr_name)

必須。エージェントのキュー・マネージャーの名前。

-agentQMGrHost (agent_qmgr_host)

オプション。エージェントのキュー・マネージャーのホスト名または IP アドレス。

-agentQMGrPort (agent_qmgr_port)

オプション。エージェントのキュー・マネージャーとのクライアント接続で使用されるポート番号。

-agentQMGrChannel (agent_qmgr_channel)

オプション。エージェントのキュー・マネージャーとの接続に使用されるチャンネル名。

-agentDesc (agent_description)

オプション。IBM MQ Explorer に表示されるエージェントの説明。

-ac または -authorityChecking

オプション。このパラメーターは、権限検査を有効にします。このパラメーターを指定すると、エージェントは、要求を実行依頼しているユーザーにその要求アクションの実行権限があることを検査します。詳しくは、[MFT エージェント・アクションのユーザー権限の制限](#)を参照してください。

Windows -s (service_name)

オプション (Windows のみ)。エージェントを Windows サービスとして実行するように指示し、Windows 管理者ユーザー ID からコマンドを実行する必要があります。service_name を指定しないと、サービスの名前は mqmftAgentAGENTQMGR になります。AGENT はエージェント名、QMGR はエージェント・キュー・マネージャー名です。

Windows の「サービス」ウィンドウの「名前」列に示されるサービスの表示名は、常に「Managed File Transfer Agent AGENT@QMGR」です。

注：再配布可能エージェントが Windows サービスとして実行される場合、このサービスを機能させるには、システム環境で BFG_DATA 環境変数を設定する必要があります。

Windows -su (user_name)

オプション (Windows のみ)。エージェントが Windows サービスとして実行される場合、このパラメーターは、サービスが実行されるアカウントの名前を指定します。Windows ドメイン・ユーザー・アカウントを使用してエージェントを実行するには、DomainName\UserName の形式で値を指定します。ローカル組み込みドメインからのアカウントを使用してサービスを実行するには、値を UserName の形式で指定します。

-su パラメーターを使用して指定する Windows ユーザー・アカウントには、**Log on as a service** 権限が必要です。この権限を付与する方法については、[MFT エージェントまたはロガーを Windows サービスとして実行するためのガイダンス](#)を参照してください。

-s が指定された場合に必須。

Windows -sp (password)

オプション (Windows のみ)。

このパラメーターは、-s が指定された場合にのみ有効です。-s パラメーターを指定した際にこのパラメーターを指定しなかった場合、警告メッセージが生成されます。このメッセージは、サービスを正常に開始するには、Windows サービス・ツールを使用してパスワードを設定する必要があることを警告します。

Windows -sj (options)

オプション (Windows のみ)。エージェントが Windows サービスとして開始される場合は、JVM に渡されるオプションのリストを -D または -X の形式で定義します。オプションは、番号記号 (#) またはセミコロン (;) 文字を使用して区切られます。# またはセミコロン (;) 文字を組み込む必要がある場合は、それらを単一引用符で囲みます。

このパラメーターは、-s が指定された場合にのみ有効です。

Windows -sl (options)

オプション (Windows のみ)。Windows サービスのログ・レベルを設定します。有効なオプションは、error、info、warn、debug です。デフォルトは info です。このオプションは、Windows サービスに関して問題が発生した場合に便利です。これを debug に設定すると、より詳細な情報がサービス・ログ・ファイルに記録されます。

このパラメーターは、-s が指定された場合にのみ有効です。

Windows -n

オプション (Windows のみ)。エージェントを通常のプロセスとして実行するように指示します。これは **-s** オプションと同時に指定できません。 **-s** パラメーターと **-n** パラメーターのどちらも指定されていない場合、エージェントは通常の Windows プロセスとして構成されます。

-p (configuration_options)

オプション。このパラメーターは、エージェントを作成するために使用される構成オプションのセットを決定します。慣例として、このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。その後、**fteCreateAgent** コマンドは、このデフォルト以外の調整キュー・マネージャーに関連付けられたプロパティ・ファイルのセットを使用します。

オプションの **-p** パラメーターは、デフォルトと異なる構成オプションを使用する場合にのみ指定します。このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。

-mquserid (userID)

オプション。調整キュー・マネージャーで認証するユーザー ID を指定します。

-mqpassword (password)

オプション。調整キュー・マネージャーで認証するパスワードを指定します。 **-mquserid** パラメーターも一緒に指定する必要があります。 **-mquserid** を指定した場合に **-mqpassword** を指定しないと、関連付けるパスワードを入力するというプロンプトが出されます。パスワードは表示されません。

-credentialsFile (filePath)

オプション。IBM MQ 認証の詳細の追加先となる、既存または新規の資格情報ファイルの絶対ファイル・パス。

このコマンドは、指定した Managed File Transfer 資格情報ファイルに IBM MQ 認証の詳細のセットを追加する操作をサポートします。このコマンドは、IBM MQ 接続認証を使用可能にしたときに使用してください。既存の詳細を更新した場合は、強制パラメーター **-f** を使用する必要があります。

-credentialPath (credentials_path)

このコマンドは、資格情報のマイグレーション先の場所を定義します。このパラメーターは、既存の資格情報ファイルのディレクトリー・パスとするか、または新しい資格情報ファイルのディレクトリー・パスとすることができます。  z/OS プラットフォームでは、資格情報ファイルは既存の拡張区分データ・セット (PDSE) とすることができます。PDSE には、資格情報ファイルの既存のメンバーまたは新規メンバーを含めることができます。PDSE の既存のメンバーは、資格情報ファイルを含めるように更新されている必要があります。PDSE の形式は可変ブロックでなければなりません。

-f

オプション。コマンドに強制的に、既存の一致しないパラメーターを上書きさせます。このパラメーターを指定しても、既存の Windows サービス・エージェントが強制的に置換されることはありません。

-?または-h

オプション。コマンド構文を表示します。

例

以下の例では、エージェント・キュー・マネージャー QM_NEPTUNE を使用して AGENT3 が作成され、それはデフォルトの調整キュー・マネージャーを使用します。

```
fteCreateAgent -agentName AGENT3 -agentQMGr QM_NEPTUNE
               -agentQMGrHost myhost.ibm.com -agentQMGrPort 1415 -agentQMGrChannel CHANNEL1
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

fteCreateBridgeAgent (MFT プロトコル・ブリッジ・エージェントの作成および構成)

fteCreateBridgeAgent コマンドは、Managed File Transfer プロトコル・ブリッジ・エージェントとその関連構成を作成します。プロトコル・ブリッジ・エージェントは、ファイルを送受信するファイル・サーバーごとに作成します。

重要: **ULW** IBM MQ for UNIX, Linux, and Windows では、IBM MQ 管理者 (および mqm グループのメンバー) であるユーザーのみが、このコマンドを実行できます。IBM MQ 管理者以外のユーザーとしてこのコマンドを実行しようとする、エラー・メッセージ BFGCL0502E: 要求された操作を実行する権限がありません。を受け取ります。コマンドは実行されません。

z/OS z/OS システムでは、マイグレーション・コマンドを実行するためには、ユーザーは以下の条件の (少なくとも) 1 つを満たす必要があります。

- mqm グループのメンバーである (システムに mqm グループが定義されている場合)。
- **V9.0.0.1** BFG_GROUP_NAME 環境変数で名前が指定されたグループのメンバーである (名前が指定されている場合)。
- **V9.0.0.1** コマンド実行時に BFG_GROUP_NAME 環境変数に値が設定されていない。

目的

fteCreateBridgeAgent コマンドは、プロトコル・ブリッジ・エージェントを作成するために使用します。プロトコル・ブリッジの使用法の概要については、[プロトコル・ブリッジ](#)を参照してください。この **fteCreateBridgeAgent** コマンドは MQSC コマンドを提供します。これをエージェントのキュー・マネージャーに対して実行し、以下のエージェント・キューを作成する必要があります。

- 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

これらのキューは、内部システム・キューであるため、エージェントを削除しない限り、メッセージの変更、削除、または読み取りを行うことはできません。実行する MQSC コマンドは、`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name\agent_name_create.mqsc` にあるファイルでも提供されます。

エージェントを後で削除する場合は、このコマンドはまた、エージェントが使用するキューをクリアし、続いて削除するために実行する必要がある MQSC コマンドも提供します。MQSC コマンドは、`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name\agent_name_delete.mqsc` にあるファイルに含まれています。

fteCreateBridgeAgent コマンドは、ディレクトリー `MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name` に `ProtocolBridgeProperties.xml` XML ファイルを作成します。

ProtocolBridgeCredentials.xml ファイルは、ユーザーが手動で作成する必要があります。このファイルは **fteCreateBridgeAgent** コマンドでは作成されなくなりました。

ProtocolBridgeCredentials.xml ファイルには、プロトコル・ブリッジ・エージェントがプロトコル・サーバーに対する権限を自分に与えるために使用するユーザー名および資格情報を定義でき、ProtocolBridgeProperties.xml ファイルには、複数のプロトコル・ファイル・サーバーを定義できるため、複数のエンドポイントに転送可能になります。

MQ_INSTALLATION_PATH/mqft/samples/credentials/ ディレクトリー内にサンプルの ProtocolBridgeCredentials.xml があります。詳しくは、[574 ページの『プロトコル・ブリッジの資格情報ファイルのフォーマット』](#)および [577 ページの『プロトコル・ブリッジ・プロパティー・ファイルのフォーマット』](#)を参照してください。

fteCreateBridgeAgent コマンドを実行して、デフォルト・プロトコル・ファイル・サーバー (パラメーター -bt) を指定する場合、このデフォルト・サーバーが ProtocolBridgeProperties.xml ファイルに含まれ、そのホスト名がサーバー名として使用されます。-bt パラメーターとともに、以下のパラメーターを指定する必要があります。

- -bh
- -btz
- -bm
- -bsl
- -bfe
- -bts

デフォルト・サーバーを指定しない場合、ProtocolBridgeProperties.xml ファイルには項目がありません。転送が実行される前に、手動で少なくとも 1 つのサーバーを追加する必要があります。

Managed File Transfer は、プロトコル・ブリッジ・エージェントの構成に役立つ拡張エージェント・プロパティーを備えています。プロトコル・ブリッジに関連するプロパティーは、先頭が **protocol** になっています。これらのプロパティーについては、『[agent.properties ファイル](#)』で説明しています。プロトコル・ブリッジで予期しない動作が発生した場合は、それらの **protocol** プロパティーがシステムで正しく設定されていることを確認してください。

fteCreateBridgeAgent コマンドから、次の出力が表示された場合:

```
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.
```

このメッセージは、調整キュー・マネージャーに接続できなかったことを示し、理由として IBM MQ の理由コードを提供しています。この情報メッセージは、調整キュー・マネージャーが現在使用不可であるか、構成が誤って定義されたことを示す可能性があります。

重要:

UNIX プラットフォームおよび Linux Managed File Transfer コマンドでは、ソケット・ファイルを使用して、同じホスト・マシン上で実行されているエージェント・プロセスと通信します。

これらのソケット・ファイルは、エージェントのログ・ディレクトリーに作成され、エージェントが停止すると削除されます。IBM MQ Managed File Transfer インストール済み環境では、このソケット・ファイルは <MQ_DATA_PATH>/mqft/logs/<COORDINATION_QM_NAME>/agents/<AGENT_NAME>/logs/<AGENT_NAME>@<AGENT_QM_NAME> というファイル・パスで作成されます。ここで、MQ_DATA_PATH はデフォルトで /var/mqm です。

再配布可能エージェントの場合、このソケット・ファイルはディレクトリー <RE_DISTRIBUTABLE_DIRECTORY>/mqft/logs/<COORDINATION_QM_NAME>/agents/<AGENT_NAME>/logs/<AGENT_NAME>@<AGENT_QM_NAME>の下に作成されます。

例えば、エージェント名が SRCAGENT、エージェント・キュー・マネージャー名が SRCAGENTQM、調整キュー・マネージャー名が COORDQM、および再配布可能エージェントがディレクトリー /home/myuser/

mqmft-redis から実行されている場合、このソケット・ファイルの絶対パスは /home/myuser/mqmft-redis/mqft/logs/COORDQM/agents/SRCAGENT/logs/SRCAGENT@SRCAGENTQM になります。

これは、85 文字の合計ファイル・パス長です。

これらのオペレーティング・システムで許可されるソケット・ファイルの最大パス長は 107 文字です。したがって、エージェントの作成時には、ソケット・ファイル・パスが 107 文字を超えないように注意してください。これは、エージェントのログ・ディレクトリーを任意のディレクトリー・ロケーションに配置できる再配布可能エージェントでは特に重要です。構成ディレクトリーのセットアップについて詳しくは、[**`fteCreateEnvironment`**](#) コマンドを参照してください。

エージェントを開始した場合、またはエージェントに接続する他のコマンドを実行した場合に、パスの長さが 107 文字を超えると、以下のメッセージを受け取ります。

BFGNV0159E: FFDC を使用してソケット・ファイルにバインドしようとして失敗しました。

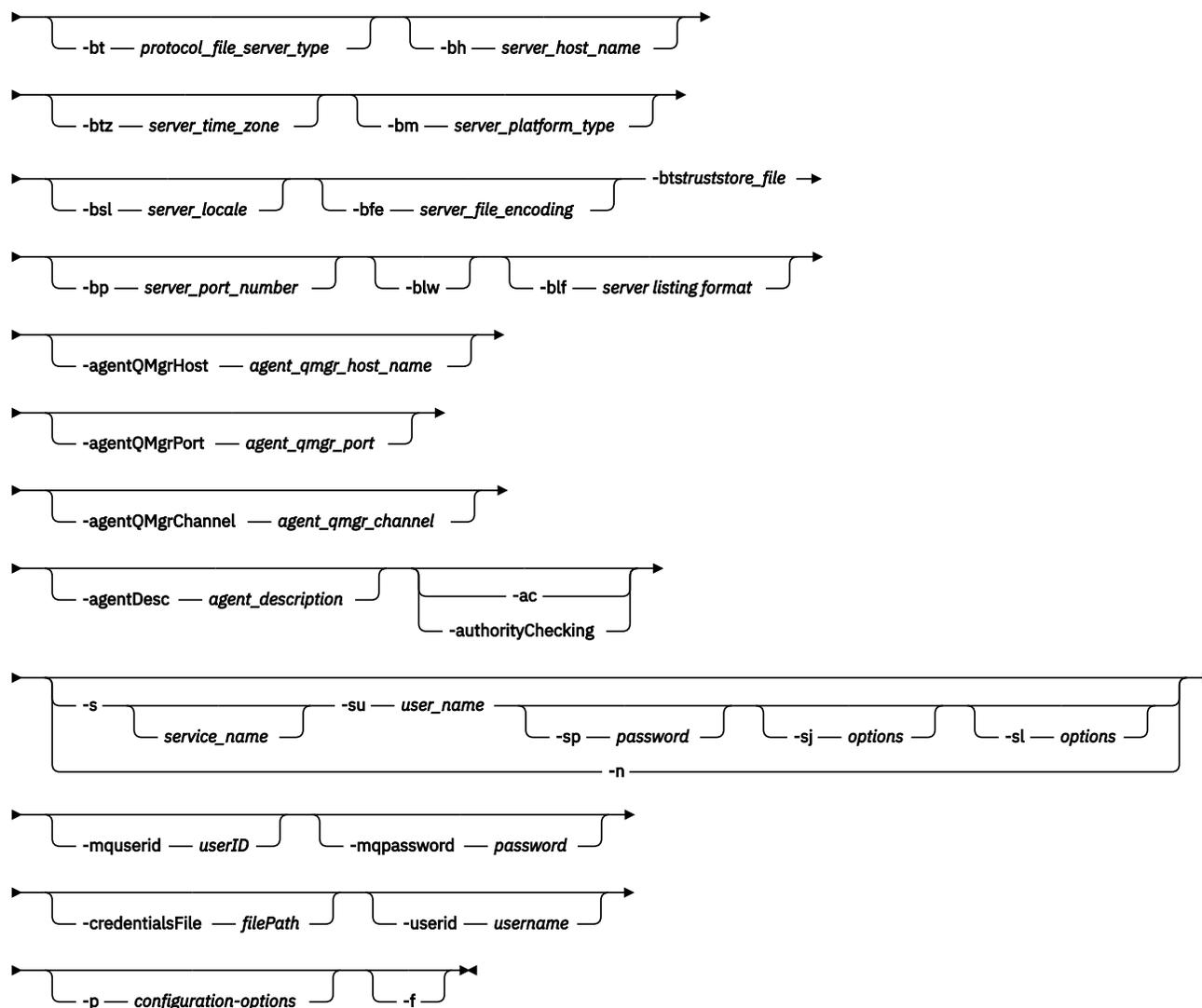
特殊文字

特殊文字が含まれているパラメーター値を使用する場合には、予期しない形で特殊文字がコマンド・シェルで解釈されないように注意してください。例えば、スペース、引用符 (単一または二重)、スラッシュ、バックスラッシュ文字などを含む完全修飾ファイル・パスや名前は、直接コマンド自体に渡されずに、コマンド・シェルで解釈されてしまう可能性があります。文字がコマンド・シェルによって解釈される動作を回避するには、パラメーター全体を二重/単一引用符で囲むか、またはコマンド・シェルのエスケープ・シーケンスを使用して特殊文字をエスケープします。

構文

fteCreateBridgeAgent

▶ fteCreateBridgeAgent — -agentName *agent_name* -agentQMgr *agent_qmgr_name* →



Parameters

-agentName (*agent_name*)

必須。作成するエージェントの名前。エージェント名は管理可能ドメイン内で固有でなければなりません。

エージェントの命名については、[オブジェクトの命名規則](#)を参照してください。

-agentQMgr (*agent_qmgr_name*)

必須。エージェントのキュー・マネージャーの名前。

-bt (*protocol_file_server_type*)

オプション。デフォルトのプロトコル・ファイル・サーバーを定義することを指定します。以下のいずれかのオプションを指定します。

FTP

標準 FTP サーバー

SFTP

SSH FTP サーバー

FTPS

SSL または TLS を使用して保護された FTP サーバー

このパラメーターを指定しなかった場合、デフォルトのプロトコル・サーバーは定義されません。

-bh (server_host_name)

-bt パラメーターを使用してデフォルトのプロトコル・ファイル・サーバーも指定した場合にのみ必須。プロトコル・ファイル・サーバーの IP ホスト名または IP アドレス。

-btz (server_time_zone)

-bt パラメーターも指定した場合にのみ必須 (FTP サーバーおよび FTPS サーバーのみ)。プロトコル・ファイル・サーバーのタイム・ゾーン。タイム・ゾーンは地域/ロケーション形式で指定します。例えば、Europe/London のように指定します。

-htz パラメーターを使用すると、**-btz** に指定できる値をリストできます。例えば、`fteCreateBridgeAgent -htz` です

-bm (server_platform)

-bt パラメーターを使用してデフォルトのプロトコル・ファイル・サーバーも指定した場合にのみ必須。プロトコル・ファイル・サーバーのプラットフォーム・タイプ。以下のいずれかのオプションを指定します。

UNIX

汎用の UNIX プラットフォームと Linux プラットフォーム

WINDOWS

汎用 Windows プラットフォーム

-bsl (server_locale)

-bt パラメーターも指定した場合にのみ必須 (FTP サーバーおよび FTPS サーバーのみ)。プロトコル・ファイル・サーバーのロケール。ロケールは、`xx_XX` 形式で指定します。例えば、`en_GB` のように指定します。

- `xx` は、ISO 言語コードです。有効な値のリストについては、[Codes for the Representation of Names of Languages](#) を参照してください。
- `XX` は、ISO 国別コードです。有効な値のリストについては、[Country names and code elements](#) を参照してください。

-bfe (server_file_encoding)

-bt パラメーターを使用してデフォルトのプロトコル・ファイル・サーバーも指定した場合にのみ必須。プロトコル・ファイル・サーバーに保管されるファイルの文字エンコード方式。例: UTF-8。

-hcs パラメーターを使用すると、**-bfe** に指定できる値をリストできます。例えば、`fteCreateBridgeAgent -hcs` です

-bts (truststore_file)

-bt パラメーターを指定した場合に必須 (FTPS サーバーのみ)。FTPS サーバーによって提示される証明書を妥当性検査するためのトラストストアへのパスを指定します。

-bts パラメーターを指定できるのは、**-bt** パラメーターで FTPS オプションも指定した場合に限られます。

-bp (server_port)

オプション。プロトコル・ファイル・サーバーが接続される IP ポート。このパラメーターは、プロトコル・ファイル・サーバーがそのプロトコル用のデフォルト・ポートを使用しない場合のみ指定します。このパラメーターを指定しない場合、Managed File Transfer は、ファイル・サーバーのプロトコル・タイプ用のデフォルト・ポートを使用します。

-blw

オプション。書き込み機能が制限されるようにプロトコル・ファイル・サーバーを定義します。デフォルトで、プロトコル・ブリッジ・エージェントは、プロトコル・ファイル・サーバーには、ファイルの削除、ファイルの名前変更、および書き込み付加用のファイルのオープンが許可されていることを想定しています。このパラメーターは、プロトコル・ファイル・サーバーにこれらのファイル・アクションが許可されていないことを示すために指定します。代わりに、ファイル・サーバーにはファイルの読み書きだけが許可されます。このパラメーターを指定した場合、転送が中断されたときはそれらの転送はリカバリーできなくなる可能性があり、結果的に実行中のファイル転送が失敗することがあります。

-blf (server listing format)

オプション (FTP サーバーおよび FTPS サーバーの場合のみ)。デフォルトのプロトコル・ファイル・サーバーからリストで返されるファイル情報のサーバー・リスト形式を定義します。オプションは以下のとおりです。

UNIX

汎用の UNIX プラットフォームと Linux プラットフォーム

WINDOWS

汎用 Windows プラットフォーム

選択する形式を識別するには、FTP クライアント・プログラムを使用し、ディレクトリーをリスト表示し、最適な形式を選択します。例:

UNIX UNIX では、次のタイプのリストが表示されます。

```
-rwxr-xr-x 2 userid groupId 4096 2009-07-23 09:36 filename
```

Windows Windows では、次のタイプのリストが表示されます。

```
437,909 filename
```

IBM i IBM i では、次のタイプのリストが表示されます。

```
OS400IFS -rwxrwsrwx 3 USERID 0 8192 Mar 7 08:33 filename
```

デフォルトは UNIX です。これは、ほとんどのサーバーで使用される形式です。

-agentQMgrHost (agent_qmgr_host)

オプション。エージェントのキュー・マネージャーのホスト名または IP アドレス。

-agentQMgrPort (agent_qmgr_port)

オプション。エージェントのキュー・マネージャーとのクライアント接続で使用されるポート番号。

-agentQMgrChannel (agent_qmgr_channel)

オプション。エージェントのキュー・マネージャーとの接続に使用されるチャンネル名。

-agentDesc (agent_description)

オプション。IBM MQ Explorer に表示されるエージェントの説明。

-ac または -authorityChecking

オプション。このパラメーターは、権限検査を有効にします。このパラメーターを指定すると、エージェントは、要求を実行依頼しているユーザーにその要求アクションの実行権限があることを検査します。詳しくは、[MFT エージェント・アクションのユーザー権限の制限](#)を参照してください。

Windows **-s (service_name)**

オプション (Windows のみ)。エージェントを Windows サービスとして実行するように指示します。*service_name* を指定しないと、サービスの名前は `mqmftAgentAGENTQMGR` になります。*AGENT* はエージェント名、*QMGR* はエージェント・キュー・マネージャー名です。

Windows の「サービス」ウィンドウの「名前」列に示されるサービスの表示名は、常に「**Managed File Transfer Agent AGENT@QMGR**」です。

Windows **-su (user_name)**

オプション (Windows のみ)。エージェントが Windows サービスとして実行される場合、このパラメーターは、サービスが実行されるアカウントの名前を指定します。Windows ドメイン・ユーザー・アカウントを使用してエージェントを実行するには、`DomainName\UserName` の形式で値を指定します。ローカル組み込みドメインからのアカウントを使用してサービスを実行するには、値を `UserName` の形式で指定します。

-su パラメーターを使用して指定する Windows ユーザー・アカウントには、**Log on as a service** 権限が必要です。この権限を付与する方法については、[MFT エージェントまたはロガーを Windows サービスとして実行するためのガイダンス](#)を参照してください。

-s が指定された場合に必須。

Windows **-sp (password)**

オプション (Windows のみ)。**-su** パラメーターによって設定されたユーザー・アカウントのパスワード。

このパラメーターは、**-s** が指定された場合にのみ有効です。**-sp** パラメーターを指定した際にこのパラメーターを指定しなかった場合、警告メッセージが生成されます。このメッセージは、サービスを正常に開始するには、Windows サービス・ツールを使用してパスワードを設定する必要があることを警告します。

Windows **-sj (options)**

オプション (Windows のみ)。エージェントが Windows サービスとして開始される場合は、JVM に渡されるオプションのリストを `-D` または `-X` の形式で定義します。オプションは、番号記号 (#) またはセミコロン (;) 文字を使用して区切られます。# またはセミコロン (;) 文字を組み込む必要がある場合は、それらを単一引用符で囲みます。

このパラメーターは、**-s** が指定されている場合にのみ有効です。

Windows **-sl (options)**

オプション (Windows のみ)。Windows サービスのログ・レベルを設定します。有効なオプションは、`error`、`info`、`warn`、`debug` です。デフォルトは `info` です。このオプションは、Windows サービスに関して問題が発生した場合に便利です。これを `debug` に設定すると、より詳細な情報がサービス・ログ・ファイルに記録されます。

このパラメーターは、**-s** が指定された場合にのみ有効です。

Windows **-n**

オプション (Windows のみ)。エージェントを通常のプロセスとして実行するように指示します。これは **-s** オプションと同時に指定できません。**-s** パラメーターと **-n** パラメーターのどちらも指定されていない場合、エージェントは通常の Windows プロセスとして構成されます。

-p (configuration-options)

オプション。このパラメーターは、エージェントを作成するために使用される構成オプションのセットを決定します。慣例として、このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、`fteCreateBridgeAgent` コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティ・ファイルのセットを使用します。

オプションの **-p** パラメーターは、デフォルトと異なる構成オプションを使用する場合にのみ指定します。**-p** を指定しない場合、`installation.properties` ファイルで定義された構成オプションが使用されます。詳しくは、[構成オプション](#)を参照してください。

-f

オプション。コマンドに強制的に既存の構成を上書きさせます。

-htz

オプション。 **-btz** パラメーターの入力として使用可能なサポートされるタイム・ゾーンのリストを表示します。

-hcs

オプション。 **-bfe** パラメーターの入力として使用可能なサポートされる文字セットのリストを表示します。

fteCreateBridgeAgent -hcs コマンドを実行して、JVM にとって既知のコード・ページをリストします。既知のコード・ページは JVM によって異なるので、この情報を外部ソースから利用することはできません。

-mquserid (userID)

オプション。コマンド・キュー・マネージャーで認証するユーザー ID を指定します。

-mqpassword (password)

オプション。コマンド・キュー・マネージャーで認証するパスワードを指定します。 **-mquserid** パラメーターと一緒に指定する必要があります。 **-mquserid** を指定した場合に **-mqpassword** を指定しないと、関連付けるパスワードを入力するというプロンプトが出されます。パスワードは表示されません。

-credentialsFile (filePath)

オプション。IBM MQ 認証の詳細の追加先となる、既存または新規の資格情報ファイルの絶対ファイル・パス。

このコマンドは、指定した Managed File Transfer 資格情報ファイルに IBM MQ 認証の詳細のセットを追加する操作をサポートします。このコマンドは、IBM MQ 接続認証を使用可能にしたときに使用してください。既存の詳細を更新した場合は、強制パラメーター **-f** を使用する必要があります。

-userid (username)

オプション。資格情報の詳細を関連付けるために使用するユーザー ID。ユーザー ID を指定しない場合、資格情報の詳細はすべてのユーザーに適用されます。 **-credentialsFile** パラメーターと一緒に指定する必要があります。

-? または -h

オプション。コマンド構文を表示します。

推奨されないパラメーター

以下のパラメーターは、IBM WebSphere MQ 7.5 または IBM WebSphere MQ File Transfer Edition 7.0.2 以降で非推奨になり、サポートされません。

-brd (reconnect_delay)

非推奨。オプション。プロトコル・ファイル・サーバーとの失われた接続の再確立を次に試行するまでの遅延時間を秒単位で指定します。デフォルト値は 10 秒です。

-brr (reconnect_retries)

非推奨。オプション。デフォルト・プロトコル・ファイル・サーバーとの失われた接続の再確立を試行する再試行の最大回数を指定します。この最大値に達すると、現在のファイル転送は失敗として分類されます。デフォルト値は 2 です。

例

以下の例では、エージェント・キュー・マネージャー QM_ACCOUNTS を使用して新規のプロトコル・ブリッジ・エージェント ACCOUNTS1 が作成され、このエージェントはデフォルトの調整キュー・マネージャーを使用します。ACCOUNTS1 は、FTP サーバー accountshost.ibm.com に接続します。この FTP サーバ

ーは、タイム・ゾーン Europe/Berlin、ロケール de_DE、ファイル・エンコード方式 UTF-8 を使用して、Windows 上で稼働します。再接続の再試行回数は 4 回です。

```
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
```

以下の例では、エージェント・キュー・マネージャー QM_ACCOUNTS を使用して新規のプロトコル・ブリッジ・エージェント ACCOUNTS2 が作成され、このエージェントはデフォルトの調整マネージャーを使用します。ACCOUNTS2 はデフォルトのプロトコル・ファイル・サーバーを使用せずに作成されます。

```
fteCreateBridgeAgent -agentName ACCOUNTS2 -agentQMgr QM_ACCOUNTS
```

注：上記は Managed File Transfer Agent 再配布可能パッケージには当てはまりません。

追加のカスタマイズ

-bt パラメーター (および必要な追加パラメーター) を使用した場合、ProtocolBridgeProperties.xml ファイルにデフォルトのサーバー名が表示されます。

さらに ftp サーバーを追加する、または資格情報ファイルの場所を変更する場合には、ProtocolBridgeProperties.xml ファイルを使用したプロトコル・ファイル・サーバーのプロパティの定義を参照してください。

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

プロトコル・ブリッジ・エージェントを開始するには、**fteStartAgent** コマンドを使用します。詳しくは、528 ページの『fteStartAgent』を参照してください。  z/OS での MFT エージェントの開始も参照してください。

関連資料

574 ページの『プロトコル・ブリッジの資格情報ファイルのフォーマット』

Managed File Transfer Agent の構成ディレクトリーにある ProtocolBridgeCredentials.xml ファイルは、プロトコル・ブリッジ・エージェントがプロトコル・サーバーでの許可に使用するユーザー名および資格情報を定義します。

577 ページの『プロトコル・ブリッジ・プロパティ・ファイルのフォーマット』

エージェント構成ディレクトリー内の ProtocolBridgeProperties.xml ファイルは、プロトコル・ファイル・サーバーのプロパティを定義します。

関連情報

[プロトコル・ブリッジ](#)

fteCreateEnvironment (Redistributable Managed File Transfer Agent の環境のセットアップ)

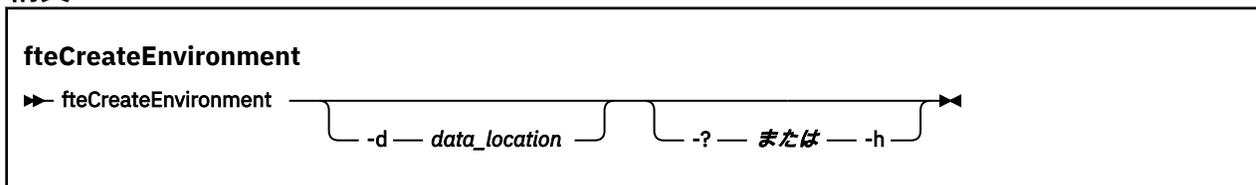
fteCreateEnvironment コマンドは、Redistributable Managed File Transfer Agent のファイルの構成と転送のための環境を設定します。

目的

fteCreateEnvironment コマンドを使用して、Redistributable Managed File Transfer Agent を使用するための環境をセットアップします。-d パラメーターを指定してこのコマンドを実行すると、MFT Agent のデータ・ファイルの場所を指定できます。-d パラメーターを指定しない場合、このコマンドは

Redistributable Managed File Transfer Agent のダウンロード場所にデータ・ファイルを作成し、そのデータ・パスを設定します。

構文



Parameters

-d (*data_location*)

オプション。このパラメーターは、環境のセットアップ時にデータ・ファイルの場所を指定するために使用します。

このパラメーターを指定しない場合は、Redistributable Managed File Transfer Agent が解凍された場所にデータ・ディレクトリーが作成され(まだ存在していない場合)、環境変数 (BFG_DATA) にその場所が設定されます。

-? または -h

オプション。コマンド構文を表示します。

例

この例では、Windows で -d パラメーターを使用してデータ・フォルダーの作成場所を指定しています。

```
fteCreateEnvironment -d C:\mftRedistributable\mftData
```

Linux では、前提条件として、このコマンドを bash シェルで実行する必要があります。bash シェルでは、このコマンドをさまざまな方法で実行できますが、コマンド・ファイルをソースとして使用することが必要です。

```
source Path_of_MFTZipBin/fteCreateEnvironment
```

別の方法として、以下の方法もあります。

```
. Path_of_MFTZipBin/fteCreateEnvironment
```

コマンド・ファイルが存在するディレクトリーから実行する場合は、以下のようになります。

```
./fteCreateEnvironment
```



重要: 最初のピリオド文字 (.) の後にスペースを入れる必要があります。

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

fteCreateCDAgent (Connect:Direct ブリッジ・エージェントの作成)

fteCreateCDAgent コマンドは、Managed File Transfer Agent、および Connect:Direct ブリッジのための関連する構成を作成します。

重要: **ULW** IBM MQ for UNIX, Linux, and Windows では、IBM MQ 管理者 (および mqm グループのメンバー) であるユーザーのみが、このコマンドを実行できます。IBM MQ 管理者以外のユーザーとしてこのコマンドを実行しようとする、エラー・メッセージ BFGCL0502E: 要求された操作を実行する権限がありません。を受け取ります。コマンドは実行されません。

z/OS z/OS システムでは、マイグレーション・コマンドを実行するためには、ユーザーは以下の条件の (少なくとも) 1 つを満たす必要があります。

- mqm グループのメンバーである (システムに mqm グループが定義されている場合)。
- **V9.0.0.1** BFG_GROUP_NAME 環境変数で名前が指定されたグループのメンバーである (名前が指定されている場合)。
- **V9.0.0.1** コマンド実行時に BFG_GROUP_NAME 環境変数に値が設定されていない。

目的

fteCreateCDAgent コマンドを使用して、Connect:Direct ブリッジ・エージェントを作成します。このタイプのエージェントは、Connect:Direct ノードを転送元または転送先とするファイル転送専用です。詳しくは、[Connect:Direct ブリッジ](#)を参照してください。Connect:Direct ブリッジでサポートされているオペレーティング・システム・バージョンの詳細については、Web ページ「[System Requirements for IBM MQ](#)」を参照してください。

このコマンドは MQSC コマンドを提供します。これをエージェントのキュー・マネージャーに対して実行し、以下のエージェント・キューを作成する必要があります。

- 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

これらのキューは、内部システム・キューであるため、エージェントを削除しない限り、メッセージの変更、削除、または読み取りを行うことはできません。実行する MQSC コマンドは、`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name\agent_name_create.mqsc`にあるファイルでも提供されます。

エージェントを後で削除する場合は、このコマンドはまた、エージェントに属するキューをクリアし、続いて削除するために実行する必要がある MQSC コマンドも提供します。MQSC コマンドは、`MQ_DATA_PATH\mqft\config\coordination_qmgr_name\agents\agent_name\agent_name_delete.mqsc`にあるファイルに含まれています。

Managed File Transfer は、エージェントの構成に役立つ拡張エージェント・プロパティを備えています。これらのプロパティについては、『[MFT agent.properties ファイル](#)』を参照してください。

fteCreateCDAgent コマンドは、エージェントのプロパティ・ディレクトリーに 2 つの XML ファイルを作成します。ConnectDirectNodeProperties.xml、これは、転送内のリモート・ノードに関する情報を定義するために使用され、ConnectDirectProcessDefinitions.xml は、転送によって開始されるユーザー定義の Connect:Direct プロセスを指定するために使用されます。

Connect:Direct ブリッジ・エージェントが Connect:Direct ノードへの接続に使用するユーザー名とパスワードを定義するには、手動で ConnectDirectCredentials.xml ファイルを作成する必要があります。

XML ファイルのサンプルは `MQ_INSTALLATION_PATH/mqft/samples/credentials/` 内にあります。詳細および例については、『584 ページの『Connect:Direct 資格情報ファイル・フォーマット』』を参照してください。

重要:

UNIX プラットフォームおよび Linux Managed File Transfer コマンドでは、ソケット・ファイルを使用して、同じホスト・マシン上で実行されているエージェント・プロセスと通信します。

これらのソケット・ファイルは、エージェントのログ・ディレクトリーに作成され、エージェントが停止すると削除されます。IBM MQ Managed File Transfer インストール済み環境では、このソケット・ファイルは `<MQ_DATA_PATH>/mqft/logs/<COORDINATION_QM_NAME>/agents/<AGENT_NAME>/logs/<AGENT_NAME>@<AGENT_QM_NAME>` というファイル・パスで作成されます。ここで、`MQ_DATA_PATH` はデフォルトで `/var/mqm` です。

再配布可能エージェントの場合、このソケット・ファイルはディレクトリー `<RE_DISTRIBUTABLE_DIRECTORY>/mqft/logs/<COORDINATION_QM_NAME>/agents/<AGENT_NAME>/logs/<AGENT_NAME>@<AGENT_QM_NAME>` の下に作成されます。

例えば、エージェント名が `SRCAGENT`、エージェント・キュー・マネージャー名が `SRCAGENTQM`、調整キュー・マネージャー名が `COORDQM`、および再配布可能エージェントがディレクトリー `/home/myuser/mqmft-redis` から実行されている場合、このソケット・ファイルの絶対パスは `/home/myuser/mqmft-redis/mqft/logs/COORDQM/agents/SRCAGENT/logs/SRCAGENT@SRCAGENTQM` になります。

これは、85 文字の合計ファイル・パス長です。

これらのオペレーティング・システムで許可されるソケット・ファイルの最大パス長は 107 文字です。したがって、エージェントの作成時には、ソケット・ファイル・パスが 107 文字を超えないように注意してください。これは、エージェントのログ・ディレクトリーを任意のディレクトリー・ロケーションに配置できる再配布可能エージェントでは特に重要です。構成ディレクトリーのセットアップについて詳しくは、**`fteCreateEnvironment`** コマンドを参照してください。

エージェントを開始した場合、またはエージェントに接続する他のコマンドを実行した場合に、パスの長さが 107 文字を超えると、以下のメッセージを受け取ります。

```
BFGNV0159E: FFDC を使用してソケット・ファイルにバインドしようとして失敗しました。
```

特殊文字

特殊文字が含まれているパラメーター値を使用する場合には、予期しない形で特殊文字がコマンド・シェルで解釈されないように注意してください。例えば、スペース、引用符 (単一または二重)、スラッシュ、バックスラッシュ文字などを含む完全修飾ファイル・パスや名前は、直接コマンド自体に渡されずに、コマンド・シェルで解釈されてしまう可能性があります。文字がコマンド・シェルによって解釈される動作を回避するには、パラメーター全体を二重/単一引用符で囲むか、またはコマンド・シェルのエスケープ・シーケンスを使用して特殊文字をエスケープします。

fteCreateCDAgent

► fteCreateCDAgent — *agent_name* — *agent_qmgr_name* →

► -cdNode — *cd_node_name* — *agent_qmgr_host* →

► -agentQMgrPort — *agent_qmgr_port* →

► -agentQMgrChannel — *agent_qmgr_channel* →

► -agentDesc — *agent_description* — *ac* — *authorityChecking* →

► -p — *configuration_options* — *f* — *cdNodeHost* — *cd_node_host* →

► -cdNodePort — *cd_node_port* — *cdTmpDir* — *cd_tmp_dir* →

► -s — *service_name* — *su* — *user_name* — *sp* — *password* — *sj* — *options* — *sl* — *options* — *n* →

► -mquserid — *userID* — *mqpassword* — *password* →

► -credentialsFile — *filePath* — *userid* — *username* →

Parameters

-agentName (*agent_name*)

必須。作成するエージェントの名前。エージェント名は、その調整キュー・マネージャーに対して固有である必要があります。

エージェントの命名については、[オブジェクトの命名規則](#)を参照してください。

-agentQMgr (*agent_qmgr_name*)

必須。エージェントのキュー・マネージャーの名前。

-cdNode *cd_node_name*

必須。このエージェントから宛先の Connect:Direct ノードにメッセージを転送するために使用する Connect:Direct ノードの名前。このパラメーターの値はロギングのために使用されます。

Connect:Direct ブリッジ・エージェントに、接続先ノードを指定するためには使用されません。

-cdNodeHost および **-cdNodePort** の値は、Connect:Direct ブリッジの一部である Connect:Direct ノードを指定します。

-agentQMgrHost (*agent_qmgr_host*)

オプション。エージェントのキュー・マネージャーのホスト名または IP アドレス。

-agentQMgrPort (*agent_qmgr_port*)

オプション。エージェントのキュー・マネージャーとのクライアント接続で使用されるポート番号。

-agentQMgrChannel (*agent_qmgr_channel*)

オプション。エージェントのキュー・マネージャーとの接続に使用されるチャンネル名。

-agentDesc (*agent_description*)

オプション。IBM MQ Explorer に表示されるエージェントの説明。

-ac または **-authorityChecking**

オプション。このパラメーターは、権限検査を有効にします。このパラメーターを指定すると、エージェントは、要求を実行依頼しているユーザーにその要求アクションの実行権限があることを検査します。詳しくは、[MFT エージェント・アクションのユーザー権限の制限](#)を参照してください。

-p (configuration_options)

オプション。このパラメーターは、エージェントを作成するために使用される構成オプションのセットを決定します。慣例として、このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、**fteCreateCDAgent** コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティ・ファイルのセットを使用します。

オプションの **-p** パラメーターは、デフォルトと異なる構成オプションを使用する場合にのみ指定します。このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。

-f

オプション。コマンドに強制的に、既存の一致しないパラメーターを上書きさせます。このパラメーターを指定しても、既存の Windows サービス・エージェントが強制的に置換されることはありません。

-cdNodeHost cd_node_host_name

オプション。 **-cdNode** パラメーターで指定された、Connect:Direct ノードが配置されているシステムのホスト名または IP アドレス。 **-cdNodeHost** パラメーターを指定しないと、デフォルトでローカル・システムのホスト名または IP アドレスが使用されます。

ほとんどの場合、Connect:Direct ノードは、Connect:Direct ブリッジ・エージェントと同じシステム上にあります。その場合、このプロパティのデフォルト値 (ローカル・システムの IP アドレス) は適正です。システムに複数の IP アドレスがある場合、または Connect:Direct ノードが Connect:Direct ブリッジ・エージェントと異なるシステム上にあり、両者のシステムが同一のファイル・システムを共有する場合は、このプロパティを使用して Connect:Direct ノード用の適切なホスト名を指定します。

-cdNodePort cd_node_port_name

オプション。 **-cdNode** パラメーターで指定されているノードと通信するためにクライアント・アプリケーションが使用する Connect:Direct ノードのポート番号。Connect:Direct の製品資料で、このポートは API ポートと呼ばれています。 **-cdNodePort** パラメーターを指定しない場合、デフォルトのポート番号 1363 が使用されます。

-cdTmpDir cd_tmp_directory

オプション。このエージェントで、宛先 Connect:Direct ノードへ転送する前に、一時的にファイルを保管するために使用するディレクトリー。このパラメーターは、ファイルが一時的に保管されるディレクトリーの絶対パスを指定します。例えば、**cdTmpDir** を /tmp に設定すると、ファイルは /tmp ディレクトリーに一時的に置かれます。 **-cdTmpDir** パラメーターを指定しない場合は、**cdbridge-agent_name** という名前のディレクトリーにファイルが一時的に保管されます。このデフォルト・ディレクトリーは、**java.io.tmpdir** プロパティの値で定義される場所に作成されます。

注 : **fteCleanAgent** コマンドを実行すると、このディレクトリーに含まれているすべてのファイルが削除されます。

Windows -s (service_name)

オプション (Windows のみ)。エージェントを Windows サービスとして実行するように指示し、Windows 管理者ユーザー ID からコマンドを実行する必要があります。 **service_name** を指定しないと、サービスの名前は **mqmftAgentAGENTQMGR** になります。 **AGENT** はエージェント名、 **QMGR** はエージェント・キュー・マネージャー名です。

Windows の「サービス」ウィンドウの「名前」列に示されるサービスの表示名は、常に「**Managed File Transfer Agent AGENT@QMGR**」です。

注 : 再配布可能エージェントが Windows サービスとして実行される場合、このサービスを機能させるには、システム環境で **BFG_DATA** 環境変数を設定する必要があります。

Windows **-su (user_name)**

オプション (Windows のみ)。エージェントが Windows サービスとして実行される場合、このパラメーターは、サービスが実行されるアカウントの名前を指定します。Windows ドメイン・ユーザー・アカウントを使用してエージェントを実行するには、DomainName\UserName の形式で値を指定します。ローカル組み込みドメインからのアカウントを使用してサービスを実行するには、値を UserName の形式で指定します。

-su パラメーターを使用して指定する Windows ユーザー・アカウントには、**Log on as a service** 権限が必要です。この権限を付与する方法については、[MFT エージェントまたはロガーを Windows サービスとして実行するためのガイダンス](#)を参照してください。

-s が指定された場合に必須。

Windows **-sp (password)**

オプション (Windows のみ)。

このパラメーターは、**-s** が指定された場合にのみ有効です。**-s** パラメーターを指定した際にこのパラメーターを指定しなかった場合、警告メッセージが生成されます。このメッセージは、サービスを正常に開始するには、Windows サービス・ツールを使用してパスワードを設定する必要があることを警告します。

Windows **-sj (options)**

オプション (Windows のみ)。エージェントが Windows サービスとして開始される場合は、JVM に渡されるオプションのリストを **-D** または **-X** の形式で定義します。オプションは、番号記号 (#) またはセミコロン (;) 文字を使用して区切られます。# またはセミコロン (;) 文字を組み込む必要がある場合は、それらを単一引用符で囲みます。

このパラメーターは、**-s** が指定された場合にのみ有効です。

Windows **-sl (options)**

オプション (Windows のみ)。Windows サービスのログ・レベルを設定します。有効なオプションは、error、info、warn、debug です。デフォルトは info です。このオプションは、Windows サービスに関して問題が発生した場合に便利です。これを debug に設定すると、より詳細な情報がサービス・ログ・ファイルに記録されます。

このパラメーターは、**-s** が指定された場合にのみ有効です。

Windows **-n**

オプション (Windows のみ)。エージェントを通常のプロセスとして実行するように指示します。これは **-s** オプションと同時に指定できません。**-s** パラメーターと **-n** パラメーターのどちらも指定されていない場合、エージェントは通常のプロセスとして構成されます。

-mquserid (userID)

オプション。コマンド・キュー・マネージャーで認証するユーザー ID を指定します。

-mqpassword (password)

オプション。コマンド・キュー・マネージャーで認証するパスワードを指定します。**-mquserid** パラメーターも一緒に指定する必要があります。**-mquserid** を指定した場合に **-mqpassword** を指定しないと、関連付けるパスワードを入力するというプロンプトが出されます。パスワードは表示されません。

-credentialsFile (filePath)

オプション。IBM MQ 認証の詳細の追加先となる、既存または新規の資格情報ファイルの絶対ファイル・パス。

このコマンドは、指定した Managed File Transfer 資格情報ファイルに IBM MQ 認証の詳細のセットを追加する操作をサポートします。このコマンドは、IBM MQ 接続認証を使用可能にしたときに使用してください。既存の詳細を更新した場合は、強制パラメーター **-f** を使用する必要があります。

-userid (username)

オプション。資格情報の詳細を関連付けるために使用するユーザー ID。ユーザー ID を指定しない場合、資格情報の詳細はすべてのユーザーに適用されます。 **-credentialsFile** パラメーターも一緒に指定する必要があります。

例

以下の例では、エージェント・キュー・マネージャー QM_NEPTUNE を使用して新規の Connect:Direct ブリッジ・エージェント CD_BRIDGE が作成されます。このエージェントは Connect:Direct ノード BRIDGE_NODE を使用して、他の Connect:Direct ノードにファイルを転送します。BRIDGE_NODE ノードはエージェントと同じシステムにあり、デフォルトのポートをクライアント接続で使用します。Connect:Direct を転送元または転送先として転送されるファイルは、/tmp/cd-bridge ディレクトリーに一時的に保管されます。

```
fteCreateCDAgent -agentName CD_BRIDGE -agentQMgr QM_NEPTUNE  
                -cdNode BRIDGE_NODE -cdTmpDir /tmp/cd-bridge
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

fteCreateLogger (MFT ファイルまたはデータベース・ロガーの作成)

fteCreateLogger コマンドを使用して、Managed File Transfer ファイルまたはデータベース・ロガーを作成します。

重要: **ULW** IBM MQ for UNIX, Linux, and Windows では、IBM MQ 管理者 (および mqm グループのメンバー) であるユーザーのみが、このコマンドを実行できます。IBM MQ 管理者以外のユーザーとしてこのコマンドを実行しようとする、エラー・メッセージ BFGCL0502E: 要求された操作を実行する権限がありません。を受け取ります。コマンドは実行されません。

z/OS z/OS システムでは、マイグレーション・コマンドを実行するためには、ユーザーは以下の条件の (少なくとも) 1 つを満たす必要があります。

- mqm グループのメンバーである (システムに mqm グループが定義されている場合)。
- **V9.0.0.1** BFG_GROUP_NAME 環境変数で名前が指定されたグループのメンバーである (名前が指定されている場合)。
- **V9.0.0.1** コマンド実行時に BFG_GROUP_NAME 環境変数に値が設定されていない。

IBM i のロガー

IBM i

Managed File Transfer ロガーは IBM i プラットフォームではサポートされません。

目的

fteCreateLogger コマンドは、以下のロガー・キューを作成するためにロガー・コマンド・キュー・マネージャーに対して実行する必要がある MQSC コマンドを提供します。

- SYSTEM.FTE.LOG.CMD.*logger_name*
- SYSTEM.FTE.LOG.RJCT.*logger_name*

これらのキューは、内部システム・キューであるため、ロガーを削除しない限り、メッセージの変更、削除、または読み取りを行うことはできません。実行する MQSC コマンドは、

MQ_DATA_PATH\mqft\config\coordination_qmgr\loggers\logger_name\logger_name_create.mqscにあるファイルでも提供されます。

後でロガーを削除する場合は、**fteDeleteLogger** コマンドを使用します。

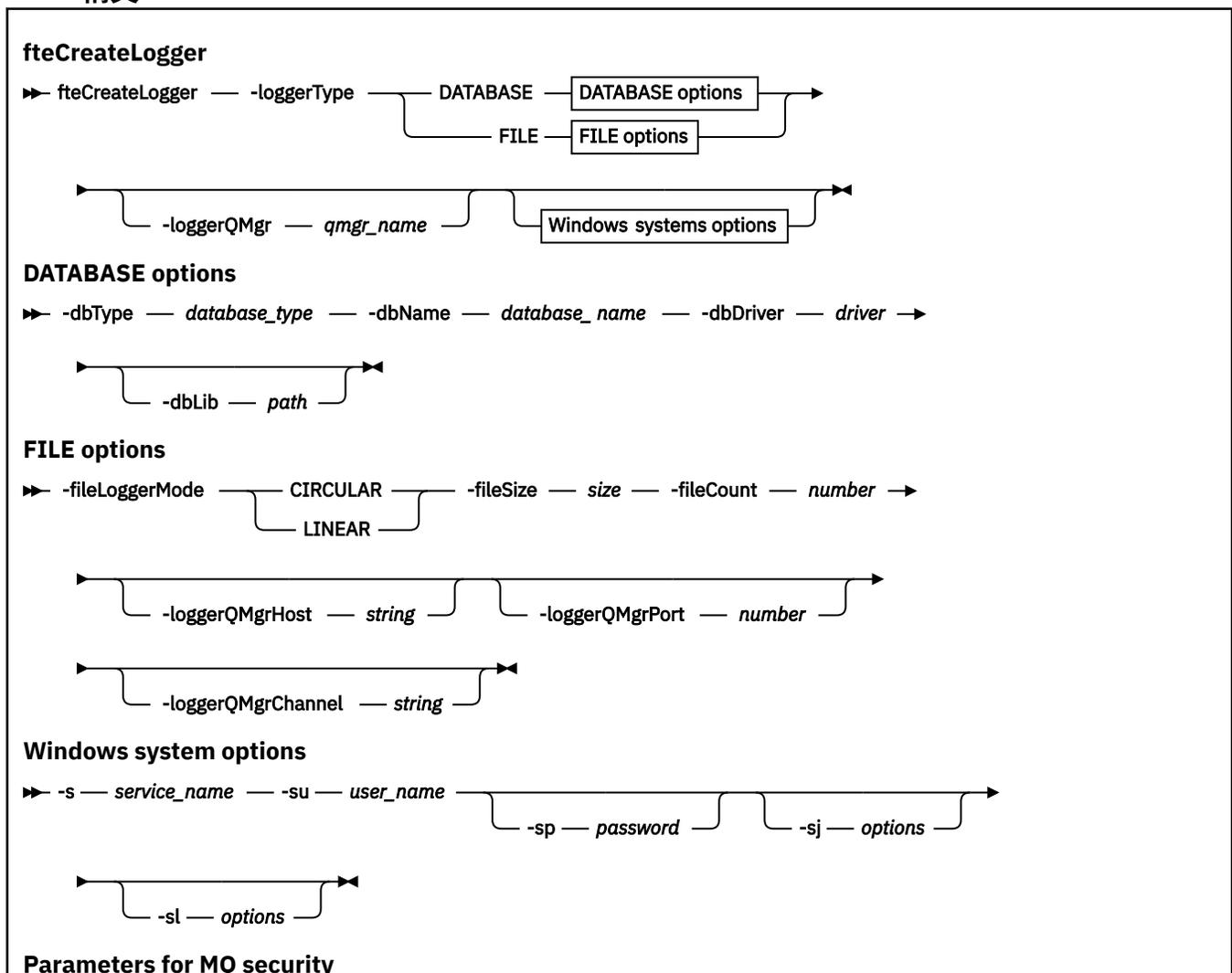
Managed File Transfer は、ロガーの構成に役立つ拡張ロガー・プロパティを備えています。 MFT ロガーの構成プロパティを参照してください。

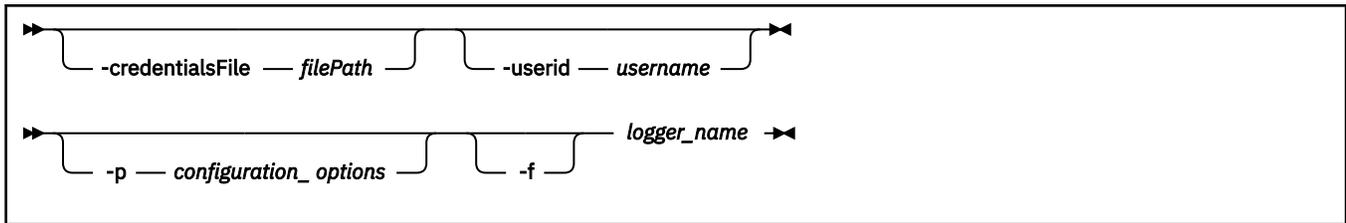
注：作成しているロガーがデータベース・ロガーであり、ローカル Db2 データベースに接続されない場合は、手動で MQMFTCredentials.xml ファイルを作成する必要があります。データベースに接続するためのユーザー名とパスワードを格納するファイル。 logger.properties ファイルの wmqfte.database.credentials プロパティ・ファイルを使用して、MQMFTCredentials.xml ファイルのパスを指定する必要があります。この資格情報ファイルのサンプルは MQ_INSTALLATION_PATH/mqft/samples/credentials/ 内にあります。

特殊文字

特殊文字が含まれているパラメーター値を使用する場合には、予期しない形で特殊文字がコマンド・シェルで解釈されないように注意してください。例えば、スペース、引用符(単一または二重)、スラッシュ、バックスラッシュ文字などを含む完全修飾ファイル・パスや名前は、直接コマンド自体に渡されずに、コマンド・シェルで解釈されてしまう可能性があります。文字がコマンド・シェルによって解釈される動作を回避するには、パラメーター全体を二重/単一引用符で囲むか、またはコマンド・シェルのエスケープ・シーケンスを使用して特殊文字をエスケープします。

構文





Parameters

-loggerType (type)

必須。どこにファイル転送管理情報が記録されるかを指定します。オプションのタイプは、転送情報をデータベースに記録する場合は DATABASE、情報をファイルに記録する場合は FILE のいずれかです。

-loggerQMgr (qmgr_name)

オプション。ファイル転送管理の情報が含まれるメッセージを受け取るために接続するキュー・マネージャーを指定します。キュー・マネージャーはロガーと同じシステム上になければなりません。

-loggerQMgr パラメーターを指定しない場合、このロガー用に設定されている構成オプションで関連付けられている調整キュー・マネージャーがデフォルトで使用されます。

-dbType (database_type)

-loggerType が DATABASE の場合に必須。ファイル転送管理情報の格納に使用するデータベース管理システムのタイプを指定します。オプションは db2 または oracle です

注: SQL ファイルを使用して表を作成する必要があります。 .sql ファイルは MQ_INSTALLATION_PATH_/mqft/sql から入手できます。

- Db2 データベースの場合: ftelog_tables_db2.sql
- Oracle データベースの場合: ftelog_tables_oracle.sql

-dbName (database_name)

-loggerType が DATABASE の場合に必須。ファイル転送管理情報が格納されるデータベースの名前。データベースは Managed File Transfer ログ表で構成する必要があります。

-dbDriver (driver)

-loggerType が DATABASE の場合に必須。データベースの JDBC ドライバー・クラスの場所。これは通常、JAR ファイルのパスおよびファイル名です。

-dbLib (path)

-loggerType が DATABASE の場合にオプション。選択したデータベース・ドライバーが必要とするネイティブ・ライブラリーの場所。

-fileLoggerMode (mode)

-loggerType が FILE の場合に必須。ファイル転送管理情報の格納に使用するファイル・システムのタイプを指定します。オプションは LINEAR または CIRCULAR です。

オプション LINEAR は、ファイルが -filesize で定義されている最大サイズに到達するまで、ファイル・ロガーがそのファイルに情報を書き込むことを意味します。最大サイズに達すると、ファイル・ロガーは新しいファイルに移ります。以前に書き込まれたファイルは削除されないため、それらをログ・メッセージの履歴レコードとして保持できます。このモードで実行している場合、ファイルは削除されず、作成可能なファイル数の上限もないため、-fileCount は無視されます。このモードで実行している場合には上限がないため、ディスク・スペースが少なくならないように、ログ・ファイルによって使用されているディスク・スペースの量を追跡する必要があります。

オプション CIRCULAR は、ファイルが -fileSize で定義されている最大サイズに到達するまで、ファイル・ロガーがそのファイルに情報を書き込むことを意味します。最大サイズに達すると、ファイル・ロガーは新しいファイルに移ります。このモードで書き込まれるファイルの最大数は、-fileCount を使

用して定義される値によって制御されます。このファイル最大数に到達すると、ファイル・ロガーは最初のファイルを削除して再作成し、それを現行のアクティブ・ファイルとして使用します。-fileSizeで定義された値が固定サイズ・バイト単位である場合、このモードで使用されるディスク・スペースの上限は、fileSize x fileCount に等しくなります。-fileSizeで定義された値が時間単位である場合、最大サイズは、ご使用のシステムにおけるその時間内のログ・メッセージのスループットによって決まります。

詳しくは、[MFT ロガーの構成プロパティ](#)を参照してください。

-fileSize (size)

-loggerType が FILE の場合に必須。ログ・ファイルが大きくなることが許可される最大サイズ。値は、ゼロより大きい正整数で、単位 KB、MB、GB、m (分)、h (時)、d (日)、w (週) のいずれか 1 つを後に付けます。例えば、-fileSize 5MB (最大サイズを 5MB に指定)、-fileSize 2d (最大 2 日分のデータを指定) です。

-fileCount (number)

-loggerType が FILE であり、-fileLoggerMode が CIRCULAR である場合に必須。作成するログ・ファイルの最大数。データ量が、このファイル数で保管できる最大量を超えた場合、最も古いファイルが削除されるので、ログ・ファイルの数はこのパラメーターで指定される値を超えることはありません。

-loggerQMGrHost

ロガー・キュー・マネージャーが実行されているマシンのホスト名または IP アドレス。

デフォルト値は none です。

-loggerQMGrHost パラメーターを指定しなかった場合、ロガーはバインディング・モードで作成されます。

-loggerQMGrPort

ロガー・キュー・マネージャーが listen するポート番号。

デフォルト値は 1414 です。

-loggerQMGrChannel

ロガー・キュー・マネージャーへの接続に使用されるチャンネルの名前。

デフォルト値は SYSTEM.DEF.SVRCONN です。

Windows -s (service_name)

オプション (Windows システムのみ)。ロガーを Windows サービスとして実行するように指示します。service_name を指定しない場合、サービスの名前は、mqmftLoggerLOGGERQMGR となります。ここで、LOGGER はロガーの名前であり、QMGR はロガー・キュー・マネージャーの名前です。

Windows の「サービス」ウィンドウの「名前」列に示されるサービスの表示名は、常に「**Managed File Transfer Logger LOGGER@QMGR**」です。

Windows -su (user_name)

オプション (Windows のみ)。ロガーが Windows サービスとして実行される場合、このパラメーターは、サービスが実行されるアカウントの名前を指定します。Windows ドメイン・ユーザー・アカウントを使用してロガーを実行するには、DomainName\UserName の形式で値を指定します。ローカル組み込みドメインからのアカウントを使用してサービスを実行するには、値を UserName の形式で指定します。

-su パラメーターを使用して指定する Windows ユーザー・アカウントには、**Log on as a service** 権限が必要です。この権限を付与する方法については、[MFT エージェントまたはロガーを Windows サービスとして実行するためのガイダンス](#)を参照してください。

-s が指定された場合に必須。

Windows **-sp (password)**

オプション (Windows のみ)。 **-su** パラメーターによって設定されたユーザー・アカウントのパスワード。

このパラメーターは、**-s** が指定された場合にのみ有効です。 **-s** パラメーターを指定した際にこのパラメーターを指定しなかった場合、警告メッセージが生成されます。 このメッセージは、サービスを正常に開始するには、Windows サービス・ツールを使用してパスワードを設定する必要があることを警告します。

Windows **-sj (options)**

オプション (Windows のみ)。 ロガーが Windows サービスとして開始される場合は、JVM に渡されるオプションのリストを **-D** または **-X** の形式で定義します。 オプションは、番号記号 (#) またはセミコロン (;) 文字を使用して区切られます。 (#) またはセミコロン (;) 文字を組み込む必要がある場合は、それらを単一引用符で囲みます。

このパラメーターは、**-s** が指定された場合にのみ有効です。

Windows **-sl (options)**

オプション (Windows のみ)。 Windows サービスのログ・レベルを設定します。 有効なオプションは、error、info、warn、debug です。 デフォルトは info です。 このオプションは、Windows サービスに関して問題が発生した場合に便利です。 これを debug に設定すると、より詳細な情報がサービス・ログ・ファイルに記録されます。

このパラメーターは、**-s** が指定された場合にのみ有効です。

-p (configuration options)

オプション。 ロガーの作成に使用する構成オプションのセットを指定します。 慣例として、この値は調整キュー・マネージャーの名前です。 このパラメーターを指定しない場合、デフォルトの構成オプションのセットが使用されます。

-f

オプション。 コマンドに強制的に既存の構成を上書きさせます。

(logger_name)

必須。 作成するロガーの名前。 これは Managed File Transfer キュー名に取り込まれるので、文字、数字、およびピリオド (.) と下線文字 (_) のみが含まれるようにする必要があります。 また、長さが最大 28 文字までに制限されています。

-credentialsFile (filePath)

オプション。 IBM MQ 認証の詳細の追加先となる、既存または新規の資格情報ファイルの絶対ファイル・パス。

このコマンドは、指定した Managed File Transfer 資格情報ファイルに IBM MQ 認証の詳細のセットを追加する操作をサポートします。 このコマンドは、IBM MQ 接続認証を使用可能にしたときに使用してください。 既存の詳細を更新した場合は、強制パラメーター **-f** を使用する必要があります。

-userid (username)

オプション。 資格情報の詳細を関連付けるために使用するユーザー ID。 ユーザー ID を指定しない場合、資格情報の詳細はすべてのユーザーに適用されます。 **-credentialsFile** パラメーターも一緒に指定する必要があります。

-? または -h

オプション。 コマンド構文を表示します。

例

この例では、filelogger1 という循環ファイル・ロガーが作成されます。 ファイル・ロガーでは、最大 10 ファイルが作成され、各ファイルのサイズは 10MB、総ディスク・スペースは最大 100MB です。

```
fteCreateLogger -loggerType FILE -fileLoggerMode CIRCULAR -fileSize 10MB -fileCount 10 filelogger1
```

この例では、dblogger1 というデータベース・ロガーが作成されます。データベース・ロガーは、FTEDB という Db2 データベースに接続します。

```
fteCreateLogger -loggerType DATABASE -dbName FTEDB -dbType DB2
                -dbDriver "C:\Program Files (x86)\IBM\SQLLIB\java\db2jcc4.jar" dblogger1
```

この例では、dblogger1 というデータベース・ロガーが作成されます。データベース・ロガーは、FTEDB という Oracle データベースに接続します。

```
fteCreateLogger -loggerType DATABASE -dbName FTEDB -dbType oracle
                -dbDriver "C:\app\oracle\product\12.1.0\dbhome_2\jdbc\lib\ojdbc7.jar" dblogger1
```

この例では、ホスト名とデフォルトのポートとチャンネルを使用して、クライアント・モードのファイル・ロガーが作成されます。

```
fteCreateLogger -loggerType FILE -loggerQMGr CORDQM -loggerQMGrHost cordqm.ibm.com
                -fileLoggerMode CIRCULAR -fileSize 10MB -fileCount 10 FL1
```

この例では、次のホスト名、ポート、チャンネルを使用して、クライアント・モードのファイル・ロガーが作成されます。

```
fteCreateLogger -loggerType FILE -loggerQMGr CORDQM -loggerQMGrHost cordqm.ibm.com
                -loggerQMGrPort 4444 -loggerQMGrChannel LOGGER_CHANNEL -fileLoggerMode CIRCULAR -fileSize 10MB
                -fileCount 10 FL1
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

fteCreateMonitor: MFT リソース・モニターの作成

fteCreateMonitor コマンドは、コマンド行から新規リソース・モニターを作成して開始します。Managed File Transfer を使用してリソース (例えば、ディレクトリーの内容) をモニターして、トリガー条件が満たされたときに、指定したタスク (ファイル転送など) を開始することができます。

目的

fteCreateMonitor コマンドは、Managed File Transfer エージェントを使用して新規リソース・モニターを作成して開始する場合に使用します。例えば、次のようにリソース・モニターを使用できます。まず、外部アプリケーションで、既知のディレクトリーに1つ以上のファイルを配置します。そして、処理の完了時に外部アプリケーションによって、モニター対象ディレクトリーにトリガー・ファイルを配置します。そうすると、トリガー・ファイルが検出されて、定義されたファイルの転送が開始し、既知のディレクトリーのファイルが宛先エージェントにコピーされます。

-ox パラメーターおよび **-ix** パラメーターを使用して、リソース・モニター構成を XML ファイルにエクスポートしたり XML ファイルからインポートしたりできます。**fteCreateMonitor** コマンドでこのファイルをインポートすると、**fteCreateMonitor** コマンドで XML ファイルをエクスポートするときに指定したリソース・モニターと同じパラメーターで新規リソース・モニターが作成されます。また、**-f** パラメーターおよび **-c** パラメーターを使用して、モニター構成を動的に上書きできます。

注: エージェントに作成できるリソース・モニターの数に制限はなく、すべてのモニターは同じ優先度で実行されます。モニター対象リソースのオーバーラップ、トリガー条件の矛盾、およびリソースをポーリングする頻度の影響を考慮してください。詳しくは、[MFT のリソース・モニターの概念](#)を参照してください。

fteCreateMonitor コマンドは、プロトコル・ブリッジ・エージェントではサポートされません。

ヒント: fteListMonitors コマンドを使用して、リソース・モニター構成を XML ファイルにエクスポートすることもできます。

- **-ox** を指定して **fteListMonitors** コマンドを使用する場合は、1つのリソース・モニターの定義をエクスポートできます。
- **V 9.0.5** IBM MQ 9.0.5 から、**-od** を指定して **fteListMonitor** コマンドを使用すると、複数のリソース・モニター定義が指定のディレクトリーにエクスポートされます。**-od** オプションを使用して、指定のディレクトリーに1つのリソース・モニター定義をエクスポートすることも可能です。

このコマンドの詳細については、「[fteListMonitors476 ページの『fteListMonitors: MFT リソース・モニターのリスト』](#)」を参照してください。

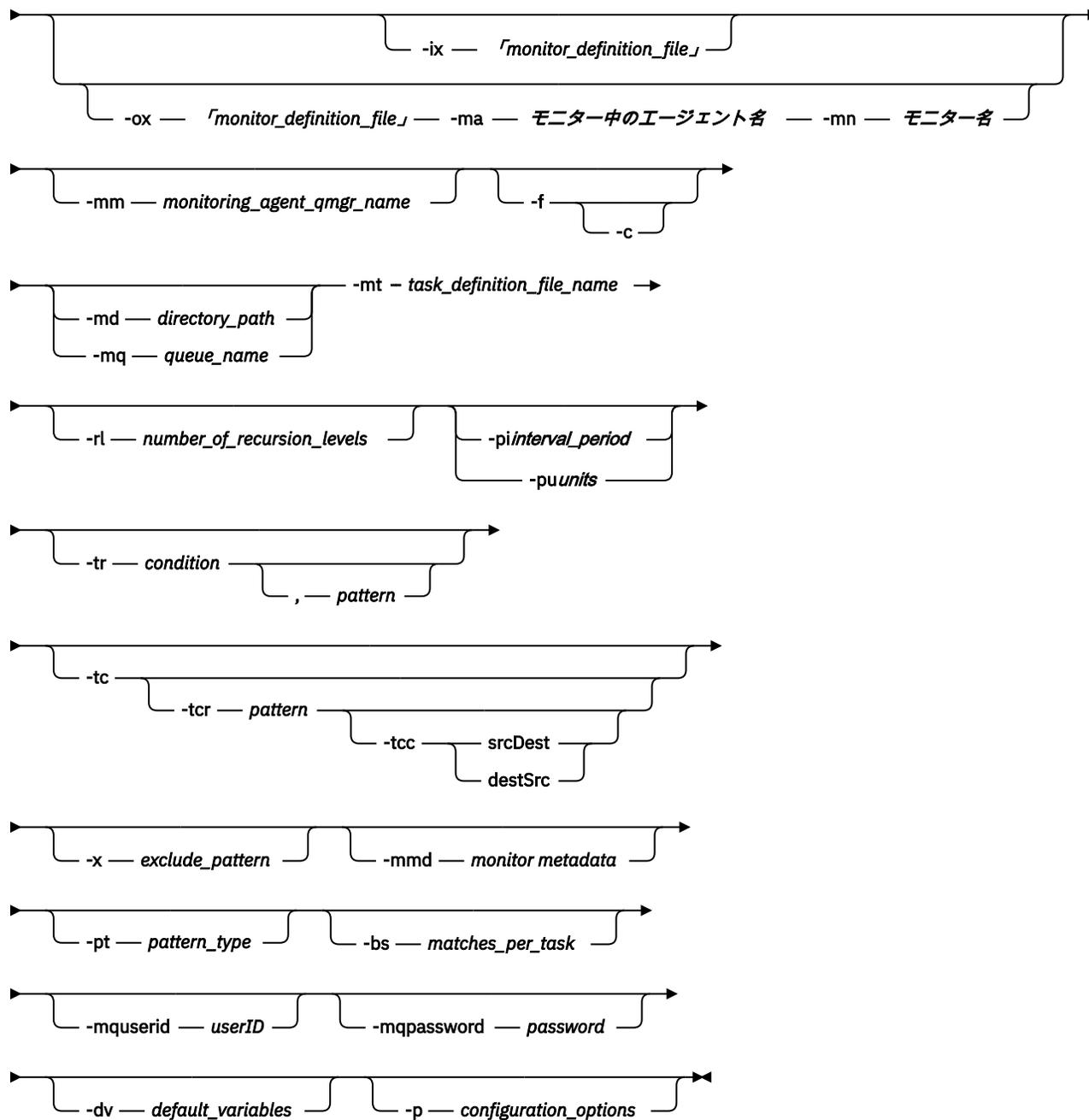
特殊文字

特殊文字が含まれているパラメーター値を使用する場合には、予期しない形で特殊文字がコマンド・シェルで解釈されないように注意してください。例えば、スペース、引用符 (単一または二重)、スラッシュ、バックスラッシュ文字などを含む完全修飾ファイル・パスや名前は、直接コマンド自体に渡されずに、コマンド・シェルで解釈されてしまう可能性があります。文字がコマンド・シェルによって解釈される動作を回避するには、パラメーター全体を二重/単一引用符で囲むか、またはコマンド・シェルのエスケープ・シーケンスを使用して特殊文字をエスケープします。

構文

fteCreateMonitor

▶ fteCreateMonitor →



Parameters

-ix (*xml_filename*)

オプション。リソース・モニター構成をXMLファイルからインポートします。

-ox (*xml_filename*)

オプション。このパラメーターは、`-ma` パラメーターおよび `-mn` パラメーターと一緒に指定する必要があります。リソース・モニター構成をXMLファイルにエクスポートします。

-mn (monitor_name)

必須。このモニターに割り当てる名前。モニター名は、モニター・エージェントに対して固有である必要があります。ただし、モニターを削除してから、同じ名前のモニターを作成できます。

リソース・モニター名の最大長は 256 文字です。リソース・モニター名は、大/小文字を区別しません。小文字または大/小文字混合で入力されたリソース・モニター名は大文字に変換されます。リソース・モニター名には、アスタリスク (*)、パーセント (%)、または疑問符 (?) を使用できません。

-ma (monitoring_agent_name)

必須。リソース・モニターを実行するエージェントの名前。このモニター・エージェントは、トリガー対象のモニター・タスクのソース・エージェントである必要があります。

-mm (monitoring_agent_qmgr_name)

モニター・エージェントの接続先キュー・マネージャーの名前。モニター・エージェントとソース・エージェントが同一である必要があるため、このキュー・マネージャーはソース・エージェントのキュー・マネージャーでもあります。

注: `fteCreateMonitor` コマンドは、Managed File Transfer トポロジーのコマンド・キュー・マネージャーに接続します。コマンド・キュー・マネージャーがモニター・エージェントのエージェント・キュー・マネージャーでもある場合、このパラメーターはオプションです。それ以外の場合、このパラメーターは必須です。

-f

オプション。このパラメーターは、リソース・モニター構成を上書きする場合に使用します。例えば、選択するリソース・モニター名がリソース・モニター・エージェントに既に存在していて、削除してモニターを同じ名前で再作成するのではなく、それを更新する場合は、このパラメーターを使用すると、エージェントによってモニター・プロセスが再開されます。

-c

オプション。このパラメーターにより、更新されるリソース・モニターの履歴がクリアされ、リソース・モニターでトリガー条件が再検査されます。このパラメーターは **-f** パラメーターと共にのみ使用できます。

-md (directory_path)

オプション。モニターするディレクトリー・パスの絶対名。 **-ix** パラメーターまたは **-ox** パラメーターを使用している場合以外は、 **-md** パラメーターまたは **-mq** パラメーターのいずれかを指定する必要があります。

-mq (queue_name)

オプション。モニターするキューの名前。このキューは、モニター・エージェントのキュー・マネージャーに存在していなければなりません。 **-ix** パラメーターまたは **-ox** パラメーターを使用している場合以外は、 **-md** パラメーターまたは **-mq** パラメーターのいずれかを指定する必要があります。

-mt (task_definition_file_name)

必須。トリガー条件が満たされたときに実行する、タスク定義を含む XML 文書の名前。詳しくは、161 ページの『[転送定義ファイルの使用](#)』を参照してください。転送定義 XML 文書へのパスは、`fteCreateMonitor` コマンドの実行元のローカル・ファイル・システムにある必要があります。ファイルのパスを指定しない場合、コマンドは現行作業ディレクトリーでファイルを探します。 **-ix** パラメーターまたは **-ox** パラメーターを使用している場合以外は、 **-mt** は必須パラメーターです。

`fteCreateTransfer` コマンドで **-gt** パラメーターを使用して、ファイル転送要求を含むテンプレート XML 文書を生成できます。モニターは、転送テンプレートをタスク定義として使用します。

V 9.0.1 `fteCreateMonitor` コマンドを実行するときに、転送リカバリー・タイムアウトの **-rt** パラメーターを **-gt** パラメーターとともに使用することもできます。停止した転送のリカバリーをソース・エージェントが試行し続ける時間を秒単位で設定できます。モニターが使用する転送定義とともにリカバリー・タイムアウト・パラメーターが XML 文書に組み込まれます。このパラメーターの設定方法については、`fteCreateTransfer` コマンドを参照してください。

z/OS z/OSでは、z/OS UNIX システム・サービスの UNIX ファイルに、タスク定義文書を保管する必要があります。タスク定義文書は、z/OS 順次ファイルおよび PDS メンバーに保管することはできません。

IBM i IBM iでは、タスク定義文書を Integrated File System に保管する必要があります。

-rl (number_of_recursion_levels)

オプション。モニター対象のルート・ディレクトリーからの、再帰的にモニターする階層。つまり、掘り下げるサブディレクトリーの階層数。例えば、C:\wmqfte\monitor をモニター対象ルート・ディレクトリーとした場合の、次のようなディレクトリー構造を例として説明します。

```
C:\wmqfte\monitor
C:\wmqfte\monitor\reports
C:\wmqfte\monitor\reports\2009
C:\wmqfte\monitor\reports\2009\April
```

-rl 2 を指定した場合、Managed File Transfer は、C:\wmqfte\monitor\reports\2009 ディレクトリーとその兄弟ディレクトリーのみを検索します。

C:\wmqfte\monitor\reports\2009\April ディレクトリーは無視されます。デフォルトでは、再帰は行われないように設定されています。

-pi (interval_period)

オプション。ディレクトリーをモニターする間隔。ポーリング間隔は、正整数値でなければなりません。-pi のデフォルト値は 1 です。

-pu (units)

オプション。モニター・ポーリング間隔の時間単位。-pu パラメーターを指定した場合には、-pi パラメーターも指定する必要があります。-pu のデフォルト値は minutes (分) です。以下のいずれかのオプションを指定します。

seconds

minutes

hours

日

-tr

オプション。定義したタスクが実行されるためにはトリガー条件が満たされなければならないことを指定します。ソース・エージェントに従って条件が満たされない場合、モニター・タスク (例えば、ファイル転送) は開始されません。トリガー条件は、2つのオプションの部分、つまり、条件とパターンで構成されており、コンマで区切ります。以下のいずれかのフォーマットを指定します。

- *condition,pattern*

ここで *condition* は以下のいずれかの値になります。

match

条件を満たしたトリガーごとに、定義されているタスクが実行されます。match はデフォルト値です。

例えば、match が *.go で、ファイル LONDON.go および MANCHESTER.go がある場合、タスクが LONDON.go に対して実行され、別のタスクが MANCHESTER.go に対して実行されます。

前のポーリングと同じトリガー・ファイルが存在する (つまり、ファイルが変更されていない) 場合、このファイルはトリガー条件を満たしません。つまり、一致するトリガー・ファイルは、新しいファイルでなければならず、しかも最後のポーリング時と定義済みのタスクの実行時の間に変更されたファイルでなければなりません。

noMatch

モニター対象ディレクトリー内のファイルが1つもパターンに一致しないかどうか。つまり、モニター対象ディレクトリー内のいかなるファイルも存在しない場合に、この条件は満たされます。モニター作成時にトリガー条件に一致するファイルがない場合、モニターは即座に開始します。しかし、ファイル・マッチングが検出されてその後除去されるまで再開されません。

noSizeChange=*n*

ディレクトリー内のファイルの中で、パターンに一致して、かつファイル・サイズが *n* ボーリング間隔の間変化しなかったものが1つ以上存在するかどうか。 *n* の値は正整数です。

fileSize>=*size*

ディレクトリー内のファイルの中で、パターンに一致して、かつファイル・サイズが *size* 以上であるものが1ファイル以上存在するかどうか。 *size* の値は、整数とオプションのサイズ単位 (B、KB、MB、または GB) を組み合わせて指定します。例えば、 `fileSize">"=10KB` のように指定します。サイズ単位を指定しなければ、バイト単位がデフォルトで使用されます。すべてのオペレーティング・システムにおいて、この例で示すように、コマンド行で `filesize` オプションを指定した場合は、より大記号 (>) は二重引用符で囲まなければなりません。

このパターンは、ワイルドカードまたは Java 正規表現フォーマットのファイル・パターン・マッチング・シーケンスです。パターンのデフォルト値は `*`、つまり、任意のファイルとの一致で、デフォルト・フォーマットはワイルドカード・フォーマットです。 `-pt` を使用してパターンのフォーマットを指定します。

例えば、モニター対象のディレクトリーに `.go` という接尾部のファイルが存在すると、以下のトリガー条件が満たされます。

```
-tr match,*.go
```

モニター対象ディレクトリー内に接尾部 `.stop` を持つファイルがない場合に、次のトリガー条件が満たされます。

```
-tr noMatch,*.stop
```

`condition, pattern` は、 `-md` パラメーターも指定する場合にのみ指定できます。

```
condition
```

ここで `condition` は以下のいずれかの値になります。

queueNotEmpty

モニターされるキューは空ではありません。つまり、モニターされるキューに何らかの IBM MQ メッセージがあれば、この条件は満たされます。キュー上のすべてのメッセージに対して単一のタスクが実行されます。

completeGroups

モニターされるキューには、完全なグループがあります。つまり、モニターされるキュー上で、何らかの IBM MQ メッセージ・グループが完全であれば、この条件は満たされます。キュー上の、それぞれの完全グループに、個別のタスクが実行されます。

グループ内にない1つのメッセージがキュー上に置かれると、このメッセージは完全なグループとして扱われ、この単一メッセージに対してタスクが実行されます。

`condition` は、 `-mq` パラメーターも指定する場合にのみ指定できます。

作成するモニターごとに、 `-tr` パラメーターを1回限り指定できます。

-tc

オプション。トリガーされるファイルに、転送要求を生成するための1つ以上のファイル・パスが含まれていることを示します。トリガー・ファイルの内容のデフォルト形式は、行ごとに1ファイルのエントリーです。 `source file path` または `source file path, destination file path` のいずれかとしてファイル・パスを指定します。このパラメーターは、ディレクトリー・モニター・トリガー `match` および `noSizeChange` にのみ使用可能です。

-tcr (pattern)

オプション。トリガー・ファイルを構文解析するための置換正規表現を指定します。 **-tcr** パラメーターを指定した場合には、 **-tc** パラメーターも指定する必要があります。

各行のエントリを1つか2つのキャプチャー・グループで完全に解析するようにパターンを設計します。グループ1でソース・ファイル・パスを定義し、オプションのグループ2で宛先ファイル・パスを定義します。これがデフォルトの動作ですが、 **-tcc** パラメーターを使用して変更することができます。

詳細および例については、『[189 ページの『トリガー・ファイルの使用』](#)』を参照してください。

-tcc

オプション。正規表現キャプチャー・グループの順序を定義します。

srcDest

デフォルト値で、グループ1がソース・ファイル・パスで、グループ2が宛先ファイル・パス。

destSrc

srcDest の逆。グループ1が宛先ファイル・パスで、グループ2がソース・ファイル・パスです。destSrc の正規表現には、2つのキャプチャー・グループがあるようにします。

-tcc パラメーターを指定した場合には、 **-tcr** パラメーターも指定する必要があります。

-x (exclude_pattern)

オプション。トリガー・パターン・マッチングから除外されるファイルを指定します。トリガー・パターンは **-tr** パラメーターで指定されます。

このパターンは、ワイルドカードまたは Java 正規表現フォーマットのファイル・パターン・マッチング・シーケンスです。デフォルト・フォーマットはワイルドカード・フォーマットです。 **-pt** パラメーターを使用してパターンのフォーマットを指定します。

-mmd (monitor metadata)

オプション。モニターの出口点に渡されるユーザー定義のメタデータを示します。パラメーターには、名前ペアをコンマ区切りで1つ以上含められます。それぞれの名前ペアの構成は、`name=value` になります。コマンド内では **-mmd** パラメーターを複数回使用できます。

-pt (pattern_type)

オプション。 **-tr** パラメーターおよび **-x** パラメーターで使用されるパターンのタイプ。有効な値は次のとおりです。

ワイルドカード

パターンがワイルドカード・パターンとして評価されます。アスタリスク (*) はゼロ個以上の文字と一致し、疑問符 (?) はちょうど1つの文字と一致します。これがデフォルトです。

正規表現

パターンが Java 正規表現として評価されます。詳しくは、『[659 ページの『MFT が使用する正規表現』](#)』を参照してください。

-bs (matches_per_task)

オプション。単一タスクに含めるトリガー・マッチングの最大数。例えば、`matches_per_task` に値5が指定されているときに、1回のポーリング間隔で9個のトリガー・マッチングが発生すると、2つのタスクが実行されます。最初のタスクがトリガー1から5に対応し、2番目のタスクがトリガー6から9に対応します。`matches_per_task` のデフォルト値は1です。

-bs パラメーターは、**-mt** パラメーターに提供するタスク定義 XML が managedTransfer である場合にのみサポートされます。managedCall では、**-bs** パラメーターはサポートされません。

-mquserid (userID)

オプション。コマンド・キュー・マネージャーで認証するユーザー ID を指定します。

-mqpassword (password)

オプション。コマンド・キュー・マネージャーで認証するパスワードを指定します。 **-mquserid** パラメーターも一緒に指定する必要があります。 **-mquserid** を指定した場合に **-mqpassword** を指定しないと、関連付けるパスワードを入力するというプロンプトが出されます。パスワードは表示されません。

-dv (default_variables)

オプション。キューをモニターする際に変数置換に使用できるデフォルト変数のコンマ区切りリスト。値は、キー値ペアの形式です。以下に例を示します。

```
-dv size=medium,color=blue
```

変数置換について詳しくは、180 ページの『変数置換を使用した MFT リソース・モニター・タスクのカスタマイズ』を参照してください。-dv パラメーターを指定できるのは、-mq パラメーターを指定した場合に限られます。

-? または -h

オプション。コマンド構文を表示します。

-p (configuration_options)

オプション。このパラメーターは、転送を取り消すために使用する構成オプションのセットを決定します。慣例として、このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティ・ファイルのセットを使用します。

このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。

例

この例では、モニター・エージェント MYAGENT を使用して、MYMONITOR という新規リソース・モニターが作成されます。5 MB を超えるファイルがディレクトリー C:\wmqfte\monitors に存在するというトリガー条件が満たされると、ファイル C:\templates\transfer_reports.xml に定義されたファイル転送が開始されます。MYAGENT は、C:\templates\transfer_reports.xml に定義されたファイル転送のソース・エージェントでもあります。

```
fteCreateMonitor -ma MYAGENT -md C:\wmqfte\monitors -mn MYMONITOR -mt C:\templates\transfer_reports.xml -tr fileSize">"=5MB,*.*go
```

この例では、エージェント AGENT1 を使用する MONITOR1 というリソース・モニターが作成されて、5 MB より大きいファイルが転送され、XML ファイル monitor.xml にエクスポートされます。

```
fteCreateMonitor -ox monitor.xml -ma AGENT1 -mn MONITOR1 -mt task.xml -tr "fileSize>=5MB,*.*zip"
```

次に、XML ファイルがインポートされ、10MB より大きいファイルが除外されるように変更されます。

```
fteCreateMonitor -ix monitor.xml -x "fileSize>=10MB,*.*zip" -f
```

この例では、エージェント MYAGENT を使用して、MYMONITOR という新規リソース・モニターが作成されます。

```
fteCreateMonitor -ma MYAGENT -md c:\wmqfte -mn MYMONITOR -mt c:\templates\transfer_reports.xml -tr "fileSize>=5MB,*.*go"
```

ただし、ここではトリガーが c:\wmqfte\monitors ではなく c:\wmqfte をモニターするように誤って設定されています。モニター・ディレクトリーが修正され、モニターを更新するために -f (上書き) パラメーターおよび -c (履歴のクリア) パラメーターが付けられて、fteCreateMonitor 要求が即時に再発行されます。

```
fteCreateMonitor -ma MYAGENT -md c:\wmqfte\monitors -mn MYMONITOR -mt c:\templates\transfer_reports.xml -tr "fileSize>=5MB,*.*go" -f -c
```

戻りコード

戻りコード	説明
0	コマンドは正常に完了しました。
1	コマンドは失敗しました。

fteCreateTemplate: 新規ファイル転送テンプレートの作成

fteCreateTemplate コマンドは、将来の使用のために保持できるファイル転送テンプレートを作成します。必須パラメーターは **-tn** (*template_name*) パラメーターのみです。その他のパラメーターはすべてオプションです。ただし、ソース・ファイル仕様を指定する場合には、宛先ファイルも提供する必要があります。同様に、宛先ファイルを指定する場合には、ソース・ファイル仕様も指定する必要があります。

目的

fteCreateTemplate コマンドは、転送の詳細を後日使用するときまで保管する、ファイル転送テンプレートを作成するために使用します。転送テンプレートを使用して、繰り返しの転送または複雑な転送を行うための共通のファイル転送設定を保管します。転送テンプレートを作成した後、IBM MQ Explorer を使用してテンプレートを送信します。転送テンプレートはコマンド行からは送信できません。

fteCreateTemplate コマンドを使用して作成する転送テンプレートは、**fteCreateTransfer** コマンドで **-gt** パラメーターを使用して作成する XML メッセージとは異なります。2つの異なるタイプのテンプレートを互換的に使用することはできません。

fteCreateTemplate コマンドは、IBM MQ ネットワークに接続してから調整キュー・マネージャーに経路指定できる任意のシステムから実行できます。特に、このコマンドを実行するには、このシステムに Managed File Transfer がインストール済みであること、および IBM MQ ネットワークと通信するために、このシステムで Managed File Transfer コンポーネントが構成済みであることが必要です。

このコマンドは `command.properties` ファイルを使用して、Managed File Transfer トポロジーのコマンド・キュー・マネージャーに接続します。`command.properties` ファイルに **connectionQMGrHost** プロパティが含まれる場合、このコマンドは CLIENT トランスポートを使用してコマンド・キュー・マネージャーに接続します。それ以外の場合には、このコマンドは BINDINGS トランスポートを使用してコマンド・キュー・マネージャーに接続します。`command.properties` ファイルが存在しない場合、コマンドは失敗し、次のエラーが生成されます。

```
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"
```

詳しくは、[command.properties](#) ファイルを参照してください。

ファイル転送に対して、ソース・ファイルは複数指定できますが、宛先エージェントは1つだけです。1つのファイルを複数の宛先エージェントに転送することはサポートされていません。ただし、複数のソース・ファイルを、1つの宛先エージェント上の複数の宛先ファイルに転送することはできます。

ファイルを転送する方法については、[630 ページの『ファイルの転送に関するガイドライン』](#)を参照してください。

特殊文字

特殊文字が含まれているパラメーターを使用する場合には、予期しない形で特殊文字がコマンド・シェルで解釈されないように注意してください。  例えば、単一引用符を含む完全修飾データ・セット名およびアスタリスク文字を含むソース指定は、転送要求で渡されずに、コマンド・シェルで解釈されてしまう可能性があります。文字がコマンド・シェルによって解釈される動作を回避するには、[436 ページの『例』](#)の最後の2例のように、パラメーター全体を二重引用符で囲むか、コマンド・シェルのエスケープ・シーケンスを使用して特殊文字をエスケープします。

相対パス

fteCreateTemplate コマンドは、相対ファイル・パスの使用をサポートします。分散システム

z/OS および z/OS UNIX システム・サービスでは、デフォルトでは、パスは、エージェントがユーザーとして動作しているホーム・ディレクトリーを基点とした相対パスとみなされます。パス名が相対パスと評価されるディレクトリーを変更するには、`agent.properties` ファイルに `transferRoot` プロパティを設定します。このファイルは `MQ_DATA_PATH/mqft/config/coordination_qmgr/agents/agent_name` ディレクトリーにあります。次の行をファイルに追加します。

```
transferRoot=directory_name
```

Windows パスを拡張するか、これを UNIX フォーマットで記述する必要があります。例えば、`C:\TransferRoot` を `C:\\TransferRoot` または `C:/TransferRoot` と指定します。

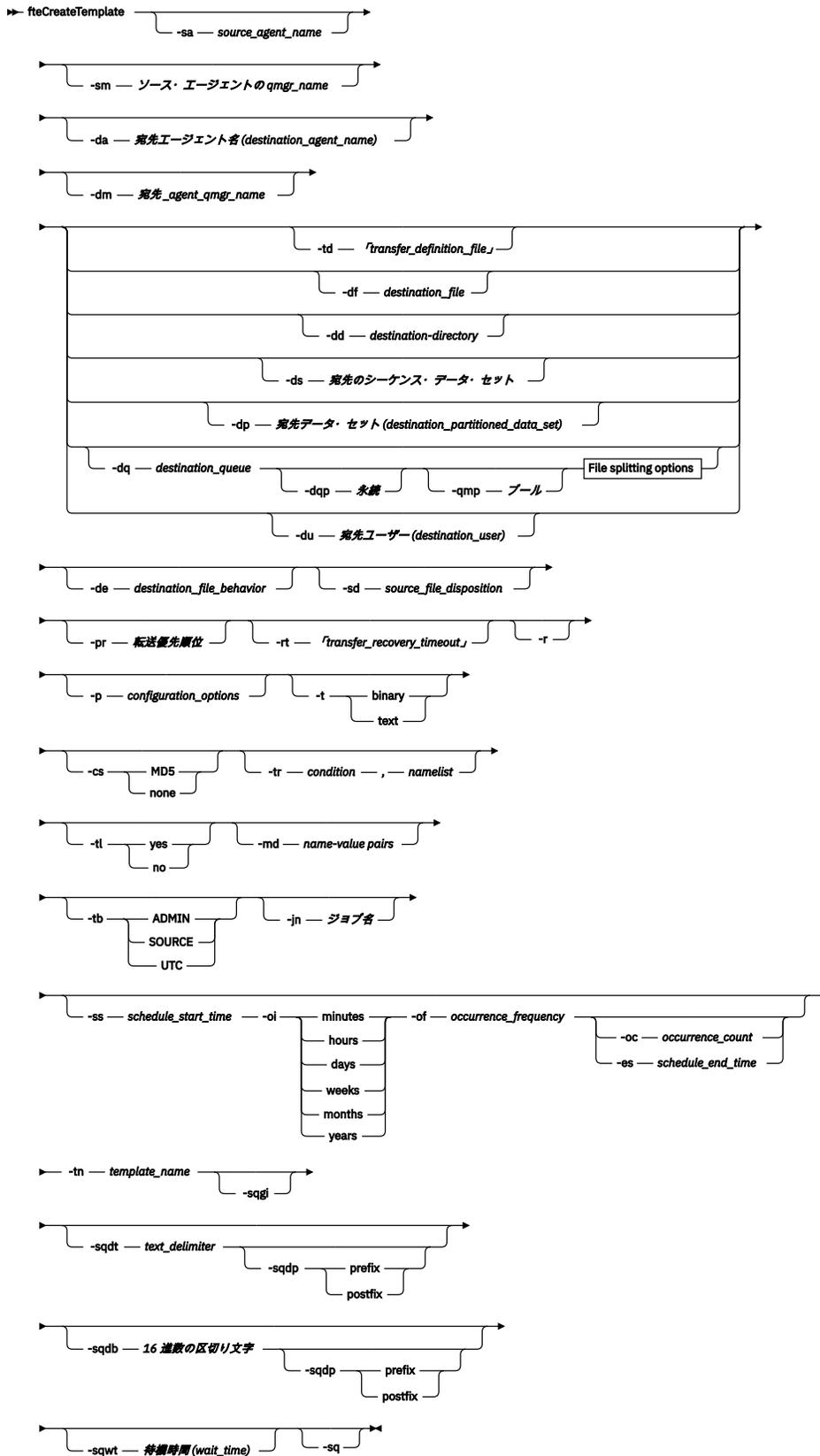
z/OS z/OS では、デフォルトでは、エージェントが現在動作しているユーザー名が、高位修飾子接頭部として完全に修飾されていないデータ・セット仕様に追加されます。例えば、`//ABC.DEF` のようになります。接頭部としてデータ・セット名に追加される値を変更するには、`transferRootHLQ` プロパティを `agent.properties` ファイルで設定します。このファイルは `MQ_DATA_PATH/mqft/config/coordination_qmgr/agents/agent_name` ディレクトリーにあります。次の行をファイルに追加します。

```
transferRootHLQ=prepend_value
```

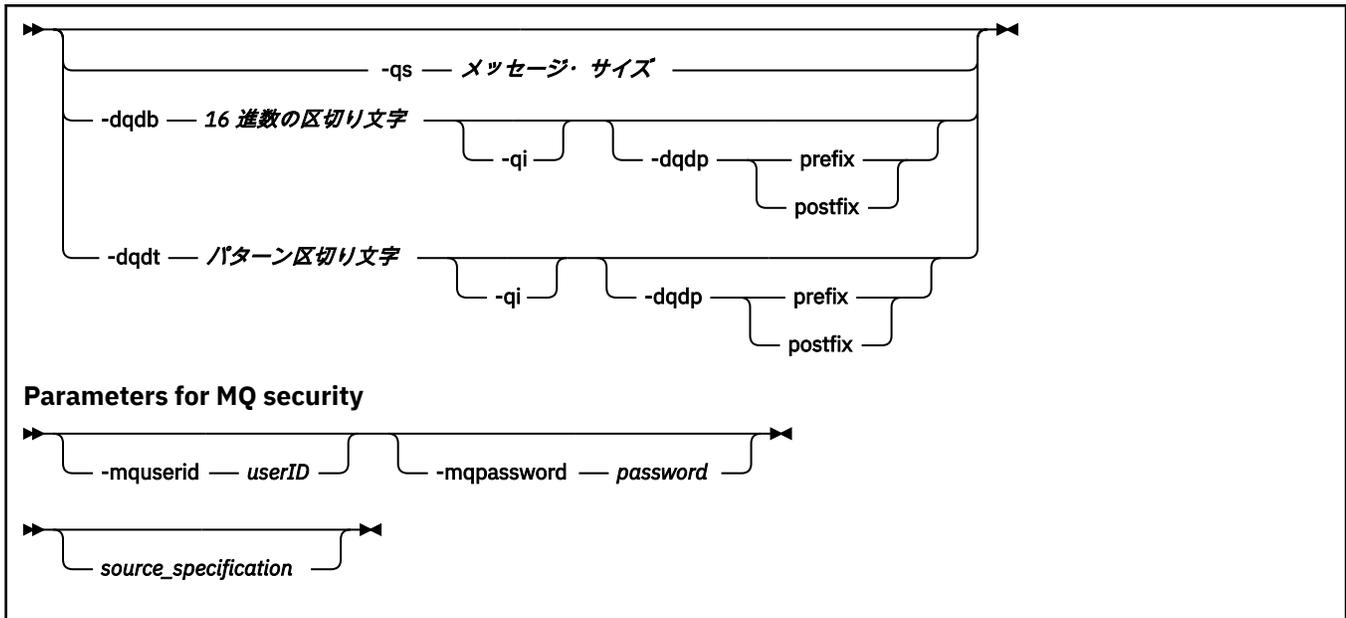
z/OS ただし、z/OS システム上の `Connect:Direct` ノードに関する転送では、データ・セット仕様は完全修飾名として解釈されます。データ・セット名に高位修飾子が追加されることはありません。

構文

fteCreateTemplate



File splitting options



Parameters

-sa source_agent_name

オプション。ソース・ファイルの転送元のエージェントの名前。テンプレートの作成時にこのエージェント名を指定しない場合、テンプレートの使用時にソース・エージェント名を指定する必要があります。

-sm source_agent_qmgr_name

オプション。ソース・エージェントの接続先のキュー・マネージャーの名前。

-sm パラメーターを指定しない場合、どのキュー・マネージャーを使用するかは、ソース・エージェントの名前に基づいて、使用している構成オプションのセットによって決定されます。これらのオプションを使用してキュー・マネージャー名を決定できない場合、転送テンプレートの作成は失敗します。例えば、ソース・エージェントの `agent.properties` ファイルが見つからない場合、テンプレートの作成は失敗します。

-da destination_agent_name

オプション。ファイルの転送先のエージェントの名前。テンプレートの作成時に宛先エージェント名を指定しない場合、テンプレートの使用時に宛先エージェント名を指定する必要があります。

-dm destination_agent_qmgr_name

オプション。宛先エージェントの接続先のキュー・マネージャーの名前。

-dm パラメーターを指定しない場合、どのキュー・マネージャーを使用するかは、宛先エージェントの名前に基づいて、使用している構成オプションのセットによって決定されます。これらのオプションを使用してキュー・マネージャー名を決定できない場合、転送テンプレートの作成は失敗します。例えば、宛先エージェントの `agent.properties` ファイルが見つからない場合、テンプレートの作成は失敗します。

-td transfer_definition_file

オプション。転送のためのソースおよび宛先のファイル仕様を1つ以上定義するXML文書の名前。

-td、**-df**、**-dd**、**-ds**、**-dq**、**-du**、および**-dp** パラメーターのいずれかが必要です。**-td** パラメーターを指定する場合、ソース・ファイルを指定することも、**-df**、**-dd**、**-ds**、**-dp**、**-dq**、**-du**、**-sd**、**-r**、**-de**、**-t**、または**-cs** パラメーターを指定することもできません。

fteCreateTemplate コマンドは、現行ディレクトリーを基準として、転送定義ファイルの場所を探します。相対パス表記を使用して転送定義ファイルの場所を指定することができない場合は、代わりに、転送定義ファイルの完全修飾パスとファイル名を使用します。

z/OS z/OS では、z/OS UNIX システム・サービスの UNIX ファイルに、転送定義ファイルを保管する必要があります。z/OS の順次ファイルまたは PDS メンバーに、転送定義ファイルを保管することはできません。

IBM i IBM i では、統合ファイル・システムに転送定義ファイルを保管する必要があります。

詳しくは、『[転送定義ファイルの使用](#)』を参照してください。

-df destination_file

オプション。宛先ファイルの名前。宛先エージェントが実行されているシステムの有効なファイル名を指定します。

宛先エージェントが Connect:Direct ブリッジ・エージェントの場合は、宛先ファイルが `connect_direct_node_name:file_path` という形式で指定されます。Connect:Direct ブリッジ・エージェントはこの形式で指定されたファイル・パスのみを受け入れます。**z/OS** 宛先エージェントが Connect:Direct ブリッジ・エージェントであり、宛先が PDS メンバーになっている場合は、さらに **-de** パラメーターで `overwrite` という値を指定する必要があります。

-td、**-df**、**-dd**、**-ds**、**-dq**、**-du**、および **-dp** パラメーターのいずれかが必要です。**-df** パラメーターを指定する場合、**-td**、**-dd**、**-dp**、**-dq**、**-du**、または **-ds** パラメーターを指定できません。これらのパラメーターは相互に排他的だからです。

-dd destination_directory

オプション。ファイルの転送先となるディレクトリーの名前。宛先エージェントが実行されているシステムの有効なディレクトリー名を指定します。

宛先エージェントが Connect:Direct ブリッジ・エージェントの場合、宛先ディレクトリーは `connect_direct_node_name:directory_path` という形式で指定されます。宛先エージェントが Connect:Direct ブリッジ・エージェントであり、宛先が PDS になっている場合は、さらに **-de** パラメーターで `overwrite` という値を指定する必要があります。

-td、**-df**、**-dd**、**-ds**、**-dq**、**-du**、および **-dp** パラメーターのいずれかが必要です。**-dd** パラメーターを指定する場合、**-td**、**-df**、**-dp**、**-dq**、**-du**、または **-ds** パラメーターを指定できません。これらのパラメーターは相互に排他的だからです。

z/OS -ds destination_sequential_data_set

z/OS のみ。オプション。ファイルの転送先の順次データ・セットまたは PDS メンバーの名前。順次データ・セット名または区分データ・セット・メンバーを指定します。

-td、**-df**、**-dd**、**-ds**、**-dq**、**-du**、および **-dp** パラメーターのいずれかが必要です。**-ds** パラメーターを指定する場合、**-td**、**-dd**、**-df**、**-dq**、**-du**、または **-dp** パラメーターを指定できません。これらのパラメーターは相互に排他的だからです。

データ・セット名の構文は次のとおりです。

```
//data_set_name{;attribute;...;attribute}
```

または

```
//pds_data_set_name(member_name){;attribute;...;attribute}
```

つまり、データ・セット名指定子の前に `//` があり、その指定子の後にはオプションで、セミコロンで区切られたいくつかの属性が続きます。

データ・セットが Connect:Direct ノードにある場合は、データ・セット名の接頭部としてノード名を使用する必要があります。以下に例を示します。

```
CD_NODE1://'OBJECT.LIB';RECFM(F,B);BLKSIZE(800);LRECL(80)
```

宛先エージェントが Connect:Direct ブリッジ・エージェントであり、宛先が PDS メンバーになっている場合は、さらに **-de** パラメーターで **overwrite** という値を指定する必要があります。Connect:Direct ノードとのデータ・セット転送の詳細については、634 ページの『Connect:Direct ノードとの間のデータ・セット転送』を参照してください。

Managed File Transfer エージェントだけがかわっている転送の場合は、データ・セット名の部分を単一引用符文字で囲むと、その指定が完全修飾データ・セット名になります。データ・セット名を単一引用符文字で囲まないと、システムによって宛先エージェントのデフォルトの高位修飾子 (transferRootHLQ エージェント・プロパティの値か、transferRootHLQ が設定されていない場合は、エージェントを実行しているユーザー ID) が追加されます。

注: **z/OS** ただし、z/OS システム上の Connect:Direct ノードに関する転送では、データ・セット仕様は完全修飾名として解釈されます。データ・セット名に高位修飾子が追加されることはありません。データ・セット名を単一引用符文字で囲んだ場合でも、そのような動作になります。

データ・セット属性は、データ・セットを作成するため、または既存のデータ・セットが互換性を持つようにするために使用されます。データ・セット属性の指定は、BPXWDYN に適した形式になります (詳細については、『Requesting dynamic allocation』を参照してください)。エージェントが宛先データ・セットを作成する場合、BPXWDYN 属性 DSN(data_set_name) NEW CATALOG MSG(numeric_file_descriptor) が自動的に指定されます。ここで、numeric_file_descriptor は Managed File Transfer が生成するファイル記述子です。データ・セットからのデータ・セットへの転送では、ソースから RECFM、LRECL、および BLKSIZE の属性が、新規宛先データ・セット用に選択されます。新規宛先データ・セットの SPACE 設定は、Managed File Transfer によって設定されず、システムのデフォルトが使用されることに注意してください。そのため、新規データ・セットが作成されるたびに、SPACE 属性を指定することをお勧めします。agent.properties ファイル内の **bpxwdynAllocAdditionalProperties** プロパティを使用して、すべての転送に適用する BPXWDYN オプションを設定できます。詳しくは、550 ページの『MFT agent.properties ファイル』を参照してください。

いくつかの BPXWDYN オプションは、**fteCreateTemplate** コマンド、**fteCreateTransfer** コマンド、または agent.properties ファイル内の **bpxwdynAllocAdditionalOptions** プロパティを使用するときに指定してはなりません。これらのプロパティのリストについては、642 ページの『MFT で使用できない BPXWDYN のプロパティ』を参照してください。

ファイルまたはデータ・セットを磁気テープに転送すると、磁気テープ内の既存データ・セットが置き換えられます。新しいデータ・セットの属性は、転送定義で渡された属性を基にして設定されます。属性は、何も指定されていなければ、ソース・データ・セットと同様、またはデフォルト値 (ソースがファイルである場合) に設定されます。既存の磁気テープ・データ・セットの属性は無視されます。

-ds パラメーターは、宛先エージェントがプロトコル・ブリッジ・エージェントである場合はサポートされません。

z/OS **-dp destination_partitioned_data_set**

z/OS のみ。オプション。ファイルの転送先となる宛先 PDS の名前。区分データ・セット名を指定します。PDS が転送の結果として作成されると、この PDS はデフォルトで PDSE として作成されます。DSNTYPE=PDS を指定することにより、デフォルトをオーバーライドすることができます。

-td、**-df**、**-dd**、**-ds**、**-dq**、**-du**、および **-dp** パラメーターのいずれかが必要です。**-dp** パラメーターを指定する場合、**-td**、**-dd**、**-df**、**-dq**、**-du**、または **-ds** パラメーターを指定できません。これらのパラメーターは相互に排他的だからです。

PDS データ・セット名の構文は次のとおりです。

```
//pds_data_set_name{;attribute;..;attribute}
```

データ・セット名の構文は、**-ds** (*destination_sequential_data_set*) パラメーターで説明しているものと同じです。Connect:Direct ノードにあるデータ・セットを指定するための構文の詳細はすべて、**-dp** パラメーターにも当てはまります。宛先エージェントが Connect:Direct ブリッジ・エージェントの場合は、さらに **-de** パラメーターで *overwrite* という値を指定する必要があります。

-dp パラメーターは、宛先エージェントがプロトコル・ブリッジ・エージェントである場合はサポートされません。

-du destination_user

オプション。ファイルの転送先の宛先ファイル・スペースを使用するユーザーの名前。

-td、**-df**、**-dd**、**-ds**、**-dp**、**-du**、および **-dq** パラメーターのいずれかが必要です。**-du** パラメーターを指定する場合、**-td**、**-dd**、**-df**、**-dp**、**-dq**、または **-ds** パラメーターを指定できません。これらのパラメーターは相互に排他的だからです。

-du パラメーターは、宛先エージェントがプロトコル・ブリッジ・エージェントまたは Connect:Direct ブリッジである場合はサポートされません。

-dq destination_queue

オプション。ファイルの転送先となる宛先キューの名前。この指定に *QUEUE@QUEUEMANAGER* というフォーマットでキュー・マネージャー名を含めることもできます。キュー・マネージャー名を指定しない場合は、*enableClusterQueueInputOutput* エージェント・プロパティを *true* に設定しない限り、宛先エージェントのキュー・マネージャーの名前が使用されます。

enableClusterQueueInputOutput エージェント・プロパティを *true* に設定した場合、宛先エージェントは、標準的な IBM MQ 解決手順を使用して、キューが配置されている場所を判別します。対象のキュー・マネージャーに存在する有効なキュー名を指定する必要があります。

-td、**-df**、**-dd**、**-ds**、**-dp**、**-du**、および **-dq** パラメーターのいずれかが必要です。**-dq** パラメーターを指定する場合、**-td**、**-dd**、**-df**、**-dp**、**-du**、または **-ds** パラメーターを指定できません。これらのパラメーターは相互に排他的だからです。

-dq パラメーターは、宛先エージェントがプロトコル・ブリッジ・エージェントまたは Connect:Direct ブリッジ・エージェントであるか、ソース指定がキューである場合はサポートされません。

-dq persistent

オプション。宛先キューに書き込むメッセージを永続メッセージにするかどうかを指定します。有効なオプションは以下のとおりです。

true

宛先キューに永続メッセージを書き込みます。これはデフォルト値です。

false

宛先キューに非永続メッセージを書き込みます。

qdef

永続性の値を宛先キューの *DefPersistence* 属性から取り込みます。

-dq パラメーターを指定できるのは、**-dq** パラメーターを指定した場合に限られます。

-qmp boolean

オプション。転送で宛先キューに書き込む最初のメッセージで IBM MQ メッセージ・プロパティを設定するかどうかを指定します。有効なオプションは以下のとおりです。

true

転送で作成する最初のメッセージでメッセージ・プロパティを設定します。

false

転送で作成する最初のメッセージでメッセージ・プロパティを設定しません。これはデフォルト値です。

-qmp パラメーターを指定できるのは、**-dq** パラメーターを指定した場合に限られます。詳しくは、[677 ページの『MFT が宛先キューに書き込むメッセージで設定する MQ メッセージ・プロパティ』](#)を参照してください

-qs message_size

オプション。ファイルを複数の固定長メッセージに分割するかどうかを指定します。メッセージは、すべて同じ IBM MQ グループ ID を持ちます。グループの最後のメッセージは IBM MQ `LAST_MSG_IN_GROUP` フラグ・セットを持ちます。メッセージのサイズは、`message_size` の値で指定します。`message_size` のフォーマットは、`lengthunits` です (`length` は正整数値、`units` は以下のいずれかの値です)。

B

バイト。指定できる最小値は、宛先メッセージのコード・ページの 1 文字あたりの最大バイト数の値の 2 倍です。

K

1024 バイトに相当します。

M

1048576 バイトに相当します。

-t パラメーターで値 `text` を指定した場合に、ファイルが 2 バイト文字セットまたはマルチバイト文字セットのファイルであれば、そのファイルは、指定のメッセージ・サイズに最も近い文字境界で、複数のメッセージに分割されます。

-qs パラメーターを指定できるのは、**-dq** パラメーターを指定した場合に限られます。**-qs**、**-dqdb**、および **-dqdt** パラメーターのいずれかのみ指定できます。

-dqdb hexadecimal_delimiter

オプション。バイナリー・ファイルを複数のメッセージに分割するときに使用する 16 進数区切り文字を指定します。メッセージは、すべて同じ IBM MQ グループ ID を持ちます。グループの最後のメッセージは IBM MQ `LAST_MSG_IN_GROUP` フラグ・セットを持ちます。区切り文字として 16 進数バイトを指定するためのフォーマットは、`xNN` です (`N` は、0-9 または a-f の範囲の文字です)。区切り文字として一連の 16 進数バイトを指定することもできます。その場合は、16 進数バイトのコンマ区切りリストを指定します (例: `x3e,x20,x20,xbf`)。

-dqdb パラメーターを指定できるのは、バイナリー・モードの転送で **-dq** パラメーターを指定した場合に限られます。**-qs**、**-dqdb**、および **-dqdt** パラメーターのいずれかのみ指定できます。

-dqdt pattern

オプション。テキスト・ファイルを複数のメッセージに分割するときに使用する正規表現を指定します。メッセージは、すべて同じ IBM MQ グループ ID を持ちます。グループの最後のメッセージは IBM MQ `LAST_MSG_IN_GROUP` フラグ・セットを持ちます。区切りとして正規表現を指定するためのフォーマットは、`(regular_expression)` (正規表現を括弧で囲んだ形式) です。このパラメーターの値は Java 正規表現として評価されます。詳しくは、[659 ページの『MFT が使用する正規表現』](#)を参照してください。

デフォルトでは、正規表現にマッチング可能なストリングの長さは、宛先エージェントによって 5 文字に制限されています。この動作を変更するには、`maxDelimiterMatchLength` エージェント・プロパティを使用します。詳しくは、[551 ページの『拡張エージェント・プロパティ』](#)を参照してください。

-dqdt パラメーターを指定できるのは、**-dq** パラメーターを指定し、**-t** パラメーターの値として `text` を指定した場合に限られます。**-qs**、**-dqdb**、および **-dqdt** パラメーターのいずれかのみ指定できます。

-dqdp

オプション。ファイルを分割するときの、宛先テキストおよびバイナリー区切り文字の想定される位置を指定します。**-dqdp** パラメーターを指定できるのは、**-dqdt** パラメーターと **-dqdb** パラメーターのいずれかを指定した場合に限られます。

以下のいずれかのオプションを指定します。

接頭部

区切り文字が各行の先頭にあることが想定されます。

postfix

区切り文字が各行の末尾にあることが想定されます。これはデフォルト・オプションです。

-qi

オプション。ファイルを複数のメッセージに分割するために使用する区切り文字をそれらのメッセージに組み込むかどうかを指定します。-qi を指定すると、区切り文字の前にあるファイル・データが含まれているメッセージの末尾に区切り文字が組み込まれます。デフォルトでは、メッセージに区切り文字は組み込まれません。

-qi パラメーターを指定できるのは、-dqdt パラメーターと -dqdb パラメーターのいずれかを指定した場合に限られます。

-de destination_file_behavior

オプション。宛先システムに宛先ファイルが存在する場合に取る処置を示します。有効なオプションは以下のとおりです。

エラー

エラーを報告し、ファイルは転送されません。これはデフォルト値です。

overwrite

既存の宛先ファイルを上書きします。

-de パラメーターを指定する場合、-td パラメーターを指定できません。これらのパラメーターは相互に排他的だからです。

-sd source_file_disposition

オプション。ソース・ファイルがその宛先に正常に転送されたときにソース・ファイルに対して取る処置を示します。有効なオプションは以下のとおりです。

leave

ソース・ファイルは変更されません。これはデフォルト値です。

delete

ソース・ファイルは、それが正常に転送された後にソース・システムから削除されます。

 z/OS では、ソースが磁気テープ・データ・セットの場合に、delete オプションを指定すると、磁気テープは再マウントされて、データ・セットが削除されます。この動作は、システム環境の動作のためにこのようになっています。

ソースがキューの場合に leave オプションを指定すると、このコマンドからエラーが返され、転送は要求されません。

ソース・エージェントが Connect:Direct ブリッジ・エージェントの場合に delete オプションを指定すると、動作は通常のソース後処理動作とは異なります。以下のいずれかになります。

- Connect:Direct が Managed File Transfer によって生成されるプロセスを使用してファイルまたはデータ・セットをソースから移動する場合、delete オプションを指定すると、転送が失敗します。ソース・ファイルの削除を指定するには、ユーザー定義の Connect:Direct プロセスを実行依頼してください。詳しくは、239 ページの『[ファイル転送要求からのユーザー定義 Connect:Direct プロセスの送信](#)』を参照してください。
- Connect:Direct がユーザー定義プロセスを使用して、ファイルまたはデータ・セットをソースから移動する場合は、%FTEFDISP 組み込みシンボリック変数によってこのパラメーターがプロセスに渡されます。ソースを削除するかどうかは、ユーザー定義プロセスによって決まります。転送から返される結果も、ユーザー定義プロセスから返される結果によって決まります。

-sd パラメーターを指定する場合、-td パラメーターを指定できません。これらのパラメーターは相互に排他的だからです。ただし、転送定義ファイルにソースの後処理の動作を指定することができます。

-pr transfer_priority

オプション。転送の優先順位を示します。priority は 0 から 9 までの範囲の値で、0 が最低の優先順位です。デフォルトの優先順位は 0 で、デフォルトではソース・エージェントの優先順位が転送で使用されます。

この値は、IBM MQ で使用されるメッセージ優先順位の値に一致します。詳細については、[キューからのメッセージの読み取り: 優先順位](#)を参照してください。ファイル転送データに対するメッセージ・トラフィックは、優先順位 0 にデフォルト設定されます。ここでは、ご使用の IBM MQ メッセージ・トラフィックを優先することができます。

V 9.0.1 -rt transfer_recovery_timeout

オプション。停止したファイル転送のリカバリーをソース・エージェントが試行し続ける時間 (秒単位) を設定します。以下のいずれかのオプションを指定します。

-1

エージェントは、停止した転送のリカバリーを、転送が完了するまで試行し続けます。このオプションを使用すると、このプロパティを設定しない場合のエージェントのデフォルトの動作と同じになります。

0

エージェントは、リカバリーに入るとすぐにファイル転送を停止します。

>0

エージェントは、指定された正整数値で設定された時間 (秒単位) だけ、停止した転送のリカバリーを試行し続けます。例:

```
-rt 21600
```

これは、エージェントがリカバリーに入ってから 6 時間にわたって転送のリカバリーを試行し続けることを示しています。このパラメーターの最大値は 999999999 です。

このように指定した場合、転送のリカバリー・タイムアウト値は転送単位で設定されます。Managed File Transfer ネットワーク内のすべての転送が対象になるグローバルな値を設定するには、プロパティを `agent.properties` ファイルに追加します。

-p configuration_options

オプション。このパラメーターは、転送テンプレートを作成するために使用される構成オプションのセットを決定します。このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティ・ファイルのセットを使用します。

このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。

-r

オプション。source_specification がワイルドカード文字を含んでいる場合に、サブディレクトリーのファイルを再帰的に転送します。Managed File Transfer が source_specification としてワイルドカード文字を使用して示されている場合、ワイルドカード文字に一致するディレクトリーは、-r パラメーターを指定している場合に限り、転送されます。source_specification がサブディレクトリーに一致する場合、そのディレクトリーとサブディレクトリー内のすべてのファイル (隠しファイルも含む) は常に転送されます。

Managed File Transfer がワイルドカード文字を処理する方法について、詳しくは『[ワイルドカード文字の使用](#)』を参照してください。

-r パラメーターを指定する場合、-td パラメーターを指定できません。これらのパラメーターは相互に排他的だからです。ただし、転送定義ファイルに再帰的動作を指定することができます。

-t

オプション。ファイル転送のタイプ (バイナリー・モードまたはテキスト・モード) を指定します。

binary

ファイル内のデータは変換されずに転送されます。これはデフォルト値です。

text

ファイルのコード・ページおよび行末文字は変換されます。実際に実行される変換は、ソース・エージェントおよび宛先エージェントのオペレーティング・システムによって異なります。

z/OS 例えば、Windows から z/OS に転送されるファイルの場合、そのコード・ページは ASCII から EBCDIC に変換されます。ファイルを ASCII から EBCDIC に変換すると、行末文字は、ASCII 復帰 (CR) および改行 (LF) 文字のペアから EBCDIC 改行 (NL) 文字に変換されます。

z/OS z/OS データ・セットの転送方法について詳しくは、『z/OS と分散システム間のファイルおよびデータ・セットの転送』および『データ・セット間の転送』を参照してください。

-t パラメーターを指定する場合、**-td** パラメーターを指定できません。これらのパラメーターは相互に排他的だからです。ただし、転送定義ファイルに転送モードの動作を指定することができます。

-cs

オプション。転送されたファイルの整合性を検査するためにファイル転送データに対してチェックサム・アルゴリズムが実行されるかどうかを指定します。以下のいずれかのオプションを指定します。

MD5

データの MD5 チェックサムを計算します。ソース・ファイルと宛先ファイルの結果チェックサムが検証のために転送ログに書き込まれます。デフォルトでは、Managed File Transfer は、すべてのファイル転送において MD5 チェックサムを計算します。

なし

ファイル転送データの MD5 チェックサムは計算されません。転送ログにはチェックサムが `none` に設定されたことが記録され、チェックサムの値はブランクになります。以下に例を示します。

```
<checksum method="none"></checksum>
```

`none` オプションを使用すると、ご使用の環境によってはファイル転送のパフォーマンスが向上する場合があります。ただし、このオプションを選択した場合、ソース・ファイルまたは宛先ファイルの検証はありません。

-cs パラメーターを指定する場合、**-td** パラメーターを指定できません。これらのパラメーターは相互に排他的だからです。ただし、転送定義ファイルにチェックサムの動作を指定することができます。

-tr

オプション。このファイル転送が実行されるために満たす必要がある条件を指定します。ソース・エージェントにおいて条件が満たされなかった場合、ファイル転送は廃棄され、転送は行われません。次のフォーマットで指定します。

```
condition, namelist
```

ここで `condition` は以下のいずれかの値になります。

file=exist

名前リストの少なくとも 1 つのファイルが存在します。つまり、名前リストのいずれかのファイルが存在する場合、この条件は TRUE となります。

file!=exist

名前リストの少なくとも 1 つのファイルが存在しません。つまり、名前リストのいずれかのファイルが存在しない場合、この条件は TRUE となります。

filesize>=size

名前リストの少なくとも 1 つのファイルが存在し、`size` で指定した最小サイズ以上です。`size` の値は、オプションで KB、MB、または GB のサイズ単位が付加される整数です。例えば、`filesize">"=10KB` のように指定します。サイズの単位が指定されなかった場合は、バイト単位が想定されます。すべてのオペレーティング・システムにおいて、この例で示すように、コマンド行で `filesize` オプションを指定した場合は、より大記号 (>) は二重引用符で囲まなければならない。

さらに、*namelist* はソース・システムにあるファイル名のコンマ区切りリストです。ご使用のオペレーティング・システムによっては、スペースが含まれる名前リストでパス名またはファイル名を使用する場合、パス名とファイル名を二重引用符で囲む必要があることがあります。

トリガー条件は、**-tr** パラメーターを複数回使用することにより複数指定できます。ただしその場合、ファイル転送が実行されるには、それぞれすべてのトリガー条件が満たされることが必要です。

注: リソースを継続的にモニターしてトリガー条件が真であるかどうかを判断する場合には、リソース・モニターを使用することをお勧めします。`fteCreateMonitor` コマンドを使用して、リソース・モニターを作成できます。

以下の例では、ファイル `file1.doc` が以下の条件で AGENT1 から AGENT2 へ転送されます。ファイル `A.txt` またはファイル `B.txt` のいずれか、あるいは両方のファイルが AGENT1 に存在し、かつファイル `A.txt` またはファイル `B.txt` のいずれか、または両方のファイルが 1 GB 以上の場合。

```
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
```

トリガー・パラメーターとスケジューリング・パラメーターは結合できます。両方のタイプのパラメーターを指定した場合、スケジューリング・パラメーターにより作成されたファイル転送にトリガー条件が適用されます。

-tl

オプション。トリガー障害がログに記録されるかどうかを指定します。以下のいずれかのオプションを指定します。

yes

失敗したトリガー転送についてのログ項目が作成されます。これは、**-tl** パラメーターを指定しなかった場合でも、デフォルトの動作になります。

いいえ

失敗したトリガー転送についてのログ項目が作成されません。

-md

オプション。エージェントの出口点に渡されるユーザー定義のメタデータを示します。**-md** パラメーターには、名前/値ペアをコンマ区切りで 1 つ以上含められます。それぞれの名前ペアの構成は、`name=value` になります。コマンド内では **-md** パラメーターを複数回使用できます。

 **z/OS** z/OS では、スペースは区切り文字を表すため、値を区切るには下線を使用する必要があります。例えば、`kw="text1 text2 text3"` ではなく、`kw=text1_text2_text3` を使用してください。

-tb

オプション。スケジュール済みファイル転送で使用する時間基準を示します。つまり、システム時刻を使用するか、協定世界時 (UTC) を使用するかを示します。このパラメーターは、**-ss** パラメーター以外と一緒に使用できません。以下のいずれかのオプションを指定します。

admin

スケジュール済み転送で使用する開始および終了時刻は、管理者が使用するシステムの日時を基準とします。これはデフォルト値です。

ソース

スケジュール済み転送で使用する開始および終了時刻は、ソース・エージェントがあるシステムの日時を基準とします。

UTC(C)

スケジュール済み転送で使用する開始および終了時刻は、協定世界時 (UTC) を基準とします。

-jn job_name

オプション。転送が開始されたときに、ログ・メッセージに追加されるユーザー定義ジョブ名 ID。

-ss schedule_start_time

オプション。スケジュール済み転送を実行する日時を示します。日時の指定には、次のいずれかの形式を使用します。24時間クロックを使用して、時刻を指定します。

```
yyyy-MM-ddThh:mm
```

```
hh:mm
```

スケジュール済みファイル転送は、転送に影響する可能性がある問題がなければ、スケジュール開始時刻から1分以内に開始します。例えば、スケジュール済み転送の開始を妨げるネットワークまたはエージェントの問題があるかもしれません。

-oi

オプション。スケジュール済み転送が発生する間隔を示します。このパラメーターは、**-ss** パラメーター以外と一緒に使用できません。以下のいずれかのオプションを指定します。

minutes

hours

日

weeks

months

years

-of occurrence_frequency

オプション。スケジュール済み転送が発生する頻度を示します。例えば、**5**週ごと、**2**カ月ごとなどです。このパラメーターは、**-oi** および **-ss** パラメーター以外と一緒に指定できません。このパラメーターを指定しない場合、デフォルト値の**1**が使用されます。

-oc occurrence_count

オプション。このスケジュール済み転送が発生する回数を示します。この発生数になると、スケジュール済み転送は削除されます。

このパラメーターは、**-oi** および **-ss** パラメーター以外と一緒に指定しないでください。

-oc パラメーターを指定する場合、**-es** パラメーターを指定できません。これらのパラメーターは相互に排他的だからです。

無期限に繰り返す転送を作成するには、**-oc** と **-es** の両方のパラメーターを省略します。

-es schedule_end_time

オプション。スケジュール済み繰り返し転送が終了する日時。

このパラメーターは、**-oi** および **-ss** パラメーター以外と一緒に指定できません。

-es パラメーターを指定する場合、**-oc** パラメーターを指定できません。これらのパラメーターは相互に排他的だからです。

無期限に繰り返す転送を作成するには、**-es** と **-oc** の両方のパラメーターを省略します。

終了日時の指定には、次のいずれかの形式を使用します。24時間クロックを使用して、時刻を指定します。

```
yyyy-MM-ddThh:mm
```

```
hh:mm
```

-tn template_name

必須。作成するテンプレートの名前。後日、転送用の正しいテンプレートを選択できるようにするための記述ストリングを使用します。このストリングの長さに固有の制限はありませんが、名前が長す

ぎると、一部のユーザー・インターフェースでは正しく表示されない場合があることを覚えておいてください。

複数のテンプレートを同じ名前で作成しないでください。

-sqgi

オプション。メッセージを IBM MQ グループ ID によってグループ化する動作を指定します。完全に揃った最初のグループが宛先ファイルに書き込まれます。このパラメーターを指定しない場合は、ソース・キューに存在するすべてのメッセージが宛先ファイルに書き込まれます。

-sqgi パラメーターを指定できるのは、**-sq** パラメーターを指定した場合に限られます。

-sqdt text_delimiter

オプション。1つのテキスト・ファイルに複数のメッセージを追加するときに区切り文字として挿入するテキストのシーケンスを指定します。ストリング・リテラルの Java エスケープ・シーケンスを区切り文字に含めることもできます。例えば、**-sqdt \u007d\n** です。

-sqdt パラメーターを指定できるのは、**-sq** パラメーターを指定し、**-t** パラメーターの値として **text** を指定した場合に限られます。

-sqdb hexadecimal_delimiter

オプション。1つのバイナリー・ファイルに複数のメッセージを追加するときに区切り文字として挿入する1つ以上のバイト値を指定します。それぞれの値は、00-FF の範囲にある2桁の16進数に接頭部 **x** を付けた形式で指定する必要があります。複数バイトの場合はコンマで区切る必要があります。例えば、**-sqdb x08,xA4** のように指定します。

-sqdb パラメーターを指定できるのは、**-sq** パラメーターを指定した場合に限られます。**-t** パラメーターの値として **text** を指定した場合は、**-sqdb** パラメーターを指定できません。

-sqdp

オプション。ソース・テキストおよびバイナリー区切り文字の挿入位置を指定します。**-sqdp** パラメーターを指定できるのは、**-sqdt** パラメーターと **-sqdb** パラメーターのいずれかを指定した場合に限られます。

以下のいずれかのオプションを指定します。

接頭部

区切り文字は各メッセージの先頭に挿入されます。

postfix

区切り文字は各メッセージの末尾に挿入されます。これはデフォルト・オプションです。

-sqwt wait_time

オプション。以下のいずれかの条件が満たされるのを待つ時間を秒単位で指定します。

- 新規メッセージがキューに置かれる
- **-sqgi** パラメーターが指定された場合、グループ全体がキューに置かれる

wait_time で指定した時間内にどちらの条件も満たされなければ、ソース・エージェントは、キューからの読み取りを停止して、転送を完了します。**-sqwt** パラメーターを指定しなければ、ソース・エージェントは、ソース・キューが空の場合にソース・キューからの読み取りをただちに停止します。あるいは、**-sqgi** パラメーターが指定されているのであれば、完全に揃ったグループがキューに存在しない場合にソース・キューからの読み取りをただちに停止します。

-sqwt パラメーターを指定できるのは、**-sq** パラメーターを指定した場合に限られます。

-sq

オプション。転送のソースがキューであることを指定します。

-mquserid (userID)

オプション。調整キュー・マネージャーで認証するユーザー ID を指定します。

-mqpassword (*password*)

オプション。調整キュー・マネージャーで認証するパスワードを指定します。 **-mquserid** パラメーターも一緒に指定する必要があります。 **-mquserid** を指定した場合に **-mqpassword** を指定しないと、関連付けるパスワードを入力するというプロンプトが出されます。パスワードは表示されません。

source_specification

-df、**-dd**、**-dp**、**-dp**、または **-ds** パラメーターのいずれかを指定する場合に必要です。 **-td** パラメーターを指定した場合は、*source_specification* を指定しないでください。

- **-sq** パラメーターを指定しない場合は、*source_specification* として、ファイル転送のソースを示すファイル指定を1つ以上記述します。ファイル指定は、スペースで区切ります。ファイルの指定では5つの形式のうちの一つを使用することができ、ワイルドカード文字を含めることができます。WMQFTEのワイルドカード文字について詳しくは、653ページの『MFTでのワイルドカード文字の使用』を参照してください。ファイル指定でアスタリスク文字を2つ(**)使用することにより、ファイル指定の一部を成すアスタリスクをエスケープすることができます。

ファイル名にスペースが含まれるファイルを転送する場合は、スペースが含まれるファイル名を二重引用符文字で囲みます。例えば、ファイル `a b.txt` をファイル `c d.txt` に転送するには、**fteCreateTemplate** コマンドの一部として以下のテキストを指定します。

```
-df "c d.txt" "a b.txt"
```

各ファイル指定は、以下のいずれかのフォーマットでなければなりません。

ファイル名

ソース・エージェントが実行されているシステムに適切な表記を使用して表現される、ファイルの名前。ファイル名をソース・ファイルとして指定すると、ファイルの内容がコピーされます。

ディレクトリー

ソース・エージェントが実行されているシステムに適切な表記を使用して表現される、ディレクトリーの名前。ディレクトリーをソース・ファイルとして指定すると、ディレクトリーの内容がコピーされます。もっと正確に表現すると、ディレクトリーとそのすべてのサブディレクトリー内のすべてのファイル(隠しファイルを含む)がコピーされます。

例えば、DIR1の内容をDIR2のみにコピーするには、`DIR1/* DIR2` と指定します。

順次データ・セット

(z/OSのみ)。順次データ・セットまたは区分データ・セット・メンバーの名前。データ・セットは、名前の先頭に2つのスラッシュ文字(//)を付けて表します。

区分データ・セット

(z/OSのみ)。区分データ・セットの名前。データ・セット名は、先頭に2つのスラッシュ文字(//)を付けて表します。

Connect:Direct ノードのファイル名またはディレクトリー

(Connect:Directブリッジ・エージェントのみ)。Connect:Directノードの名前、コロン文字(:)、およびConnect:Directノードをホストしているシステム上のファイル・パスまたはディレクトリー・パス。例えば、`connect_direct_node_name:file_path` のように指定します。

ソース・エージェントがConnect:Directブリッジ・エージェントの場合、この形式のソース指定のみが受け入れられます。

注: ソース・エージェントがConnect:Directブリッジ・エージェントである場合は、ファイル・パス内のワイルドカード文字はサポートされません。

- **-sq** パラメーターを指定した場合は、*source_specification* として、ソース・エージェントのキュー・マネージャーに存在するローカル・キューの名前を指定します。指定できるソース・キューは、1つだけです。ソース・キューを指定するためのフォーマットは、以下のとおりです。

```
QUEUE_NAME
```

ソース・キューの指定には、キュー・マネージャー名を含めません。そのキュー・マネージャーは、ソース・エージェントのキュー・マネージャーと同じでなければなりません。

-?または-h

オプション。コマンド構文を表示します。

例

この例では、payroll accounts monthly report template という転送テンプレートが作成されます。このテンプレートが送信されると、拡張子.xls が付いたすべてのファイルが、指定されたディレクトリーのエージェント PAYROLL1 からエージェント ACCOUNTS に転送されます。

```
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
```

この例では、jupiter_neptune_sched_template という転送テンプレートが作成されます。実行依頼されると、テンプレートはファイル originalfile.txt を QM_JUPITER があるシステムから QM_NEPTUNE があるシステムに転送します。ファイル転送は、ソース・エージェントがあるシステムのシステム時刻を基準として 09:00 に実行されるようにスケジュールされます。これは 2 時間ごとに 4 回実行されます。

```
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
```

この例では、jupiter neptune trigger template という転送テンプレートが作成されます。テンプレートが送信されると、ファイル originalfile.txt は、ファイル A.txt が AGENT1 に存在するという条件で、AGENT1 から AGENT2 に転送されます。

```
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 この例では、ascii_ebcdic_template というテンプレートが作成されます。テンプレートが実行依頼されると、ファイル originalfile.txt が、AGENT1 があるシステムから AGENT2 があるシステム上のデータ・セット //'USERID.TRANS.FILE.TXT' に転送されます。ASCII から EBCDIC へのデータ変換にテキスト・モードが選択されています。

```
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 この例では、ebcdic_ascii_template というテンプレートが作成されます。テンプレートが実行依頼されると、AGENT1 があるシステム上の完全修飾データ・セットのメンバーが、AGENT2 があるシステム上のファイルに転送されます。EBCDIC から ASCII にファイルを変換するためにテキスト・モードが選択されています。

```
fteCreateTemplate -tn ebcdic_ascii_template -t text -sa AGENT1 -da AGENT2 -df /tmp/IEEUJV.txt ";//'SYS1.SAMPLIB(IEEUJV)'"
```

戻りコード

戻りコード	説明
0	コマンドは正常に完了しました。

戻りコード	説明
1	コマンドは失敗しました。

fteCreateTransfer: 新規ファイル転送の開始

fteCreateTransfer コマンドは、コマンド行から新規のファイル転送を作成して開始します。このコマンドは、ファイル転送の即時開始、将来の日時におけるファイル転送のスケジュール、ファイル転送の1回以上の繰り返し、および特定の条件に基づくファイル転送の起動を実行できます。

目的

fteCreateTransfer コマンドは、Managed File Transfer エージェントから新規ファイル転送を作成して開始する場合に使用します。

注: ファイル転送は、同じ Managed File Transfer トポロジー内のエージェント間でのみ実行できます。

ファイルを転送する方法については、630 ページの『[ファイルの転送に関するガイドライン](#)』を参照してください。z/OS プラットフォームの場合は、テキスト・ファイル、データ・セット、および世代別データ・グループ (GDG) を転送できます。

fteCreateTransfer コマンドは、IBM MQ ネットワークに接続してからソース・エージェントのキュー・マネージャーに経路指定のできる、任意のシステムから実行できます。具体的には、このコマンドを実行する場合、このシステムに Managed File Transfer コンポーネント (Service またはエージェント) をインストールし、IBM MQ ネットワークと通信するようにこのシステム上の Managed File Transfer コンポーネントを構成する必要があります。

このコマンドは、`command.properties` というプロパティ・ファイルを使用して、IBM MQ ネットワークに接続します。`command.properties` ファイルにプロパティ情報が含まれていない場合、ローカル・システムのデフォルト・キュー・マネージャーに対してバインディング・モードで接続されます。`command.properties` ファイルが存在しない場合、エラーが生成されます。詳しくは、546 ページの『[MFT command.properties ファイル](#)』を参照してください。

1つのファイル転送で複数のソース・ファイルを指定することができますが、それらのソース・ファイルの起点は1つのソース・エージェントでなければならず、終点も1つの宛先エージェントでなければなりません。1つの転送で、1つのソース・ファイルを同じエージェントまたは複数の異なるエージェントに存在する複数の宛先ファイルに転送する操作は、サポートされていません。1つ以上のエージェントに存在する複数の宛先に同じソース・ファイルを送信する場合は、Ant スクリプトを使用できます。詳しくは、252 ページの『[MFTでのApache Antの使用](#)』を参照してください。

特殊文字

特殊文字が含まれているパラメーターを使用する場合には、予期しない形で特殊文字がコマンド・シェルで解釈されないように注意してください。例えば、単一引用符を含む完全修飾データ・セット名およびアスタリスク文字を含むソース指定は、転送要求で渡されずに、コマンド・シェルで解釈されてしまう可能性があります。文字がコマンド・シェルによって解釈される動作を回避するには、パラメーター全体を二重引用符で囲むか、またはコマンド・シェルのエスケープ・シーケンスを使用して特殊文字をエスケープします。

相対パス

fteCreateTransfer コマンドは、相対ファイル・パスの使用をサポートします。次のプラットフォームの場合、デフォルトでは、エージェントを実行しているユーザーのホーム・ディレクトリーからの相対パスとみなされます。

- ▶ **Multi** Multiplatforms
- ▶ **z/OS** z/OS 上の UNIX システム・サービス

パス名が相対パスと評価されるディレクトリーを変更するには、`agent.properties` ファイルに `transferRoot` プロパティーを設定します。このファイルは `MQ_DATA_PATH/mqft/config/coordination_qmgr/agents/agent_name` ディレクトリーにあります。次の行をファイルに追加します。

```
transferRoot=directory_name
```

Windows 例えば、`C:\TransferRoot` を `C:\\TransferRoot` または `C:/TransferRoot` と指定します。

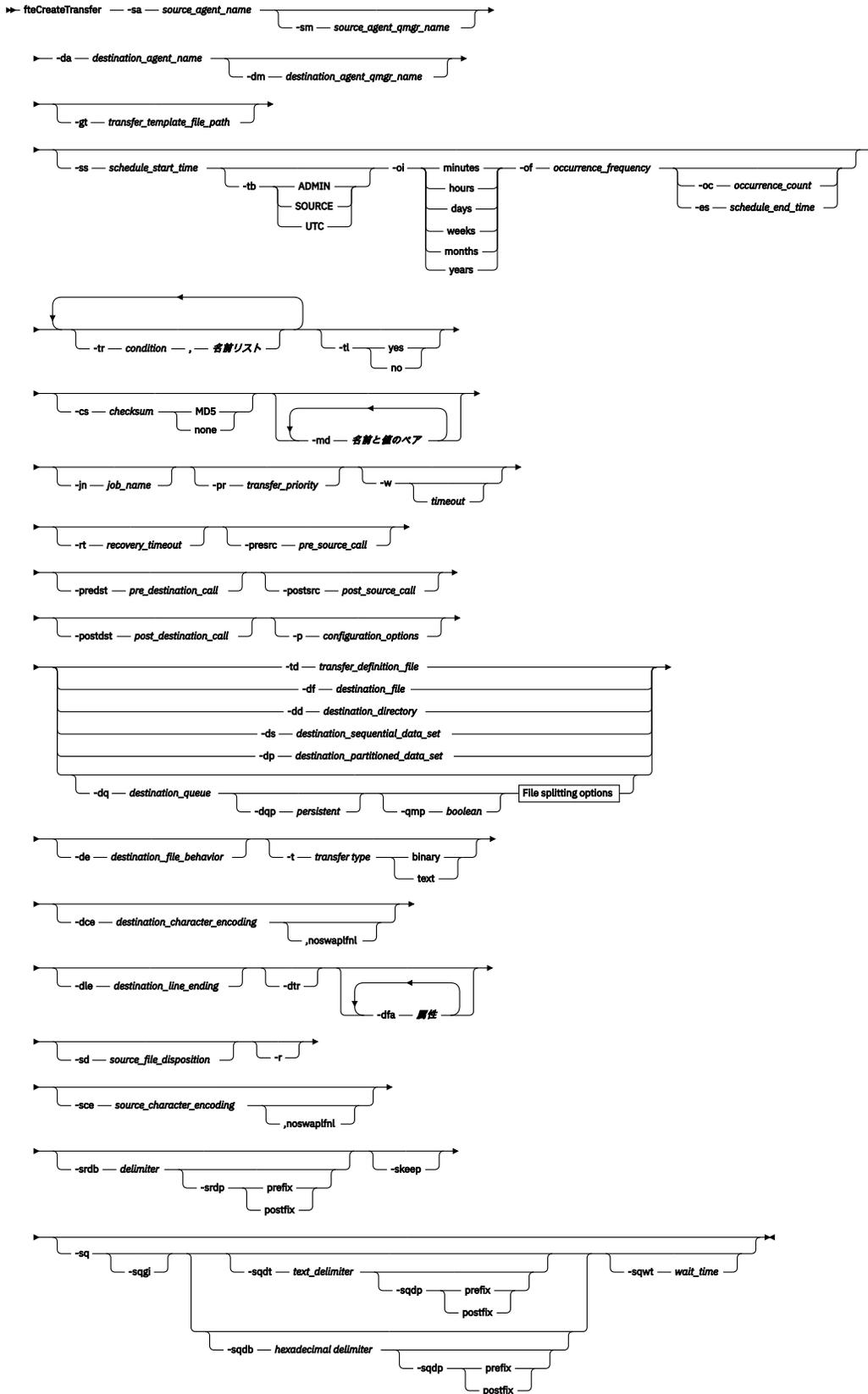
z/OS z/OS では、デフォルトでは、エージェントが動作しているユーザー名が、高位修飾子接頭部として完全に修飾されていないデータ・セット仕様に追加されます。例えば、`//ABC.DEF` のようになります。接頭部としてデータ・セット名に追加される値を変更するには、`transferRootHLQ` プロパティーを `agent.properties` ファイルで設定します。このファイルは `MQ_DATA_PATH/mqft/config/coordination_qmgr/agents/agent_name` ディレクトリーにあります。次の行をファイルに追加します。

```
transferRootHLQ=prepend_value
```

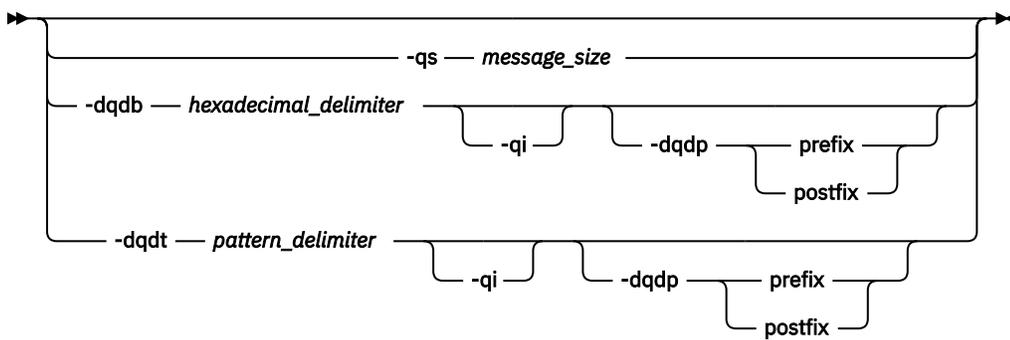
z/OS ただし、z/OS システム上の `Connect:Direct` ノードに関する転送では、データ・セット仕様は完全修飾名として解釈されます。データ・セット名に高位修飾子が追加されることはありません。

構文

fteCreateTransfer



File splitting options



Parameters for MQ security



➤ source_specification ➤

エージェント指定のパラメーター

-sa source_agent_name

必須。ソース・ファイルの転送元のエージェントの名前。

Z/OS プロトコル・ブリッジ・エージェントをソース・エージェントとして指定する場合は、データ・セットをソース・ファイルとして指定することはできません。

-td パラメーターを指定し、転送に使用するソース・エージェントが転送定義ファイルに含まれている場合には、**-sa** パラメーターは指定しないでください。

-sm source_agent_qmgr_name

オプション。ソース・エージェントの接続先のキュー・マネージャーの名前。

-sm パラメーターを指定しない場合、どのキュー・マネージャーを使用するかは、使用している構成オプションのセットによって決定されます (これは、ソース・エージェントの名前に基づきます)。ソース・エージェントの agent.properties ファイルが見つからなかった場合、ファイル転送は失敗します。

-da destination_agent_name

必須。ソース・ファイルの転送先のエージェントの名前。

-td パラメーターを指定し、転送に使用する宛先エージェントが転送定義ファイルに含まれている場合には、**-da** パラメーターは指定しないでください。

-dm destination_agent_qmgr_name

オプション。宛先エージェントの接続先のキュー・マネージャーの名前。

-dm パラメーターを指定しない場合、どのキュー・マネージャーを使用するかは、使用している構成オプションのセットによって決定されます (これは、宛先エージェントの名前に基づきます)。宛先エージェントの agent.properties ファイルが見つからなかった場合、ファイル転送は失敗します。

転送テンプレートの生成に関するパラメーター

-gt transfer_template_file_path

オプション。転送テンプレート XML メッセージを生成して、このメッセージをファイルに書き込みます。このパラメーターを指定すると、転送要求は Managed File Transfer に送信されません。その代わりに、転送要求メッセージの内容が指定された XML 文書に書き込まれます。その後、この XML 文書を使用して、リソースをモニターするタスクを定義できます。リソース・モニターの作成方法について

は、`fteCreateMonitor` コマンドを参照してください。このパラメーターを指定しない場合、デフォルトの動作が行われ、実際に転送要求が実行されます。

このパラメーターの入力として、XML 出力ファイルの絶対パスおよび名前を指定する必要があります。例えば、`C:\templates\transfer_reports.xml` のように指定します。

z/OS z/OS では、z/OS UNIX システム・サービスの UNIX ファイルに、転送テンプレート文書を保管する必要があります。z/OS の順次ファイルまたは PDS メンバーに、転送テンプレート文書を保管することはできません。

IBM i IBM i では、統合ファイル・システムに転送テンプレート文書を保管する必要があります。

-gt パラメーターを使用して作成した転送テンプレート XML メッセージは、**fteCreateTemplate** コマンドを使用して作成した転送と同じではありません。つまり、この 2 つの異なるタイプのテンプレートを交換して使用することはできません。

注 **fteCreateTransfer** コマンドに **-gt** パラメーターを指定して実行することによって転送テンプレート XML 文書を生成してから、**fteCreateTransfer** コマンドの入力として **-td** パラメーターを使用してその転送テンプレート XML 文書を提供する場合は、**-td** オプションと同時に使用できないパラメーターを指定して転送テンプレート XML 文書を生成しておく必要があります。

-td オプションと同時に使用できないパラメーターは、以下のとおりです。

- **-dd** *destination_directory*
- ソース・パス
- **-df** *destination_file*
- **-cs** チェックサム
- **-de** *destination_file_behavior*
- **-dq** *destination_queue*
- **-ttransfer** *type*
- **-sd** *source_file_disposition*

例えば、**fteCreateTransfer** コマンドで、**-td** パラメーターと **-t** パラメーター (バイナリー転送とテキスト転送のどちらであるかを示す) の両方を指定することはできません。つまり、転送テンプレート XML 文書をコマンドに渡し、転送をテキスト転送にすることを指定する場合は、**-gt** パラメーターと **-t** テキスト・パラメーターを指定して XML 文書を作成する必要があります。

転送のスケジューリングに関するパラメーター

-ss *schedule_start_time*

オプション。スケジュール済み転送を実行する日時を示します。日時の指定には、次のいずれかの形式を使用します。24 時間クロックを使用して、時刻を指定します。

```
yyyy-MM-ddThh:mm  
hh:mm
```

スケジュール済みファイル転送は、転送に影響する可能性がある問題がなければ、スケジュール開始時刻から 1 分以内に開始します。例えば、スケジュール済み転送の開始を妨げるネットワークまたはエージェントの問題があるかもしれません。

-tb

オプション。スケジュール済みファイル転送で使用する時間基準を示します。つまり、システム時刻を使用するか、協定世界時 (UTC) を使用するかを示します。このパラメーターは、**-ss** パラメーター以外と一緒に使用できません。以下のいずれかのオプションを指定します。

admin

スケジュール済み転送で使用する開始および終了時刻は、ローカル管理者が使用するシステムの日時を基準とします。これはデフォルト値です。

ソース

スケジュール済み転送で使用する開始および終了時刻は、ソース・エージェントがあるシステムの日時を基準とします。

UTC(C)

スケジュール済み転送で使用する開始および終了時刻は、協定世界時 (UTC) を基準とします。

-oi

オプション。スケジュール済み転送が発生する間隔を示します。このパラメーターは、**-ss** パラメーター以外と一緒に使用できません。以下のいずれかのオプションを指定します。

minutes

hours

日

weeks

months

years

-of occurrence_frequency

オプション。スケジュール済み転送が発生する頻度を示します。例えば、**5** 週ごと、**2** カ月ごとなどです。このパラメーターは、**-oi** および **-ss** パラメーター以外と一緒に指定できません。このパラメーターを指定しない場合、デフォルト値の **1** が使用されます。

-oc occurrence_count

オプション。このスケジュール済み転送が発生する回数を示します。この発生数になると、スケジュール済み転送は削除されます。

このパラメーターは、**-oi** および **-ss** パラメーター以外と一緒に指定しないでください。

-oc パラメーターを指定する場合、**-es** パラメーターを指定できません。これらのパラメーターは相互に排他的だからです。

無期限に繰り返す転送を作成するには、**-oc** と **-es** の両方のパラメーターを省略します。

-es schedule_end_time

オプション。スケジュール済み繰り返し転送が終了する日時。

このパラメーターは、**-oi** および **-ss** パラメーター以外と一緒に指定できません。

-es パラメーターを指定する場合、**-oc** パラメーターを指定できません。これらのパラメーターは相互に排他的だからです。

無期限に繰り返す転送を作成するには、**-es** と **-oc** の両方のパラメーターを省略します。

終了日時の指定には、次のいずれかの形式を使用します。24 時間クロックを使用して、時刻を指定します。

```
yyyy-MM-ddThh:mm
```

```
hh:mm
```

転送のトリガーに関するパラメーター

-tr

オプション。このファイル転送が実行されるために満たす必要がある条件を指定します。ソース・エージェントにおいて条件が満たされなかった場合、ファイル転送は廃棄され、転送は行われません。次のフォーマットで指定します。

```
condition,namelist
```

ここで *condition* は以下のいずれかの値になります。

file=exist

名前リストの少なくとも1つのファイルが存在します。つまり、名前リストのいずれかのファイルが存在する場合、この条件は TRUE となります。

file!=exist

名前リストの少なくとも1つのファイルが存在しません。つまり、名前リストのいずれかのファイルが存在しない場合、この条件は TRUE となります。

filesize>=size

名前リストの少なくとも1つのファイルが存在し、*size* で指定した最小サイズ以上です。*size* は、オプションで KB、MB、または GB のサイズ単位がある整数です。例えば、`filesize">"=10KB` のように指定します。サイズの単位が指定されなかった場合は、バイト単位が想定されます。すべてのオペレーティング・システムにおいて、この例で示すように、コマンド行で `filesize` オプションを指定した場合は、より大記号 (>) は二重引用符で囲まなければなりません。

さらに、*namelist* はソース・エージェントと同じシステムにあるファイル名のコンマ区切りリストです。ご使用のオペレーティング・システムによっては、スペースが含まれる名前リストでパス名またはファイル名を使用する場合、パス名とファイル名を二重引用符で囲む必要があることがあります。

トリガー条件は、**-tr** パラメーターを複数回使用することにより複数指定できます。ただしその場合、ファイル転送が実行されるには、それぞれすべてのトリガー条件が満たされることが必要です。

注: リソースを継続的にモニターしてトリガー条件が真であるかどうかを判断する場合には、[リソース・モニター](#)を使用することを強くお勧めします。`fteCreateMonitor` コマンドを使用して、リソース・モニターを作成できます。

以下の例では、ファイル `file1.doc` が以下の条件で AGENT1 から AGENT2 へ転送されます。ファイル `A.txt` またはファイル `B.txt` のいずれか、あるいは両方のファイルが AGENT1 に存在し、かつファイル `A.txt` またはファイル `B.txt` のいずれか、または両方のファイルが 1 GB 以上の場合。

```
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
```

トリガー・パラメーターとスケジューリング・パラメーターは結合できます。両方のタイプのパラメーターを指定した場合、スケジューリング・パラメーターにより作成されたファイル転送にトリガー条件が適用されます。

-tr パラメーターは、プロトコル・ブリッジ・エージェントではサポートされません。

-tl

オプション。トリガー障害を転送ログに書き込むかどうかを指定します。以下のいずれかのオプションを指定します。

yes

失敗したトリガー転送についての転送ログ項目が作成されます。これは、**-tl** パラメーターを指定しなかった場合でも、デフォルトの動作になります。

いいえ

失敗したトリガー転送についての転送ログ項目は作成されません。

転送オプションの指定に関するパラメーター

-jn *job_name*

オプション。転送が開始されたときに、転送ログ・メッセージに追加されるユーザー定義ジョブ名 ID。

-md

オプション。エージェントにより実行される出口点に渡される、ユーザー定義のメタデータを指定します。 **-md** パラメーターには、名前/値ペアをコンマ区切りで 1 つ以上含められます。それぞれの名前ペアの構成は、*name=value* になります。コマンド内では **-md** パラメーターを複数回使用できます。

エージェント・プロパティ `enableUserMetadataOptions` の値が `true` に設定されている場合には、特定のユーザー定義のメタデータ・キーによって、転送に対する追加オプションが指定されます。現在サポートされているユーザー定義のメタデータ・キーについては、572 ページの『[enableUserMetadataOptions: サポートされている MFT ユーザー定義メタデータ・キー](#)』を参照してください。 `enableUserMetadataOptions` プロパティが `true` に設定されている場合には、ユーザー定義の使用について `com.ibm.wmqfte.` で始まるキー名はサポートされません。

-cs *checksum*

オプション。転送されたファイルの整合性を検査するためにファイル転送データに対してチェックサム・アルゴリズムが実行されるかどうかを指定します。以下のいずれかのオプションを指定します。

MD5

データの MD5 チェックサムを計算します。ソース・ファイルと宛先ファイルの結果チェックサムが検証のために転送ログに書き込まれます。デフォルトでは、Managed File Transfer は、すべてのファイル転送において MD5 チェックサムを計算します。

なし

ファイル転送データの MD5 チェックサムは計算されません。転送ログにはチェックサムが `none` に設定されたことが記録され、チェックサムの値は空白になります。以下に例を示します。

```
<checksum method="none"></checksum>
```

`none` オプションを使用すると、ご使用の環境によってはファイル転送のパフォーマンスが向上する場合があります。ただし、このオプションを選択した場合、ソース・ファイルまたは宛先ファイルの検証はありません。

-cs パラメーターを指定する場合、**-td** パラメーターを指定できません。これらのパラメーターは相互に排他的だからです。ただし、転送定義ファイルにチェックサムの動作を指定することができます。

-pr *transfer_priority*

オプション。転送の優先順位を示します。 `priority` は 0 から 9 までの範囲の値で、0 が最低の優先順位です。デフォルトの優先順位は、ソース・エージェントの優先順位です。

この値は、IBM MQ のメッセージ優先順位の値に一致します。詳細については、[キューからのメッセージの読み取り: 優先順位](#)を参照してください。ファイル転送データに対するメッセージ・トラフィックは、優先順位 0 にデフォルト設定されます。ここでは、ご使用の IBM MQ メッセージ・トラフィックを優先することができます。

-qmp *boolean*

オプション。転送で宛先キューに書き込む最初のメッセージで IBM MQ メッセージ・プロパティを設定するかどうかを指定します。有効なオプションは以下のとおりです。

true

転送で作成する最初のメッセージでメッセージ・プロパティを設定します。

false

転送で作成する最初のメッセージでメッセージ・プロパティを設定しません。これはデフォルト値です。

-qmp パラメーターを指定できるのは、**-dq** パラメーターを指定した場合に限られます。詳しくは、677 ページの『MFT が宛先キューに書き込むメッセージで設定する MQ メッセージ・プロパティ』を参照してください

-qs message_size

オプション。ファイルを複数の固定長メッセージに分割するかどうかを指定します。メッセージは、すべて同じ IBM MQ グループ ID を持ちます。グループの最後のメッセージは IBM MQ `LAST_MSG_IN_GROUP` フラグ・セットを持ちます。メッセージのサイズは、`message_size` の値で指定します。`message_size` のフォーマットは、`lengthunits` です (`length` は正整数値、`units` は以下のいずれかの値です)。

B

バイト。指定できる最小値は、宛先メッセージのコード・ページの 1 文字あたりの最大バイト数の値の 2 倍です。

K

1024 バイトに相当します。

M

1048576 バイトに相当します。

ファイルをテキスト・モードで転送していて、ファイルが 2 バイト文字セットまたはマルチバイト文字セットのファイルである場合、そのファイルは、指定のメッセージ・サイズに最も近い文字境界で複数のメッセージに分割されます。

-qs パラメーターを指定できるのは、**-dq** パラメーターを指定した場合に限られます。**-qs**、**-dqdb**、および **-dqdt** パラメーターのいずれかのみ指定できます。

-qi

オプション。このオプションを使用すると、ファイルを複数のメッセージに分割するために使用する区切り文字がメッセージに組み込まれます。**-dqdp** パラメーター (`prefix` または `postfix` を指定する) に従って、区切り文字はメッセージの先頭または末尾に組み込まれます。デフォルトでは、メッセージに区切り文字は組み込まれません。

-qi パラメーターを指定できるのは、**-dqdt** パラメーターと **-dqdb** パラメーターのいずれかを指定した場合に限られます。

-p configuration_options

オプション。このパラメーターは、ファイル転送を作成するために使用される構成オプションのセットを決定します。このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティ・ファイルのセットを使用します。

このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。

-w timeout

オプション。**-w** パラメーターを指定すると、**fteCreateTransfer** コマンドはエージェントからの応答を待機してから戻ります。このパラメーターを指定しなかった場合、**fteCreateTransfer** コマンドは、転送要求を受け取ったという確認応答を転送のソース・エージェントから受け取るまで、最大 5 秒間待機します。5 秒間の待機中に確認応答を受け取らなかった場合、**fteCreateTransfer** コマンドは以下の警告メッセージを返します。

```
BFGCL0253W: No acknowledgment to command from agent within timeout.
```

`timeout` 引数はオプションです。`timeout` を指定すると、**fteCreateTransfer** コマンドは `timeout` で示される秒数までエージェントの応答を待機します。制限時間に達する前にエージェントが応答しない場合、コマンドは警告を生成し、戻りコード 2 で終了します。`timeout` 値を指定しない場合、または `timeout` 値に `-1` を指定した場合、コマンドはエージェントが応答するまで待機します。

V 9.0.1 **-rt recovery_timeout**

オプション。停止したファイル転送のリカバリーをソース・エージェントが試行し続ける時間(秒単位)を設定します。以下のいずれかのオプションを指定します。

-1

エージェントは、停止した転送のリカバリーを、転送が完了するまで試行し続けます。このオプションを使用すると、このプロパティを設定しない場合のエージェントのデフォルトの動作と同じになります。

0

エージェントは、リカバリーに入るとすぐにファイル転送を停止します。

>0

エージェントは、指定された正整数値で設定された時間(秒単位)だけ、停止した転送のリカバリーを試行し続けます。例:

```
-rt 21600
```

これは、エージェントがリカバリーに入ってから6時間にわたって転送のリカバリーを試行し続けることを示しています。このパラメーターの最大値は999999999です。

このように指定した場合、転送のリカバリー・タイムアウト値は転送単位で設定されます。Managed File Transfer ネットワーク内のすべての転送が対象になるグローバルな値を設定するには、プロパティを [agent.properties](#) ファイルに追加します。

プログラムの呼び出しに関するパラメーター

Managed File Transfer からプログラムを開始する方法については、250 ページの『[MFTで実行するプログラムの指定](#)』を参照してください。ここで説明されているパラメーターを使用して起動するプログラムを指定する例については、844 ページの『[fteCreateTransfer を使用してプログラムを開始する例](#)』を参照してください。

-presrc pre_source_call

オプション。転送開始前にソース・エージェントで呼び出すプログラムを指定します。`pre_source_call` には次の形式を使用します。

```
[type:]commandspec[, [retrycount][, [retrywait][, successsrc]]]
```

この構文の変数は、以下のとおりです。

タイプ

オプション。有効な値は、**executable**、**antscript**、および **jcl** です。デフォルト値は **executable** です。

z/OS **jcl** の値は、z/OS 環境内のエージェントがターゲットである場合のみ適用されます。その場合は、このコマンドで、ZFS ファイル、QSAM 読み取り可能データ・セット、PDS メンバーのいずれかを参照します。その内容は、実行可能な JCL でなければなりません。

commandspec

必須。コマンドの指定。以下のいずれかの形式を使用します。

- タイプが **executable** の場合: `command[(arg1,arg2,...)]`

引数に `${FilePath}` または `${FileName}` のような変数置換が含まれている場合、変数は転送要求の最初の項目で置換されます。

例えば、転送要求がファイル「reports01.csv、reports02.csv、reports03.csv」で構成され、宛先ディレクトリが「/output」の場合、次の転送要求は次のようになります。

```
fteCreateTransfer -sa 1 -da 2 -presrc "executable:archive(${FileName})"  
-dd TargetDir "${FilePath}" -gt task.xml
```

は、以下と置換されます

```
fteCreateTransfer -sa 1 -da 2 -presrc "executable:archive(reports01.csv)"  
-dd TargetDir "/ouptut" -gt task.xml
```

- タイプが **antscript** の場合: `command[(name1=var1|target1,name2=var2|target2,...)]`
- タイプが **jcl** の場合: `command`

ここで、

command

必須。呼び出すプログラムの名前。

jcl の値は、z/OS 環境内のエージェントがターゲットである場合のみ適用されます。

大括弧 ([]) 内の引数はオプションです。また構文はコマンドのタイプによって異なります。コマンドまたはパラメーター内で括弧、コンマ (,)、および円記号 (¥) を使用する場合は、それらを円記号 (¥) でエスケープしてください。

retrycount

オプション。プログラムが成功を示す戻りコードを戻さなかった場合に、プログラムの呼び出しを再試行する回数。デフォルト値は 0 です。

retrywait

オプション。プログラム呼び出しを再度試行するまでの待機時間 (秒数)。デフォルト値は 0 (次の再試行まで待機なし)。

successsrc

オプション。プログラム呼び出しが正常に実行される条件を決定するために使用される式。この式は、1 つ以上の式の組み合わせにすることができます。それらの式は、ブール値の OR を表す垂直バー文字 (!)、またはブール値の AND を表すアンパーサンド (&) 文字で結合します。それぞれの式は、次の形式になります。

```
[>|<|!]value
```

この

>

オプション。value より大きい値であるかのテスト。

<

オプション。value より小さい値であるかのテスト。

!

オプション。value に等しくない値であるかのテスト。

値

必須。有効な整数。

-predst pre_destination_call

オプション。転送開始前に宛先エージェントで呼び出すプログラムを指定します。pre_destination_call は pre_source_call と同じ形式です。

-postsrc post_source_call

オプション。転送完了後にソース・エージェントで呼び出すプログラムを指定します。post_source_call は pre_source_call と同じ形式です。

-postdst post_destination_call

オプション。転送完了後に宛先エージェントで呼び出すプログラムを指定します。
`post_destination_call` は `pre_source_call` と同じ形式です。

宛先の指定に関するパラメーター

-td、**-df**、**-dd**、**-ds**、**-dq**、および **-dp** パラメーターのいずれかが必要です。これらのパラメーターは、相互に排他的であるため、転送要求に複数指定することができません。

-td transfer_definition_file

オプション。転送のためのソースおよび宛先のファイル仕様を 1 つ以上定義する XML 文書の名前。あるいは、管理対象の転送要求 (**-gt** パラメーターで生成した転送要求など) が含まれている XML 文書の名前を指定することもできます。**-td** パラメーターを指定した場合に、コマンド行で他のパラメーターも指定すると、それらのパラメーターによって、転送定義ファイルの対応する値がオーバーライドされます。

fteCreateTransfer コマンドは、現行ディレクトリーを基準として、転送定義ファイルの場所を探します。相対パス表記を使用して転送定義ファイルの場所を指定することができない場合は、代わりに、転送定義ファイルの完全修飾パスとファイル名を使用します。

z/OS z/OS では、z/OS UNIX システム・サービスの UNIX ファイルに、転送定義ファイルを保管する必要があります。z/OS の順次ファイルまたは PDS メンバーに、転送定義ファイルを保管することはできません。

IBM i IBM i では、統合ファイル・システムに転送定義ファイルを保管する必要があります。

詳しくは、『[転送定義ファイルの使用](#)』を参照してください。

-df destination_file

オプション。宛先ファイルの名前。

宛先エージェントが Connect:Direct ブリッジ・エージェントの場合は、宛先ファイルが `connect_direct_node_name:file_path` という形式で指定されます。Connect:Direct ブリッジ・エージェントはこの形式で指定されたファイル・パスのみを受け入れます。**z/OS** 宛先エージェントが Connect:Direct ブリッジ・エージェントであり、宛先が PDS メンバーになっている場合は、さらに **-de** パラメーターで `overwrite` という値を指定する必要があります。

次のことに注意してください。

- 宛先エージェントがプロトコル・ブリッジ・エージェントであり、ファイルのエンドポイントを指定する場合は、次の形式を使用します。

```
protocol_server:file_path
```

`protocol_server` はプロトコル・サーバーの名前 (これはオプション) で、`file_path` はプロトコル・サーバー・システム上のファイルのパスです。プロトコル・サーバーを指定しない場合は、デフォルトのプロトコル・サーバーが使用されます。

- 宛先エージェントに対して定義した Managed File Transfer 転送入出力ユーザー出口のいずれかを呼び出す場合は、転送に **-df** パラメーターを使用できます。
- z/OS** 宛先エージェントが z/OS 上にある場合、指定したファイルが `//` で始まるときは、z/OS 区分データ・セットが想定されます。

-dd destination_directory

オプション。ファイルの転送先となるディレクトリーの名前。宛先エージェントが実行されているシステムの有効なディレクトリー名を指定します。

宛先エージェントが Connect:Direct ブリッジ・エージェントの場合、宛先ディレクトリーは `connect_direct_node_name:directory_path` という形式で指定されます。宛先エージェント

が Connect:Direct ブリッジ・エージェントであり、宛先が PDS になっている場合は、さらに **-de** パラメーターで `overwrite` という値を指定する必要があります。

次のことに注意してください。

- 宛先エージェントがプロトコル・ブリッジ・エージェントであり、特定のエンドポイントにあるディレクトリーを指定する場合は、次の形式を使用します。

```
protocol_server:directory_path
```

`protocol_server` はプロトコル・サーバーの名前 (これはオプション) で、`directory_path` はプロトコル・サーバー・システム上でのディレクトリーのパスです。プロトコル・サーバーを指定しない場合は、デフォルトのプロトコル・サーバーが使用されます。

- 宛先エージェントに対して定義した Managed File Transfer 転送入出力ユーザー出口のいずれかを呼び出す場合は、転送に **-dd** パラメーターを使用できます。
- z/OS** 宛先エージェントが z/OS 上にある場合、指定したファイルが // で始まるときは、z/OS 区分データ・セットが想定されます。

z/OS **-ds destination_sequential_data_set**

z/OS のみ。オプション。ファイルの転送先の順次データ・セットまたは PDS メンバーの名前。順次データ・セット名または区分データ・セット・メンバーを指定します。データ・セットの転送について詳しくは、630 ページの『[ファイルの転送に関するガイドライン](#)』を参照してください。

データ・セット名の構文は次のとおりです。

```
//data_set_name{;attribute(value);...;attribute(value)}
```

または

```
//pds_data_set_name(member_name){;attribute(value);...;attribute(value)}
```

つまり、データ・セット名指定子の前に // があり、その指定子の後にはオプションで、セミコロンで区切られたいくつかの属性が続きます。

以下に例を示します。

```
//'TEST.FILE.NAME';DSNTYPE(PDS);RECFM(F,B);BLKSIZE(800);LRECL(80);CYL;SPACE(2,2)
```

データ・セットが Connect:Direct ノードにある場合は、データ・セット名の接頭部としてノード名を使用する必要があります。以下に例を示します。

```
CD_NODE1://'OBJECT.LIB';RECFM(F,B);BLKSIZE(800);LRECL(80)
```

宛先エージェントが Connect:Direct ブリッジ・エージェントであり、宛先が PDS メンバーになっている場合は、さらに **-de** パラメーターで `overwrite` という値を指定する必要があります。Connect:Direct ノードとのデータ・セット転送の詳細については、634 ページの『[Connect:Direct ノードとの間のデータ・セット転送](#)』を参照してください。

Managed File Transfer エージェントだけがかわっている転送の場合は、データ・セット名の部分を単一引用符文字で囲むと、その指定が完全修飾データ・セット名になります。データ・セット名を単一引用符文字で囲まないと、システムによって宛先エージェントのデフォルトの高位修飾子 (`transferRootHLQ` エージェント・プロパティの値か、`transferRootHLQ` が設定されていない場合は、エージェントを実行しているユーザー ID) が追加されます。

注: **z/OS** ただし、z/OS システム上の Connect:Direct ノードに係る転送では、データ・セット仕様は完全修飾名として解釈されます。データ・セット名に高位修飾子が追加されることはありません。データ・セット名を単一引用符文字で囲んだ場合でも、そのような動作になります。

ファイルまたはデータ・セットを磁気テープに転送すると、磁気テープ内の既存データ・セットが置き換えられます。新しいデータ・セットの属性は、転送定義で渡された属性を基にして設定されます。属性は、何も指定されていなければ、ソース・データ・セットと同様、またはデフォルト値(ソースがファイルである場合)に設定されます。既存の磁気テープ・データ・セットの属性は無視されます。

データ・セット属性は、データ・セットを作成するため、または既存のデータ・セットが互換性を持つようにするために使用されます。データ・セット属性の指定は、BPXWDYN に適した形式になります(詳細については、『[Requesting dynamic allocation](#)』を参照してください)。エージェントが宛先データ・セットを作成する場合、BPXWDYN 属性 DSN(*data_set_name*) NEW CATALOG MSG(*numeric_file_descriptor*) が自動的に指定されます。*numeric_file_descriptor* の値は Managed File Transfer によって生成されます。データ・セットからのデータ・セットへの転送では、ソースから RECFM、LRECL、および BLKSIZE の属性が、新規宛先データ・セット用に選択されます。新規宛先データ・セットの SPACE 設定は、Managed File Transfer によって設定されず、システムのデフォルトが使用されます。そのため、新規データ・セットが作成されるたびに、SPACE 属性を指定することをお勧めします。agent.properties ファイル内の **bpxwdynAllocAdditionalProperties** プロパティを使用して、すべての転送に適用する BPXWDYN オプションを設定できます。詳しくは、[550](#) ページの『[MFT agent.properties ファイル](#)』を参照してください。

z/OS いくつかの BPXWDYN オプションは、**fteCreateTemplate** コマンド、**fteCreateTransfer** コマンド、または agent.properties ファイル内の **bpxwdynAllocAdditionalProperties** プロパティを使用するときに指定してはなりません。これらのプロパティのリストについては、[642](#) ページの『[MFT で使用できない BPXWDYN のプロパティ](#)』を参照してください。

-ds パラメーターは、宛先エージェントがプロトコル・ブリッジ・エージェントである場合はサポートされません。

エージェントに対して定義した Managed File Transfer 転送入出力ユーザー出口のいずれかと呼び出す場合は、転送に **-ds** パラメーターを指定しないでください。**-ds** パラメーターを使用すると、転送入出力ユーザー出口は宛先に対して呼び出されなくなり、その代わりに標準の Managed File Transfer 入出力が使用されます。

z/OS **-dp destination_partitioned_data_set**

z/OS のみ。オプション。ファイルの転送先となる宛先 PDS の名前。区分データ・セット名を指定します。PDS が転送の結果として作成されると、この PDS はデフォルトで PDSE として作成されます。DSNTYPE=PDS を指定することにより、デフォルトをオーバーライドすることができます。

PDS データ・セット名の構文は次のとおりです。

```
//pds_data_set_name{;attribute;..;attribute}
```

データ・セット名の構文は、**-ds** (*destination_sequential_data_set*) パラメーターで説明しているものと同じです。Connect:Direct ノードにあるデータ・セットを指定するための構文の詳細はすべて、**-dp** パラメーターにも当てはまります。宛先エージェントが Connect:Direct ブリッジ・エージェントの場合は、さらに **-de** パラメーターで **overwrite** という値を指定する必要があります。

-dp パラメーターは、宛先エージェントがプロトコル・ブリッジ・エージェントである場合はサポートされません。

エージェントに対して定義した Managed File Transfer 転送入出力ユーザー出口のいずれかと呼び出す場合は、転送に **-dp** パラメーターを指定しないでください。**-dp** パラメーターを使用すると、転送入出力ユーザー出口は宛先に対して呼び出されなくなり、その代わりに標準の Managed File Transfer 入出力が使用されます。

-dq destination_queue

オプション。ファイルの転送先となる宛先キューの名前。この指定に QUEUE@QUEUEMANAGER というフォーマットでキュー・マネージャー名を含めることもできます。キュー・マネージャー名を指定しない場合は、宛先エージェントのキュー・マネージャー名が使用されます。対象のキュー・マネージャーに存在する有効なキュー名を指定する必要があります。

-dq パラメーターは、宛先エージェントがプロトコル・ブリッジ・エージェントまたは Connect:Direct ブリッジ・エージェントであるか、ソース指定がキューである場合はサポートされません。

エージェントに対して定義した Managed File Transfer 転送入力ユーザー出口のいずれかを呼び出す場合は、転送に **-dq** パラメーターを指定しないでください。 **-dq** パラメーターを使用すると、転送入力ユーザー出口は宛先に対して呼び出されなくなり、その代わりに標準の Managed File Transfer 入出力が使用されます。

-dq persistent

オプション。宛先キューに書き込むメッセージを永続メッセージにするかどうかを指定します。有効なオプションは以下のとおりです。

true

宛先キューに永続メッセージを書き込みます。これはデフォルト値です。

false

宛先キューに非永続メッセージを書き込みます。

qdef

永続性の値を宛先キューの DefPersistence 属性から取り込みます。

-dq パラメーターを指定できるのは、**-dq** パラメーターを指定した場合に限られます。

-dqdb hexadecimal_delimiter

オプション。バイナリー・ファイルを複数のメッセージに分割するときに使用する 16 進数区切り文字を指定します。メッセージは、すべて同じ IBM MQ グループ ID を持ちます。グループの最後のメッセージは IBM MQ LAST_MSG_IN_GROUP フラグ・セットを持ちます。区切り文字として 16 進数バイトを指定するためのフォーマットは、xNN です (N は、0-9 または a-f の範囲の文字です)。区切り文字として一連の 16 進数バイトを指定することもできます。その場合は、16 進数バイトのコンマ区切りリストを指定します (例: x3e, x20, x20, xbf)。

-dqdb パラメーターを指定できるのは、バイナリー・モードの転送で **-dq** パラメーターを指定した場合に限られます。 **-qs**、**-dqdb**、および **-dqdt** パラメーターのいずれかのみ指定できます。

-dqdt pattern

オプション。テキスト・ファイルを複数のメッセージに分割するときに使用する Java 正規表現を指定します。メッセージは、すべて同じ IBM MQ グループ ID を持ちます。グループの最後のメッセージは IBM MQ LAST_MSG_IN_GROUP フラグ・セットを持ちます。区切り文字として正規表現を指定するためのフォーマットは、(regular_expression) (正規表現を括弧で囲んだ形式) または "regular_expression" (正規表現を二重引用符で囲んだ形式) です。詳しくは、[659 ページの『MFT が使用する正規表現』](#)を参照してください。

デフォルトでは、正規表現にマッチング可能なストリングの長さは、宛先エージェントによって 5 文字に制限されています。この動作を変更するには、**maxDelimiterMatchLength** エージェント・プロパティを編集します。詳しくは、[551 ページの『拡張エージェント・プロパティ』](#)を参照してください。

-dqdt パラメーターを指定できるのは、**-dq** パラメーターを指定し、**-t** パラメーターの値として text を指定した場合に限られます。 **-qs**、**-dqdb**、および **-dqdt** パラメーターのいずれかのみ指定できます。

-dqdp position

オプション。ファイルを分割するときの、宛先テキストおよびバイナリー区切り文字の想定される位置を指定します。 **-dqdp** パラメーターを指定できるのは、**-dqdt** パラメーターと **-dqdb** パラメーターのいずれかを指定した場合に限られます。

以下のいずれかのオプションを指定します。

接頭部

区切り文字が各行の先頭にあることが想定されます。

postfix

区切り文字が各行の末尾にあることが想定されます。これはデフォルト・オプションです。

-de destination_file_behavior

オプション。宛先システムに宛先ファイルが存在する場合に取る処置を示します。有効なオプションは以下のとおりです。

エラー

エラーを報告し、ファイルは転送されません。これはデフォルト値です。

overwrite

既存の宛先ファイルを上書きします。

-de パラメーターを指定する場合、**-td** パラメーターを指定できません。これらのパラメーターは相互に排他的だからです。ただし、宛先ファイルが存在する場合の動作を転送定義ファイルに指定することができます。

-t transfer type

オプション。ファイル転送のタイプ(バイナリー・モードまたはテキスト・モード)を指定します。

binary

ファイル内のデータは変換されずに転送されます。これはデフォルト値です。

text

ファイルのコード・ページおよび行末文字は変換されます。**-sce**、**-dce**、または**-dle** パラメーターを使用して、変換に使用するコード・ページおよび行末を指定することができます。**-sce**、**-dce**、**-dle** のいずれのパラメーターも指定しない場合には、実際に実行される変換は、ソース・エージェントおよび宛先エージェントのオペレーティング・システムによって異なります。

 例えば、Windows から z/OS に転送されるファイルの場合、そのコード・ページは ASCII から EBCDIC に変換されます。ファイルを ASCII から EBCDIC に変換すると、行末文字は、ASCII 復帰 (CR) および改行 (LF) 文字のペアから EBCDIC 改行 (NL) 文字に変換されます。

 z/OS データ・セットの転送方法について詳しくは、『z/OS と分散システム間のファイルおよびデータ・セットの転送』および『データ・セット間の転送』を参照してください。

-t パラメーターを指定する場合、**-td** パラメーターを指定できません。これらのパラメーターは相互に排他的だからです。ただし、転送定義ファイルに転送モードの動作を指定することができます。

-dce destination_character_encoding

オプション。宛先でのファイルの書き込みに使用する文字エンコードを指定します。このオプションはテキスト・ファイルだけに適用可能なので、**-t text** も指定する必要があります。変換に使用可能なコード・ページは、宛先エージェントのプラットフォームによって異なることになります。使用可能なコード・ページのリストについては、683 ページの『MFT で使用できるコード・ページ』のトピックを参照してください。

noswaplfnl

デフォルトでは、Managed File Transfer は、サポートされる EBCDIC 文字セットで swaplfnl を使用します。swaplfnl を使用すると、EBCDIC LF 0x25 文字との間の文字セット・マッピングの動作が変わります。ただし、これにより、望ましくないマッピングが生じることがあります。この動作をオーバーライドするには、noswaplfnl を使用します。

-dle destination_line_ending

オプション。宛先でファイルが書き込まれるときに使用される行末文字を指定します。このオプションはテキスト・ファイルにのみ適用されるため、**-t text** パラメーターも指定してください。有効なオプションは次のとおりです。

LF

改行。次のプラットフォームでは、これがデフォルトです。

-  UNIX プラットフォーム
-  z/OS UNIX システム・サービスのファイル

EBCDIC ファイルに Managed File Transfer に付属の標準 EBCDIC コード・ページを使用する場合は、行末文字が LF 文字 (0x25) ではなく、NL 文字 (0x15) にマップされます。

CRLF

復帰の後に改行。 **Windows** これは Windows のデフォルトです。

z/OS 転送の宛先が z/OS データ・セットである場合、このオプションは無視されます。

-dtr

オプション。LRECL データ・セット属性よりも長い宛先レコードが切り捨てられることを指定します。このパラメーターを指定しない場合、それらのレコードは折り返されます。このパラメーターは、宛先がデータ・セットであるテキスト・モードの転送のみに有効です。

-dfa attributes

オプション。4690 で実行されている IBM MQ 8.0 Managed File Transfer エージェントに転送する場合、このパラメーターを使用して、転送の宛先ファイルに関連付けられているファイル属性のセミコロンの区切りリストを指定します。**-dfa** パラメーターは、値の有無に関わらず指定できます。例えば、値を指定しない場合は、以下のようになります。

```
-dfa ATTRIBUTE1;ATTRIBUTE2
```

例えば、値を指定する場合は以下のようになります。

```
-dfa ATTRIBUTE1(VALUE);ATTRIBUTE2(VALUE)
```

例えば、1つの属性に値を指定し、もう1つの属性に値を指定しない場合は、以下のようになります。

```
-dfa ATTRIBUTE1;ATTRIBUTE2(VALUE)
```

コマンド内では **-dfa** パラメーターを複数回使用できます。

4690 上のファイル属性について詳しくは、「IBM Documentation」の「[ファイル配布属性](#)」を参照してください。

セキュリティに関するパラメーター

-mquserid (userID)

オプション。コマンド・キュー・マネージャーで認証するユーザー ID を指定します。

-mqpassword (password)

オプション。コマンド・キュー・マネージャーで認証するパスワードを指定します。**-mquserid** パラメーターも一緒に指定する必要があります。**-mquserid** を指定した場合に **-mqpassword** を指定しないと、関連付けるパスワードを入力するというプロンプトが出されます。パスワードは表示されません。

ソースの指定に関するパラメーター

-sd source_file_disposition

オプション。ファイルからファイルへの転送またはファイルからメッセージへの転送で、ソース・ファイルが宛先に正常に転送された場合にそのソース・ファイルに対して実行するアクションを指定します。有効なオプションは以下のとおりです。

leave

ソース・ファイルは変更されません。これはデフォルト値です。

delete

ソース・ファイルは、それが正常に転送された後にソース・システムから削除されます。

注: メッセージからファイルへの転送では、ソース・キューにあるメッセージが正常に転送されると、そのメッセージは常に削除されます。つまり、メッセージからファイルへの転送で **-sd** パラメーターを **leave** に設定しても、その値は無視されます。

z/OS z/OSでは、ソースが磁気テープ・データ・セットの場合に、delete オプションを指定すると、磁気テープは再マウントされて、データ・セットが削除されます。この動作は、システム環境の動作のためにこのようになっています。

ソースがキューの場合に leave オプションを指定すると、このコマンドからエラーが返され、転送は要求されません。

ソース・エージェントが Connect:Direct ブリッジ・エージェントの場合に delete オプションを指定すると、動作は通常のソース後処理動作とは異なります。以下のいずれかになります。

- Connect:Direct が Managed File Transfer によって生成されるプロセスを使用してファイルまたはデータ・セットをソースから移動する場合、delete オプションを指定すると、転送が失敗します。ソース・ファイルの削除を指定するには、ユーザー定義の Connect:Direct プロセスを実行依頼してください。詳しくは、239 ページの『ファイル転送要求からのユーザー定義 Connect:Direct プロセスの送信』を参照してください。
- Connect:Direct がユーザー定義プロセスを使用して、ファイルまたはデータ・セットをソースから移動する場合は、%FTEFDISP 組み込みシンボリック変数によってこのパラメーターがプロセスに渡されます。ソースを削除するかどうかは、ユーザー定義プロセスによって決まります。転送から返される結果も、ユーザー定義プロセスから返される結果によって決まります。

-sd パラメーターを指定する場合、-td パラメーターを指定できません。これらのパラメーターは相互に排他的だからです。ただし、転送定義ファイルにソースの後処理の動作を指定することができます。

-r

オプション。source_specification がワイルドカード文字を含んでいる場合に、サブディレクトリーのファイルを再帰的に転送します。Managed File Transfer が source_specification としてワイルドカード文字を使用して示されている場合、ワイルドカード文字に一致するサブディレクトリーは、-r パラメーターを指定している場合に限り、転送されます。source_specification がサブディレクトリーに一致する場合、そのディレクトリーとサブディレクトリー内のすべてのファイル (隠しファイルも含む) は常に転送されます。

Managed File Transfer がワイルドカード文字を処理する方法について、詳しくは『ワイルドカード文字の使用』を参照してください。

-r パラメーターを指定する場合、-td パラメーターを指定できません。これらのパラメーターは相互に排他的だからです。ただし、転送定義ファイルに再帰的動作を指定することができます。

-sce source_character_encoding

オプション。文字変換を実行するときにソース・ファイルの読み取りに使用する文字エンコードを指定します。このオプションはテキスト・ファイルだけに適用可能なので、-t text も指定する必要があります。変換に使用可能なコード・ページは、宛先エージェントのプラットフォームによって異なります。変換は宛先システムで実行されるからです。使用可能なコード・ページのリストについては、683 ページの『MFT で使用できるコード・ページ』のトピックを参照してください。

noswaplfnl

デフォルトでは、Managed File Transfer は、サポートされる EBCDIC 文字セットで swaplfnl を使用します。swaplfnl を使用すると、EBCDIC LF 0x25 文字との間の文字セット・マッピングの動作が変わります。ただし、これにより、望ましくないマッピングが生じることがあります。この動作をオーバーライドするには、noswaplfnl を使用します。

z/OS -skeep

オプション。固定長形式のレコード単位ファイルからテキスト・モード転送の一部として読み取ったソース・レコードで (z/OS データ・セットなど)、末尾のスペースを保持することを指定します。このパラメーターを指定しない場合、末尾のスペースはソース・レコードから除去されます。

-srdb delimiter

オプション。レコード単位である (z/OS データ・セットなど) ソース・ファイルの場合、バイナリー・ファイルにレコードを追加するときに区切り文字として挿入する 1 つ以上のバイト値を指定します。それぞれの値は、接頭部 *x* を付けた 00 から FF の範囲の 2 桁の 16 進数字として指定する必要があります。複数のエントリーはコンマで区切ってください。以下に例を示します。

```
-srdb x0A
```

または

```
-srdb x0D,x0A
```

この転送はバイナリー・モードで構成してください。

-srdp position

オプション。ソース・レコードの区切り文字を挿入する位置を指定します。-srdp パラメーターを指定できるのは、-srdb パラメーターを指定した場合に限られます。

以下のいずれかのオプションを指定します。

接頭部

区切り文字は各レコードの先頭に挿入されます。

postfix

区切り文字は各レコードの末尾に挿入されます。これはデフォルト・オプションです。

-sq

オプション。転送のソースがキューであることを指定します。

エージェントに対して定義した Managed File Transfer 転送入出力ユーザー出口のいずれかを呼び出す場合は、転送に -sq パラメーターを指定しないでください。-sq パラメーターを使用すると、転送入出力ユーザー出口はソースに対して呼び出されなくなり、その代わりに標準の Managed File Transfer 入出力が使用されます。

-sqgi

オプション。メッセージを IBM MQ グループ ID によってグループ化する動作を指定します。完全に揃った最初のグループが宛先ファイルに書き込まれます。このパラメーターを指定しない場合は、ソース・キューに存在するすべてのメッセージが宛先ファイルに書き込まれます。

-sqgi パラメーターを指定できるのは、-sq パラメーターを指定した場合に限られます。

-sqdt text_delimiter

オプション。1 つのテキスト・ファイルに複数のメッセージを追加するときに区切り文字として挿入するテキストのシーケンスを指定します。ストリング・リテラルの Java エスケープ・シーケンスを区切り文字に含めることもできます。例えば、-sqdt \u007d\n です。

テキスト区切り文字は、転送のソース・エンコード方式に基づいてバイナリー・フォーマットにエンコードされます。各メッセージはバイナリー・フォーマットで読み取られます。(-sqdp パラメーターの指定どおりに) エンコードされた区切り文字がバイナリー・フォーマットでメッセージの前または後に付加され、結果がバイナリー・フォーマットで宛先エージェントに転送されます。ソース・エージェントのコード・ページにシフトイン状態とシフトアウト状態が含まれていれば、エージェントは、各メッセージがメッセージの末尾でシフトアウト状態になると想定します。宛先エージェントでは、ファイルからファイルへのテキスト転送の場合と同じ要領でバイナリー・データが変換されます。

-sqdt パラメーターを指定できるのは、-sq パラメーターを指定し、-t パラメーターの値として text を指定した場合に限られます。

-sqdb hexadecimal_delimiter

オプション。1 つのバイナリー・ファイルに複数のメッセージを追加するときに区切り文字として挿入する 1 つ以上のバイト値を指定します。それぞれの値は、00-FF の範囲にある 2 桁の 16 進数に接頭

部 x を付けた形式で指定する必要があります。複数バイトの場合はコンマで区切る必要があります。例えば、`-sqdb x08,xA4` のように指定します。

`-sqdb` パラメーターを指定できるのは、`-sq` パラメーターを指定した場合に限られます。`-t` パラメーターの値として `text` を指定した場合は、`-sqdb` パラメーターを指定できません。

-sqdp position

オプション。ソース・テキストおよびバイナリー区切り文字の挿入位置を指定します。`-sqdp` パラメーターを指定できるのは、`-sqdt` パラメーターと `-sqdb` パラメーターのいずれかを指定した場合に限られます。

以下のいずれかのオプションを指定します。

接頭部

区切り文字は各メッセージの先頭に挿入されます。

postfix

区切り文字は各メッセージの末尾に挿入されます。これはデフォルト・オプションです。

-sqwt wait_time

オプション。以下のいずれかの条件が満たされるのを待つ時間を秒単位で指定します。

- 新しいメッセージがキューに書き込まれるという条件
- `-sqgi` パラメーターを指定した場合は、完全に揃ったグループがキューに書き込まれるという条件

`wait_time` で指定した時間内にどちらの条件も満たされなければ、ソース・エージェントは、キューからの読み取りを停止して、転送を完了します。`-sqwt` パラメーターを指定しなければ、ソース・エージェントは、ソース・キューが空の場合にソース・キューからの読み取りをただちに停止します。あるいは、`-sqgi` パラメーターが指定されているのであれば、完全に揃ったグループがキューに存在しない場合にソース・キューからの読み取りをただちに停止します。

`-sqwt` パラメーターの使用については、682 ページの『メッセージからファイルへの転送の待機時間を指定する際のガイダンス』を参照してください。

`-sqwt` パラメーターを指定できるのは、`-sq` パラメーターを指定した場合に限られます。

source_specification

ファイル転送のソース (複数可) を決める 1 つ以上のファイル指定。

`-df`、`-dd`、`-dp`、`-dq`、`-ds` のいずれかのパラメーターを指定した場合に必要です。`-td` パラメーターを指定した場合は、`source_specification` を指定しないでください。

- `-sq` パラメーターを指定しない場合は、`source_specification` として、ファイル転送のソースを示すファイル指定を 1 つ以上記述します。ファイルの指定では 5 つの形式のうち 1 つを使用することができ、ワイルドカード文字を含めることができます。ワイルドカード文字について詳しくは、653 ページの『MFT でのワイルドカード文字の使用』を参照してください。ファイル指定でアスタリスク文字を 2 つ (**) 使用することにより、ファイル指定の一部を成すアスタリスクをエスケープすることができます。

複数のソース・ファイルをスペース文字で区切って指定することができます。ただし、`-df` パラメーターまたは `-ds` パラメーターに複数のソースを指定して、かつ `-de overwrite` を指定した場合、宛先には最後に指定したソース・ファイルのデータのみが入ります。`-de overwrite` を指定しない場合は、転送は一部だけが正常に行われます。宛先ファイルが存在していなかった場合は、最初に指定したソース・ファイルのデータが入ります。

ファイル名にスペースが含まれるファイルを転送する (例えば、`a b.txt` をファイル `c d.txt` に転送する) には、スペースが含まれるファイル名を二重引用符文字で囲みます。`fteCreateTransfer` コマンドの一部として次のテキストを指定します。

```
-df "c d.txt" "a b.txt"
```

ファイル指定はそれぞれ、以下のカテゴリーのいずれかでなければなりません。

ファイル名

ソース・エージェントが実行されているシステムに適切な表記で表現される、ファイルの名前。ファイル名をソース・ファイルとして指定すると、ファイルの内容がコピーされます。

ディレクトリー

ソース・エージェントが実行されているシステムに適切な表記で表現される、ディレクトリーの名前。ディレクトリーをソース・ファイルとして指定すると、ディレクトリーの内容がコピーされます。もっと正確に表現すると、ディレクトリーとそのすべてのサブディレクトリー内のすべてのファイル (隠しファイルを含む) がコピーされます。

例えば、DIR1 の内容を DIR2 のみにコピーするには、`fteCreateTransfer ... -dd DIR2 DIR1/*`と指定します。

順次データ・セット

順次データ・セットまたは区分データ・セット・メンバーの名前。データ・セットは、名前の先頭に 2 つのスラッシュ文字 (//) を付けて表します。

プロトコル・ブリッジ・エージェントをソース・エージェントとして指定する場合は、データ・セットをソース・ファイルとして指定することはできません。

区分データ・セット

区分データ・セットの名前。データ・セット名は、先頭に 2 つのスラッシュ文字 (//) を付けて表します。

プロトコル・ブリッジ・エージェントをソース・エージェントとして指定する場合は、データ・セットをソース・ファイルとして指定することはできません。

Connect:Direct ノードのファイル名またはディレクトリー

(Connect:Direct ブリッジ・エージェントのみ)。Connect:Direct ノードの名前、コロン文字 (:)、および Connect:Direct ノードをホストしているシステム上のファイル・パスまたはディレクトリー・パス。例えば、`connect_direct_node_name:file_path` のように指定します。

ソース・エージェントが Connect:Direct ブリッジ・エージェントの場合、この形式のソース指定のみが受け入れられます。

注: ソース・エージェントが Connect:Direct ブリッジ・エージェントである場合は、ファイル・パス内のワイルドカード文字はサポートされません。

プロトコル・ファイル・サーバー上のファイル名またはディレクトリー

プロトコル・ファイル・サーバーの名前、コロン文字 (:)、およびプロトコル・サーバー・システムでのファイルまたはディレクトリー・パス。例えば、`protocol_server:file_path` のように指定します。

プロトコル・サーバーを指定しない場合は、デフォルトのプロトコル・サーバーが使用されます。

- **-sq** パラメーターを指定した場合は、*source_specification* として、ソース・エージェントのキュー・マネージャーに存在するローカル・キューの名前を指定します。指定できるソース・キューは、1 つだけです。ソース・キューを指定するためのフォーマットは、以下のとおりです。

QUEUE_NAME

ソース・キューの指定には、キュー・マネージャー名を含めません。そのキュー・マネージャーは、ソース・エージェントのキュー・マネージャーと同じでなければなりません。

-  ソース・エージェントが z/OS 上にある場合、//で始まるソース・ファイルは z/OS 区分データ・セットであると想定されます。

その他のパラメーター

-? または -h

オプション。コマンド構文を表示します。

例

この基本的な例では、ファイル `originalfile.txt` は同じシステム上の AGENT1 から AGENT2 に転送され、`transferredfile.txt` に名前変更されます。

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

この例では、`originalfile.txt` ファイルと `originalfile2.txt` ファイルを同じシステムにある AGENT1 から AGENT2 に転送します。宛先のディレクトリーは、`C:\import` です。

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -dd C:\import C:\export\originalfile.txt  
C:\export\originalfile2.txt
```

この例では、ファイル `originalfile.txt` は AGENT1 のシステムから AGENT2 のシステムに転送されます。ファイル転送は、ソース・エージェントのシステムのシステム時刻を基準として 09:00 に実行されるようにスケジュールされます。これは 2 時間ごとに 4 回実行されます。

```
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
```

この例では、ファイル `originalfile.txt` は、ファイル `A.txt` が AGENT1 に存在するという条件で AGENT1 から AGENT2 に転送されます。

```
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 この例では、ファイル `originalfile.txt` が AGENT1 のシステムから AGENT2 のシステムにおけるデータ・セット `///USERID.TRANS.FILE.TXT` へ転送されます。ASCII から EBCDIC へのデータ変換にテキスト・モードが選択されています。

```
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 この例では、AGENT1 のシステムの完全修飾データ・セットのメンバーが、AGENT2 のシステム上のファイルに転送されます。EBCDIC から AGENT2 のシステムのデフォルト・コード・ページにファイルを変換するためにテキスト・モードが選択されています。

```
fteCreateTransfer -t text -sa AGENT1 -da AGENT2 -df /tmp/IEEUJV.txt "'/'SYS1.SAMPLIB(IEEUJV)'"
```

この例では、宛先エージェント BRIDGE1 を使用して、エージェント AGENT1 にある `file.bin` というファイルが、プロトコル・ファイル・サーバー `accountshost.ibm.com` 上の `file.bin` という宛先ファイルに転送されます。

```
fteCreateTransfer -sa AGENT1 -da BRIDGE1 -df accountshost.ibm.com:/tmp/file.bin /tmp/file.bin
```

次の例では、引用符を付けないワイルドカードを使用しています。`.txt` で終わる AGENT1 の現行作業ディレクトリー内のすべてのファイルは、AGENT2 上のディレクトリー `C:\import` に転送されます。ファイル名は変更されません。

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -dd C:\import *.txt
```

次の例では、二重引用符を付けたワイルドカードを使用しています。 .txt で終わる AGENT1 の転送ルート・ディレクトリー内のすべてのファイルは、AGENT2 上のディレクトリー C:\import に転送されます。ファイル名は変更されません。

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -dd C:\import "*.txt"
```

戻りコード

戻りコード	説明
0	コマンドは正常に完了しました。
1	コマンドは失敗しました。
2	コマンドはタイムアウトになりました。 コマンドはメッセージをエージェントに送信しましたが、エージェントは指定された時間内に応答しませんでした。
20	コマンドが部分的に成功して完了し、一部のファイルが転送されました。
21	fteCreateTransfer コマンドの接続先のキュー・マネージャーが、転送結果の判定前に停止されました。
40	失敗しました。 指定したどのファイルも転送できませんでした。
41	転送は取り消されました。
42	転送が条件付きであり、必要な条件が満たされなかったため、転送は実行されませんでした。
43	転送要求メッセージの形式が誤っていました。
44	ソース・エージェントに、転送を実行するための容量が十分に存在しません。
45	宛先エージェントに、転送を実行するための容量が十分に存在しません。
46	転送ファイル数が、ソース・エージェントの制限を超過しました。
47	転送ファイル数が、宛先エージェントの制限を超過しました。

fteDefine (構成スクリプトの生成)

fteDefine コマンドを使用して、指定されたエージェント・キュー・マネージャー・オブジェクトを定義するために必要な構成スクリプトを生成します。

目的

fteDefine コマンドは、構成データが含まれているシステムに対してリモートであるシステムで構成ステップを実行する必要がある場合に使用することが想定されています。例えば、クライアント接続を介してアクセスするように、キュー・マネージャー上のエージェントのキューを構成するときに使用します。

構文

<pre> fteDefine ▶▶ fteDefine -t type -d outputDir name ▶▶ </pre>

Parameters

-t type

必須。 定義するオブジェクトのタイプ。 タイプのオプションは agent です。

-d outputDir

オプション。スクリプトを書き込む先のディレクトリーのパス。指定しなかった場合は、スクリプトが標準出力ストリームに書き込まれます。

名前

必須。定義する1つ以上のオブジェクトの名前。複数のオブジェクトの名前を指定するには、オブジェクトをスペースで区切ります。name1 name2... に例を示します

-? または -h

オプション。コマンド構文を表示します。

例

この例では、**-t agent** パラメーターと単一のエージェント名を使用して **fteDefine** コマンドが指定されています。出力はファイルに書き込まれます。

```
fteDefine -t agent EXAMPLE.AGENT >EXAMPLE.AGENT_create.mqsc
```

このコマンドから生成される出力は、必要なエージェント・キューを作成するためにエージェント・キュー・マネージャーに対して実行する MQSC コマンド・スクリプトです。

```
$ fteDefine -t agent EXAMPLE.AGENT
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
DEFINE QLOCAL(SYSTEM.FTE.COMMAND.EXAMPLE.AGENT) +
  DEFPRTY(0) +
  DEFSOPT(SHARED) +
  GET(ENABLED) +
  MAXDEPTH(5000) +
  MAXMSGL(4194304) +
  MSGDLVSQ(PRIORITY) +
  PUT(ENABLED) +
  RETINTVL(999999999) +
  SHARE +
  NOTRIGGER +
  USAGE(NORMAL) +
  REPLACE
DEFINE QLOCAL(SYSTEM.FTE.DATA.EXAMPLE.AGENT) +
  DEFPRTY(0) +
  DEFSOPT(SHARED) +
  GET(ENABLED) +
  MAXDEPTH(5000) +
  MAXMSGL(4194304) +
  MSGDLVSQ(PRIORITY) +
  PUT(ENABLED) +
  RETINTVL(999999999) +
  SHARE +
  NOTRIGGER +
  USAGE(NORMAL) +
  REPLACE
...
etc.
```

この例では、**-d outputDir** パラメーターといくつかのエージェント名を使用して **fteDefine** コマンドが指定されています。

```
fteDefine -t agent -d /tmp EXAMPLE.AGENT.1 EXAMPLE.AGENT.2 EXAMPLE.AGENT.3
```

このコマンドから生成される出力は、MQSC コマンド・スクリプトの場所を表す絶対ファイル・パスです。

```
$ 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'.
```

戻りコード

0
コマンドは正常に完了しました。

1
コマンドは失敗しました。

関連資料

461 ページの『[fteDelete \(構成を削除するスクリプトの生成\)](#)』

fteDelete コマンドを使用して、指定されたエージェント・キュー・マネージャー・オブジェクトを除去するために必要な構成スクリプトを生成します。

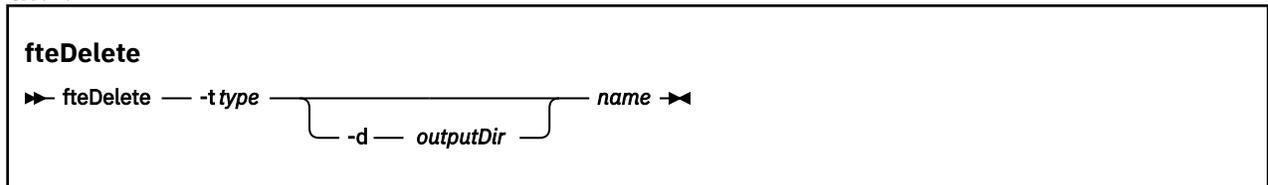
fteDelete (構成を削除するスクリプトの生成)

fteDelete コマンドを使用して、指定されたエージェント・キュー・マネージャー・オブジェクトを除去するために必要な構成スクリプトを生成します。

目的

fteDelete コマンドは、構成データが含まれているシステムに対してリモートであるシステムで構成ステップを実行する必要がある場合に使用することが想定されています。例えば、ローカル・キュー・マネージャー上のリモート・クライアント・エージェントのキューを削除するときに使用します。

構文



Parameters

-t type

必須。削除するオブジェクトのタイプ。タイプのオプションは agent です。

-d outputDir

オプション。スクリプトを書き込む先のディレクトリーのパス。指定しなかった場合は、スクリプトが標準出力ストリームに書き込まれます。

名前

必須。削除する 1 つ以上のオブジェクトの名前。複数のオブジェクトの名前を指定するには、オブジェクトをスペースで区切ります。name1 name2... に例を示します

-? または -h

オプション。コマンド構文を表示します。

例

この例では、**-t agent** パラメーターと単一のエージェント名を使用して **fteDelete** コマンドが指定されています。出力はファイルに書き込まれます。

```
fteDelete -t agent EXAMPLE.AGENT >EXAMPLE.AGENT_delete.mqsc
```

このコマンドから生成される出力は、エージェント・キューを削除するためにエージェント・キュー・マネージャーに対して実行する MQSC コマンド・スクリプトです。

```
$ 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.
```

この例では、**-d outputDir** パラメーターといくつかのエージェント名を使用して **fteDelete** コマンドが指定されています。

```
fteDelete -t agent -d /tmp EXAMPLE.AGENT.1 EXAMPLE.AGENT.2 EXAMPLE.AGENT.3
```

このコマンドから生成される出力は、MQSC コマンド・スクリプトの場所を表す絶対ファイル・パスです。

```
$ 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'.
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

関連資料

459 ページの『[fteDefine \(構成スクリプトの生成\)](#)』

fteDefine コマンドを使用して、指定されたエージェント・キュー・マネージャー・オブジェクトを定義するために必要な構成スクリプトを生成します。

fteDeleteAgent

fteDeleteAgent コマンドは、Managed File Transfer Agent とその構成を削除します。エージェントがプロトコル・ブリッジ・エージェントの場合は、ユーザー資格情報ファイルがファイル・システムに残ります。

目的

fteDeleteAgent コマンドを実行する前に、[fteStopAgent](#) コマンドを使用してエージェントを停止します。

エージェントを Windows サービスとして実行するように構成した場合、**fteDeleteAgent** コマンドを実行すると、サービス定義が削除されます。

V 9.0.4 IBM MQ 9.0.4 以降では、エージェントが削除されると、すべてのリソース・モニターおよびスケジュールされた転送が除去されます。

このコマンドを実行できるユーザーは、IBM MQ の管理者 (および mqm グループのメンバー) のみです。IBM MQ の管理者以外のユーザーとしてこのコマンドを実行しようとする、エラー・メッセージが表示され、コマンドは実行されません。

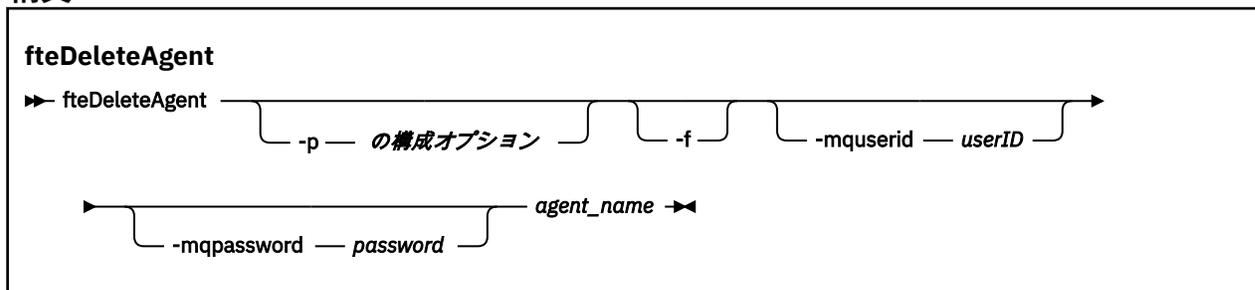
fteDeleteAgent コマンドは MQSC コマンドを提供します。これをエージェントのキュー・マネージャーに対して実行し、エージェントのシステム・キューをクリアして削除する必要があります。キューは以下のとおりです。

- 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

さらに、**fteCreateAgent** コマンドは、これらのコマンドを次の場所にあるファイルに提供します。

```
MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name/agent_name_delete.mqsc
```

構文



Parameters

-p (configuration_options)

オプション。複数の調整キュー・マネージャーがある場合は、このパラメーターを使用して、削除するエージェント構成を明示的に指定します。このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられた構成オプションを使用します。

オプションの **-p** パラメーターは、デフォルトと異なる構成オプションを使用する場合にのみ指定します。**-p** を指定しない場合、`installation.properties` ファイルで定義された構成オプションが使用されます。詳しくは、『57 ページの『MFT 構成オプション (Multiplatforms)』』を参照してください。

-f

オプション。エージェントの構成ファイルが見つからない場合でも、調整キュー・マネージャーからエージェントを登録解除するコマンドを強制的に実行します。この状態では、エージェントのキュー・マネージャーに関する情報は使用できないので、コマンドは通常であれば使用するエージェントのキュー・マネージャーを使用せずに、調整キュー・マネージャーに直接接続します。

-mquserid (userID)

オプション。エージェント・キュー・マネージャーで認証するユーザー ID を指定します (強制パラメーター **-f** が指定されている場合を除く)。 **-f** パラメーターが指定されている場合は、調整キュー・マネージャーで認証するユーザー ID を指定します。

-mqpassword (password)

オプション。エージェント・キュー・マネージャーで認証するパスワードを指定します (強制パラメーター **-f** が指定されている場合を除く)。 **-f** パラメーターが指定されている場合は、調整キュー・マネージャーで認証するパスワードを指定します。 **-mquserid** パラメーターも一緒に指定する必要があります。 **-mquserid** を指定した場合に **-mqpassword** を指定しないと、関連付けるパスワードを入力するようというプロンプトが出されます。パスワードは表示されません。

agent_name

必須。削除するエージェントの名前。

-? または -h

オプション。コマンド構文を表示します。

例

以下の例では、AGENT3 と、調整キュー・マネージャー QM_COORD1 上のこのエージェントの構成が削除されます。

```
fteDeleteAgent -p QM_COORD1 AGENT3
```

このコマンド例では、エージェントの 3 つのキューを削除するための以下の MQSC コマンドが出力されます。

```
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)
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

関連資料

[531 ページの『fteStopAgent』](#)

fteStopAgent コマンドは、Managed File Transfer のエージェントを制御された仕方で停止させるか、必要であれば **-i** パラメーターを使用してエージェントを即時に停止させるために使用します。

[384 ページの『fteCleanAgent: MFT エージェントのクリーンアップ』](#)

fteCleanAgent コマンドを使用して、Managed File Transfer Agent が使用する永続的および非永続的キューからメッセージを削除することにより、エージェントが使用するキューをクリーンアップします。エージェントの開始に問題があり、エージェントが使用するキューに情報が残っていることが原因である可能性がある場合、**fteCleanAgent** コマンドを使用します。

388 ページの『[fteCreateAgent \(MFT エージェントの作成\)](#)』

fteCreateAgent コマンドは、Managed File Transfer Agent とその関連構成を作成します。

528 ページの『[fteStartAgent](#)』

fteStartAgent コマンドは、Managed File Transfer のエージェントをコマンド行から開始します。

fteDeleteLogger

fteDeleteLogger コマンドは、Managed File Transfer ロガーとその構成を削除するために使用します。ロガーに関連付けられている既存のログ・ファイルは、保持することも削除することも可能です。

重要: **ULW** IBM MQ for UNIX, Linux, and Windows では、IBM MQ 管理者 (および mqm グループのメンバー) であるユーザーのみが、このコマンドを実行できます。IBM MQ 管理者以外のユーザーとしてこのコマンドを実行しようとすると、エラー・メッセージ BFGCL0502E: 要求された操作を実行する権限がありません。を受け取ります。コマンドは実行されません。

z/OS z/OS システムでは、マイグレーション・コマンドを実行するためには、ユーザーは以下の条件の (少なくとも) 1 つを満たす必要があります。

- mqm グループのメンバーである (システムに mqm グループが定義されている場合)。
- **V9.0.0.1** BFG_GROUP_NAME 環境変数で名前が指定されたグループのメンバーである (名前が指定されている場合)。
- **V9.0.0.1** コマンド実行時に BFG_GROUP_NAME 環境変数に値が設定されていない。

目的

fteDeleteLogger コマンドを実行する前に、**fteStopLogger** コマンドでロガーを停止します。

ロガーを Windows サービスとして実行するように構成した場合、**fteDeleteLogger** コマンドを実行すると、サービス定義が削除されます。

ロガーの構成ディレクトリーには、そのロガーのキューおよびサブスクリプションを削除するための MQSC スクリプトが含まれます。キューは以下のとおりです。

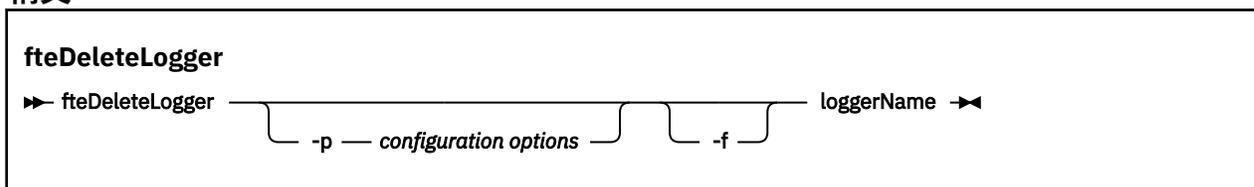
- SYSTEM.FTE.LOG.CMD.*logger_name*
- SYSTEM.FTE.LOG.RJCT.*logger_name*

サブスクリプション名は以下のとおりです。

- SYSTEM.FTE.AUTO.SUB.*logger_name*

MQSC スクリプトは、
MQ_DATA_PATH\mqft\config\coordination_qmgr\loggers*logger_name**logger_name_delete*.mqsc にあります

構文



Parameters

-p (configuration_options)

オプション。スタンドアロン・データベース・ロガーの開始に使用される構成オプションのセットを決定します。 **-p** パラメーターの値として、一連の構成オプションの名前を使用します。慣例として、この値は調整キュー・マネージャーの名前です。このパラメーターを指定しない場合、デフォルトの構成オプションのセットが使用されます。

-f

オプション。このロガーによって作成されたすべてのログ・ファイルの削除を強制します。このパラメーターを省略した場合、このロガーによって作成されたログ・ファイルはすべて保存されるので、不要になったときには手動で削除する必要があります。

logger_name

必須。削除するロガーの名前。

-? または -h

オプション。コマンド構文を表示します。

例

この例では、`logger1` というロガーが削除されます。 **-f** パラメーターが指定されました。これにより、ロガーのログ・ファイルとロガーの構成ファイルが削除されます。

```
fteDeleteLogger -f logger1
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

関連資料

[408 ページの『fteCreateLogger \(MFT ファイルまたはデータベース・ロガーの作成\)』](#)

fteCreateLogger コマンドを使用して、Managed File Transfer ファイルまたはデータベース・ロガーを作成します。

[530 ページの『fteStartLogger』](#)

fteStartLogger コマンドは、Managed File Transfer ロギング・アプリケーションを開始します。

[533 ページの『fteStopLogger』](#)

fteStopLogger コマンドは Managed File Transfer ロガーを停止します。

[492 ページの『fteModifyLogger \(Windows サービスとしての MFT ロガーの実行\)』](#)

Managed File Transfer ロガーを Windows サービスとして実行できるように変更するには、**fteModifyLogger** コマンドを使用します。このコマンドは Windows プラットフォームでのみ使用できます。IBM MQ 管理者であり、mqm グループのメンバーであるユーザーが実行する必要があります。まず、**fteStopLogger** コマンドを使用してロガーを停止する必要があります。

fteDeleteMonitor: MFT リソース・モニターの削除

コマンド行で Managed File Transfer の既存のリソース・モニターを停止して削除する場合には、**fteDeleteMonitor** コマンドを使用します。リソース・モニター・エージェントに対してこのコマンドを発行します。

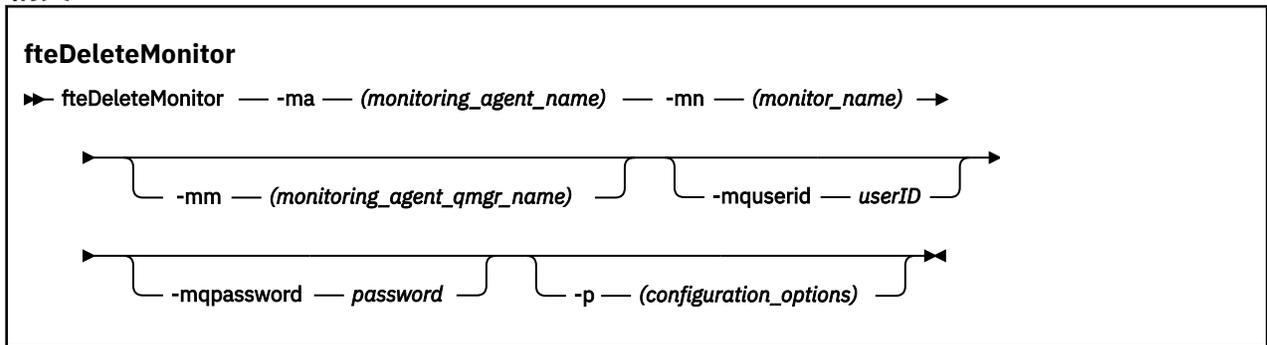
目的

fteDeleteMonitor コマンドは、リソースのモニターを停止して、モニターの定義をモニター・エージェントから削除する場合に使用します。このコマンドを実行すると、それ以降、リソースのポーリングは発生せず、タスクも開始されなくなります。

fteDeleteMonitor コマンドは、IBM MQ ネットワークに接続してからエージェントのキュー・マネージャーに経路指定のできる、任意のシステムから実行できます。具体的には、このコマンドを実行する場合、このシステムに Managed File Transfer コンポーネント (サービスまたはエージェント) がインストールされている必要があります。さらに IBM MQ ネットワークと通信するようにこのシステムの Managed File Transfer が構成されていることが必要です。接続の詳細を使用できない場合、接続にそのエージェントのキュー・マネージャーの詳細が使用されます (詳細が使用可能な場合)。

オプションの **-p** パラメーターは、デフォルト・セットと異なる構成オプションのセットを使用する場合にのみこのコマンドに指定します。詳しくは、[構成オプション](#)を参照してください。

構文



Parameters

-ma (*monitoring_agent_name*)

必須。リソース・モニターを実行するエージェントの名前。このモニター・エージェントは、トリガー対象のファイル転送のソース・エージェントでもある必要があります。

-mn (*monitor_name*)

必須。このリソース・モニターに割り当てた名前。リソース・モニターを削除することができ、その後同じ名前の新規モニターを作成できます。

-mm (*monitoring_agent_qmgr_name*)

オプション。モニター・エージェントのキュー・マネージャーの名前。モニター・エージェントとモニターがトリガーした転送のソース・エージェントが同一である必要があるため、このキュー・マネージャーはソース・エージェントのキュー・マネージャーでもあります。

-mquserid (*userID*)

オプション。コマンド・キュー・マネージャーで認証するユーザー ID を指定します。

-mqpassword (*password*)

オプション。コマンド・キュー・マネージャーで認証するパスワードを指定します。 **-mquserid** パラメーターも一緒に指定する必要があります。 **-mquserid** を指定した場合に **-mqpassword** を指定しないと、関連付けるパスワードを入力するというプロンプトが出されます。パスワードは表示されません。

-p (*configuration_options*)

オプション。このパラメーターは、転送を取り消すために使用する構成オプションのセットを決定します。慣例として、このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティ・ファイルのセットを使用します。

このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。

-?または-h

オプション。コマンド構文を表示します。

例

以下の例では、モニター中の (そしてファイル転送ソース・エージェントである) AGENT1 を持つリソース・モニター MONITOR1 が削除されます。

```
fteDeleteMonitor -ma AGENT1 -mm QM_JUPITER -mn MONITOR1
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

関連タスク

[170 ページの『MFT リソースのモニター』](#)

キューやディレクトリーなどの Managed File Transfer リソースをモニターできます。そのリソースで条件が満たされると、リソース・モニターがファイル転送などのタスクを開始します。**fteCreateMonitor** コマンドを使用するか、IBM MQ Explorer に対応した Managed File Transfer プラグインの「モニター」ビューを使用して、リソース・モニターを作成できます。

関連資料

[413 ページの『fteCreateMonitor: MFT リソース・モニターの作成』](#)

fteCreateMonitor コマンドは、コマンド行から新規リソース・モニターを作成して開始します。

Managed File Transfer を使用してリソース (例えば、ディレクトリーの内容) をモニターして、トリガー条件が満たされたときに、指定したタスク (ファイル転送など) を開始することができます。

[476 ページの『fteListMonitors: MFT リソース・モニターのリスト』](#)

コマンド行から **fteListMonitors** コマンドを使用すると、Managed File Transfer ネットワークの既存のリソース・モニターをすべてリストできます。

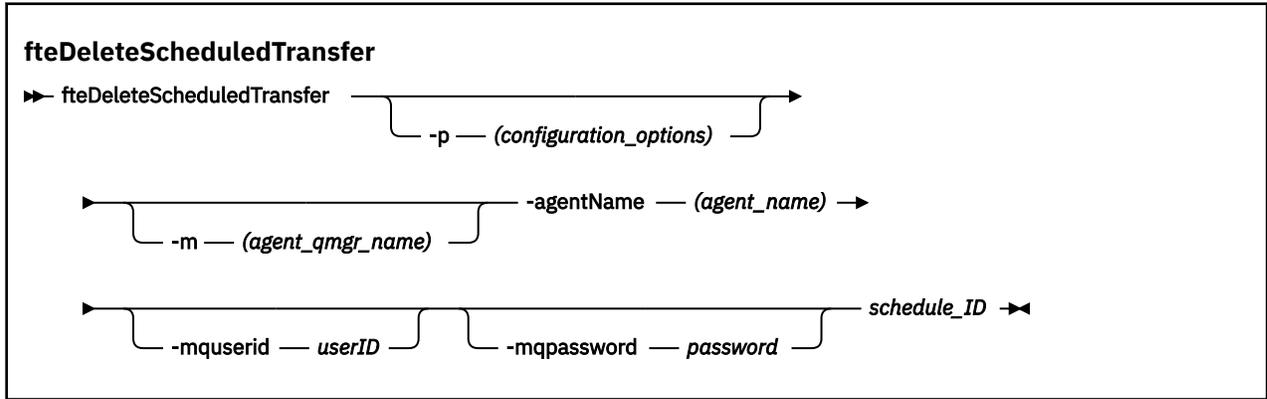
fteDeleteScheduledTransfer

目的

fteDeleteScheduledTransfer コマンドを使用して、コマンド行または IBM MQ Explorer を使用して以前に作成した Managed File Transfer スケジュール済み転送を削除します。

オプションの **-p** パラメーターは、デフォルトと異なる構成オプションを使用する場合にのみこのコマンドに指定します。**-p** を指定しない場合、`installation.properties` で定義された構成オプションが使用されます。詳しくは、『[57 ページの『MFT 構成オプション \(Multiplatforms\)』](#)』を参照してください。

構文



Parameters

-p (*configuration_options*)

オプション。複数の調整キュー・マネージャーがある場合は、このパラメーターを使用して、削除するスケジュール済み転送を明示的に指定します。このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられた構成オプションを使用します。

このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションが使用されます。

-m (*agent_qmgr_name*)

オプション。ソース・エージェントの接続先のキュー・マネージャーの名前。このパラメーターを指定しない場合、エージェントのキュー・マネージャーは、使用中の構成オプションから判別されます。

-agentName (*agent_name*)

必須。スケジュール済み転送を削除するソース・エージェントの名前。

-mquserid (*userID*)

オプション。コマンド・キュー・マネージャーで認証するユーザー ID を指定します。

-mqpassword (*password*)

オプション。コマンド・キュー・マネージャーで認証するパスワードを指定します。-mquserid パラメーターも一緒に指定する必要があります。-mquserid を指定した場合に -mqpassword を指定しないと、関連付けるパスワードを入力するというプロンプトが出されます。パスワードは表示されません。

schedule_ID

必須。削除するスケジュール済み転送の ID。

スケジュール ID は、[fteListScheduledTransfers](#) コマンドをソース・エージェントの名前に対して実行することにより検出できます。

-? または -h

オプション。コマンド構文を表示します。

例

この例では、ソース・エージェント AGENT2、ID 27 のスケジュール済み転送が削除されます。

```
fteDeleteScheduledTransfer -agentName AGENT2 27
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

関連タスク

163 ページの『スケジュール済みファイル転送の作成』

IBM MQ Explorer エクスプローラーまたはコマンド行のいずれかを使用して新規ファイル転送をスケジュールに入れられます。スケジュール済みの転送には、単一のファイルまたは1つのグループの複数のファイルを含めることができます。スケジュール済みファイル転送は、1回実行することも複数回転送を繰り返すこともできます。

関連資料

479 ページの『fteListScheduledTransfers』

fteListScheduledTransfers コマンドは、コマンド行または IBM MQ Explorer を使用して以前に作成した Managed File Transfer のすべての転送をリストするために使用します。

fteDeleteTemplates

fteDeleteTemplates コマンドを使用して、調整キュー・マネージャーから既存の Managed File Transfer テンプレートを削除します。

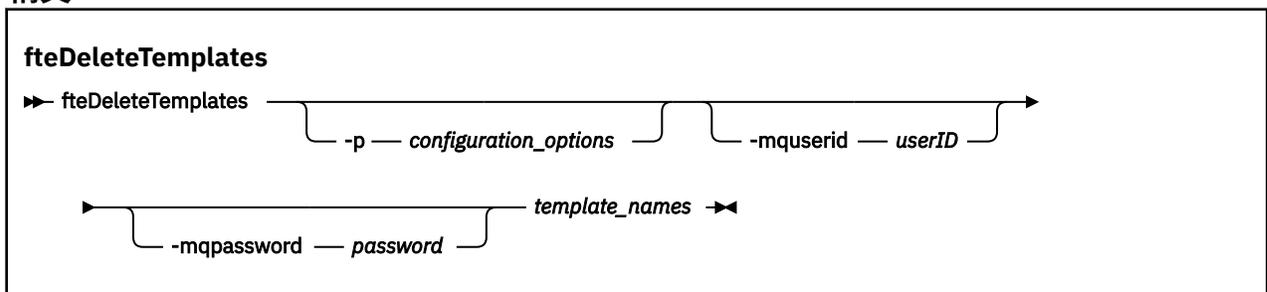
目的

fteDeleteTemplates コマンドを使用すると、調整キュー・マネージャーから1つ以上のファイル転送テンプレートが削除されます。このコマンドを実行すると、調整キュー・マネージャーからテンプレートを削除する要求が IBM MQ システムに渡され、対象のテンプレートは、IBM MQ エクスプローラーやコマンド行で使用できなくなります。コマンドの完了後も、IBM MQ システムが要求を処理するまでの短い時間、削除するテンプレートへのアクセスが引き続き可能になることもあります。

fteDeleteTemplates コマンドは、IBM MQ ネットワークに接続してから調整キュー・マネージャーに経路指定できる、どのシステムからでも実行できます。特に、このコマンドを実行するには、このシステムに Managed File Transfer がインストール済みであること、および IBM MQ ネットワークと通信するために、このシステムで Managed File Transfer が構成済みであることが必要です。接続の詳細を使用できない場合、接続にそのエージェントのキュー・マネージャーの詳細が使用されます (詳細が使用可能な場合)。

オプションの **-p** パラメーターは、デフォルト・セットと異なる構成オプションのセットを使用する場合にのみこのコマンドに指定します。詳しくは、[構成オプション](#)を参照してください。

構文



Parameters

-p (configuration_options)

オプション。このパラメーターは、テンプレートを削除するために使用する構成オプションのセットを決定します。慣例として、このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティ・ファイルのセットを使用します。

このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。

-mquserid (userID)

オプション。調整キュー・マネージャーで認証するユーザー ID を指定します。

-mqpassword (password)

オプション。調整キュー・マネージャーで認証するパスワードを指定します。 **-mquserid** パラメーターも一緒に指定する必要があります。 **-mquserid** を指定した場合に **-mqpassword** を指定しないと、関連付けるパスワードを入力するというプロンプトが出されます。パスワードは表示されません。

(template_names)

必須。削除するテンプレート名を 1 つ以上指定します。 **fteListTemplates** コマンドで表示される名前を指定してください。

-? または -h

オプション。コマンド構文を表示します。

例

この例では、STANDBY テンプレートを削除します。

```
fteDeleteTemplates STANDBY
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

関連タスク

[195 ページの『ファイル転送テンプレートの処理』](#)

ファイル転送テンプレートを使用すると、繰り返しの転送または複雑な転送を行うための共通のファイル転送設定を保管できます。転送テンプレートは **fteCreateTemplate** コマンドを使用してコマンド行から作成します。また、IBM MQ Explorer で、「ファイル転送管理のテンプレート新規作成」ウィザードを使用して転送テンプレートを作成することも、ファイル転送の作成時に「転送設定をテンプレートとして保存する」チェック・ボックスを選択してテンプレートを保存することもできます。「転送テンプレート」ウィンドウには、Managed File Transfer ネットワーク内に作成した転送テンプレートがすべて表示されます。

[197 ページの『IBM MQ Explorer を使用したファイル転送テンプレートの作成』](#)

ファイル転送テンプレートを IBM MQ Explorer またはコマンド行から作成することができます。その後そのテンプレートを使用してそのテンプレート詳細を使用する新規ファイル転送を作成したり、そのテンプレートを送信してファイル転送を開始したりできます。

関連資料

[421 ページの『fteCreateTemplate: 新規ファイル転送テンプレートの作成』](#)

fteCreateTemplate コマンドは、将来の使用のために保持できるファイル転送テンプレートを作成します。必須パラメーターは **-tn (template_name)** パラメーターのみです。その他のパラメーターはすべてオプションです。ただし、ソース・ファイル仕様を指定する場合には、宛先ファイルも提供する必要があります。同様に、宛先ファイルを指定する場合には、ソース・ファイル仕様も指定する必要があります。

[480 ページの『fteListTemplates』](#)

fteListTemplates コマンドを使用して、調整キュー・マネージャーで使用できる Managed File Transfer の転送テンプレートをリスト表示します。

fteDisplayVersion

fteDisplayVersion コマンドを使用して、インストールした Managed File Transfer のバージョンを表示します。

目的

問題判別に役立つために、IBM サービス担当員から **fteDisplayVersion** コマンドの実行を依頼される場合があります。

構文

fteDisplayVersion

```
▶▶ fteDisplayVersion -v
```

Parameters

-v

オプション。製品のバージョンに関する詳細情報を表示します。

-v パラメーターを指定したときに表示される詳細情報は、製品のリリースによって異なる場合があります。 **fteDisplayVersion -v** コマンドの出力として表示される特定の情報には依存しないようにすることをお勧めします。

```
z/OS V 9.0.3
```

z/OS では、**-v** は **productId** プロパティの値を表示します (製品 ID が指定されている場合)。

-? または -h

オプション。コマンド構文を表示します。

パラメーターを指定しない例

この例では、パラメーターなしで **fteDisplayVersion** コマンドを指定します。

```
fteDisplayVersion
```

このコマンドの出力には、製品のバージョン・レベルが示されます。例えば、IBM MQ 9.0 の場合の出力は以下ようになります。

```
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
```

-v パラメーターを指定した場合の例

この例では、**-v** パラメーターを付けて **fteDisplayVersion** コマンドを指定します。

```
fteDisplayVersion -v
```

このコマンドの出力には、製品バージョンに関する詳細情報が含まれます。以下に例を示します。

```
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
```

```
Product: J9CL - 20160427_301573
Configuration: C:\Program Files\IBM\MQ_1
Configuration: C:\ProgramData\IBM\MQ\mqft
```

```
Name: IBM MQ JMS Provider
Version: 9.0.0.0
Level: p900-L160512.4
```

```
Name: IBM MQLight Service for Bluemix JMS Provider
Version:
Level: p900-L160512.4
```

```
Name: Common Services for Java Platform, Standard Edition
Version: 9.0.0.0
Level: p900-L160512.4
```

```
Name: Java Message Service Client
Version: 9.0.0.0
Level: p900-L160512.4
```

```
Name: IBM MQ classes for Java Message Service
Version: 9.0.0.0
Level: p900-L160512.4
```

```
Name: IBM MQ classes for Java
Version: 9.0.0.0
Level: p900-L160512.4
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

fteListAgents: 調整キュー・マネージャーの MFT エージェントのリスト

fteListAgents コマンドは、特定の調整キュー・マネージャーに登録されているすべての Managed File Transfer のエージェントをリストするために使用します。

目的

fteListAgents コマンドは、調整キュー・マネージャーに接続できる任意のシステムから実行できます。標準出力装置 (STDOUT) に、各エージェントの以下の詳細が送信されます。

- エージェント名
- エージェント・キュー・マネージャー
- エージェントがプロトコル・ブリッジ・エージェントの場合は、エージェント名に **bridge** が付加されます。
- エージェントが Connect:Direct ブリッジ・エージェントの場合、エージェント名の後ろに (Connect:Direct bridge) が付きます。
- エージェント状況

このコマンドは、`coordination.properties` ファイルを使用して、調整キュー・マネージャーに接続します。詳しくは、[540 ページの『MFT coordination.properties ファイル』](#)を参照してください。

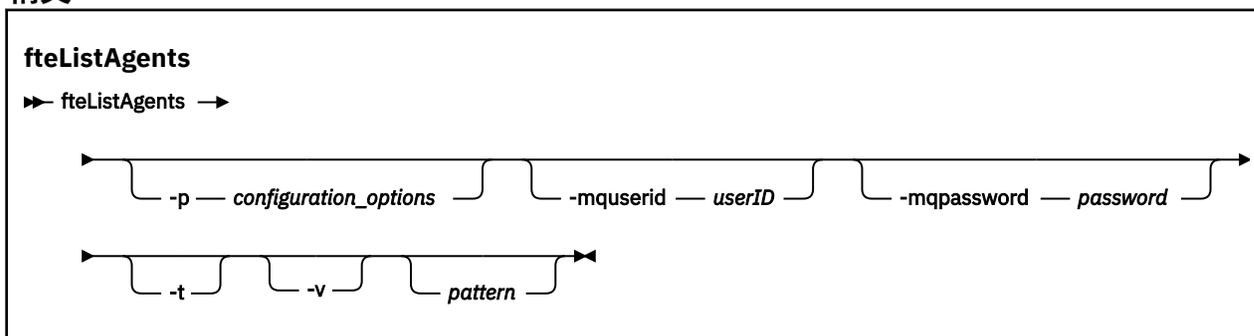
オプションの **-p** パラメーターは、デフォルト・セットと異なる構成オプションのセットを使用する場合にのみこのコマンドに指定します。詳しくは、[57 ページの『MFT 構成オプション \(Multiplatforms\)』](#)を参照してください。

エージェントが **fteListAgents** コマンドによってリストされない場合は、トピック「[エージェントが fteListAgents コマンドによってリストされない場合](#)」の診断フローチャートを使用して、問題を見つけて修正してください。

エージェントの状況情報

このコマンドで生成されるエージェント状況情報は、エージェントが SYSTEM.FTE トピックにパブリッシュする状況メッセージから生成されます。これらのメッセージについては、734 ページの『MFT エージェント状況メッセージ・フォーマット』のトピックで説明しています。 **fteListAgents** コマンドで生成される状況情報を調べれば、最後の状況メッセージがパブリッシュされた時点でのエージェントの状況を確認できます。それらの状況メッセージの頻度は、agentStatusPublishRateLimit プロパティの値によって決まります。このプロパティについて詳しくは、550 ページの『MFT agent.properties ファイル』のトピックを参照してください。

構文



Parameters

-p (*configuration_options*)

オプション。このパラメーターは、エージェントをリストする要求を発行するために使用される構成オプションのセットを決定します。このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティ・ファイルのセットを使用します。

このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。

-mquserid (*userID*)

オプション。調整キュー・マネージャーで認証するユーザー ID を指定します。

-mqpassword (*password*)

オプション。調整キュー・マネージャーで認証するパスワードを指定します。 **-mquserid** パラメーターも一緒に指定する必要があります。 **-mquserid** を指定した場合に **-mqpassword** を指定しないと、関連付けるパスワードを入力するというプロンプトが出されます。パスワードは表示されません。

-v

オプション。冗長モードを指定します。冗長モードでは、Source/Destination の形式の現在の転送数を含む、各エージェントの追加出力が生成されます。ここで、Source はソース転送の現在の数、Destination は宛先転送の現在の数です。

現在の転送情報は、『734 ページの『MFT エージェント状況メッセージ・フォーマット』』のトピックで説明されている、エージェント状況のパブリケーションから取得されます。その結果、この転送情報は、agentStatusPublishRateLimit エージェント・プロパティ値 (デフォルトで 30 秒に設定) の設定内のみで正確です。

-t

オプション。簡潔モードを指定します。IBM MQ 9.0.2 では、**Status Age** 列がデフォルトで出力に含まれます。 **Status Age** 情報を表示しないようにするには、 **-t** パラメーターを指定してコマンドを発行し、その列を非表示にします。詳しくは、 **fteListAgents** コマンドでエージェント状況 UNKNOWN が表示される場合の対応策を参照してください。

pattern

オプション。Managed File Transfer エージェントのリストをフィルターに掛けるために使用するパターン。このパターンはエージェント名に対して突き合わせられます。アスタリスク (*) 文字はワイルドカードとして解釈されます。これは、ゼロ文字を含むあらゆる値と一致します。

UNIX および Linux システムでは、アスタリスク (*) や番号記号 (#) などの特殊文字をリテラルとして扱うには、それらを単一引用符 (') または二重引用符 (") でエスケープする必要があります。これらの文字をエスケープしない場合、これらは特定の UNIX または Linux システムでの意味に従って解釈されます。

このパラメーターを指定しない場合、調整キュー・マネージャーに登録されているすべてのエージェントがリストされます。

-? または -h

オプション。コマンド構文を表示します。

例

この例では、構成オプションで詳細が指定されているキュー・マネージャーに登録されているすべてのエージェントのうち、名前が B で始まるものがリストされます。

```
fteListAgents "B*"
```

この例では、調整キュー・マネージャー QM_EUROPE (デフォルト以外の調整キュー・マネージャー) に登録されているエージェントのリストを冗長モードで表示します。

```
fteListAgents -p QM_EUROPE -v
```

このコマンドの出力は次のようになります。

Agent Name:	Queue Manager Name:	Transfers: (Source/Destination)	Status:
BERLIN	QM_BERLIN	7/0	RUNNING
LONDON	QM_LONDON	0/0	RUNNING
MADRID	QM_MADRID	0/1	UNREACHABLE

有効なエージェント状況値とそれぞれの意味をまとめたリストについては、[624 ページの『MFT エージェントの状況値』](#)のトピックを参照してください。

この例では、調整キュー・マネージャーに登録されているエージェントで、かつ、名前が BRIDGE で始まるすべてのエージェントが冗長モードでリストされます。

```
fteListAgents -v "BRIDGE*"
```

このコマンドの出力は次のようになります。

```
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
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

関連タスク

212 ページの『MFT エージェントのリスト』

特定のキュー・マネージャーに登録された Managed File Transfer エージェントは、コマンド行または IBM MQ Explorer を使用してリストできます。

関連資料

624 ページの『MFT エージェントの状況値』

fteListAgents と **fteShowAgentDetails** のいずれかのコマンドを実行すると、エージェント状況情報が生成されます。この状況として生成される可能性がある値は、以下のとおりです。

518 ページの『fteShowAgentDetails』

fteShowAgentDetails コマンドは、特定の Managed File Transfer Agent の詳細を表示するために使用します。これらは、エージェントの Managed File Transfer 調整キュー・マネージャーによって保管される詳細です。

fteListMonitors: MFT リソース・モニターのリスト

コマンド行から **fteListMonitors** コマンドを使用すると、Managed File Transfer ネットワークの既存のリソース・モニターをすべてリストできます。

目的

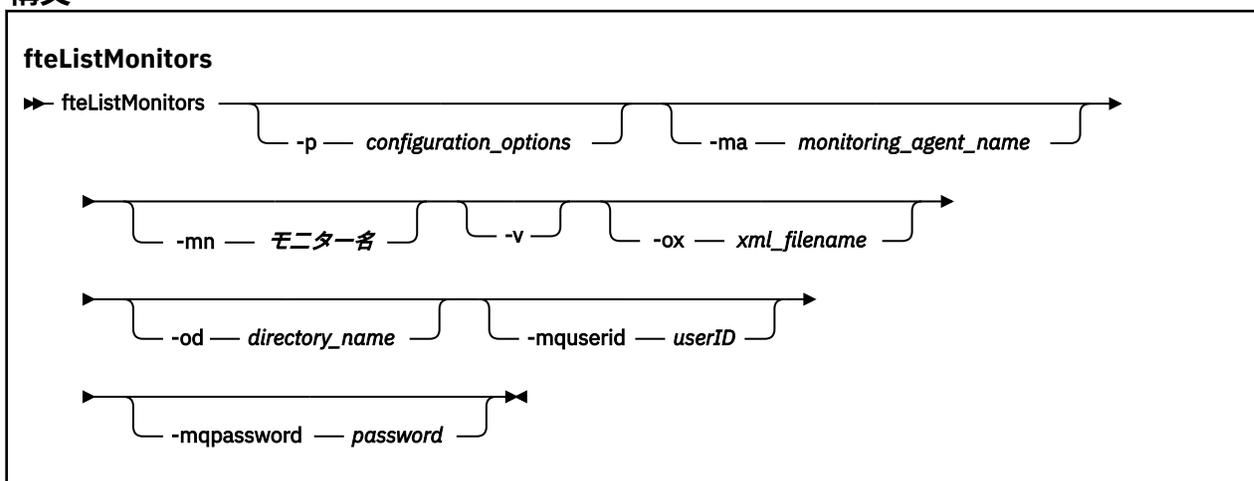
fteListMonitors コマンドは、既存のリソース・モニターをリストします。エージェント名およびリソース・モニター名を指定すれば、このコマンドの出力をフィルターに掛けることができます。

このコマンドは、`coordination.properties` ファイルを使用して、調整キュー・マネージャーに接続します。詳しくは、540 ページの『MFT `coordination.properties` ファイル』を参照してください。

-ox パラメーターを使用して、リソース・モニターを XML ファイルにエクスポートすることができます。この XML ファイルの詳しい使用方法については、413 ページの『fteCreateMonitor: MFT リソース・モニターの作成』を参照してください。

オプションの **-p** パラメーターは、デフォルト・セットと異なる構成オプションのセットを使用する場合にのみこのコマンドに指定します。詳しくは、[構成オプション](#)を参照してください。

構文



Parameters

-p (configuration_options)

オプション。このパラメーターは、転送を取り消すために使用する構成オプションのセットを決定します。慣例として、このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティ・ファイルのセットを使用します。

このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。

-ma (monitoring_agent_name)

オプション。入力データとして提供したパターンを使用して、エージェント名でリソース・モニターをフィルタリングします。アスタリスク (*) 文字は、ゼロ以上の文字と一致するワイルドカードとして解釈されます。-ma パラメーターを指定しない場合、デフォルトの調整キュー・マネージャーの全エージェントに関連付けられたリソース・モニターのすべてが、デフォルトでリストされます。

-mn (monitor_name)

オプション。入力データとして提供したパターンを使用して、モニター名でリソース・モニターをフィルタリングします。アスタリスク (*) 文字は、ゼロ以上の文字と一致するワイルドカードとして解釈されます。-mn パラメーターを指定しない場合、デフォルトの調整キュー・マネージャーの全エージェントに関連付けられたリソース・モニターのすべてが、デフォルトでリストされます。

-mquserid (userID)

オプション。調整キュー・マネージャーで認証するユーザー ID を指定します。

-mqpassword (password)

オプション。調整キュー・マネージャーで認証するパスワードを指定します。-mquserid パラメーターも一緒に指定する必要があります。-mquserid を指定した場合に -mqpassword を指定しないと、関連付けるパスワードを入力するというプロンプトが出されます。パスワードは表示されません。

-v

オプション。モニター状況の追加情報を含む詳細出力を生成します。ここでは、モニターが開始されているか、または停止されているか、モニターされているディレクトリー・リソース・パス、およびトリガー条件などが含まれます。

-ox (xml_filename)

オプション。このパラメーターを指定する場合は、-ma パラメーターと -mn パラメーターも一緒に組み合わせて指定する必要があります。リソース・モニターを XML ファイルにエクスポートします。その XML ファイルを **fteCreateMonitor** コマンドと **-ix** パラメーターで使用できます。

V 9.0.5

-ox パラメーターを -od パラメーターと組み合わせることはできません。

V 9.0.5

-od (directory_name)

オプション。複数のリソース・モニター定義を指定のディレクトリーにエクスポートします。各リソース・モニター定義は、agent name.monitor name.xml という形式の名前で別々の XML ファイルに保存されます。XML ファイルを保管する有効なターゲット・ディレクトリーを指定する必要があります。そうしないと、エラー・メッセージが表示されます。このパラメーターを -ox パラメーターと組み合わせることはできません。

-? または -h

オプション。コマンド構文を表示します。

例: リソース・モニターのリストを表示する

この例では、モニター・エージェント (およびモニターに関連付けられたファイル転送のソース・エージェント) AGENT1 に関連付けられたすべてのリソース・モニターがリストされます。

```
fteListMonitors -ma AGENT1
```

例: 1つのリソース・モニターを XML ファイルにエクスポートする

この例では、AGENT1 にある 1つのリソース・モニター MONITOR1 を XML ファイル filename1.xml にエクスポートします。そのために、-ox パラメーターで XML ファイルの名前を指定します。

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

例: 指定のディレクトリーに1つのリソース・モニターをエクスポートする

V 9.0.5

この例では、AGENT1にある1つのリソース・モニター MONITOR1 を、**-od** パラメーターで指定するディレクトリーにエクスポートします。この例は **-ox** パラメーターの使用例に似ています。違うのは XML ファイル名のフォーマットです。

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

例: 指定のディレクトリーにある XML ファイルにリソース・モニターのバッチをエクスポートする

V 9.0.5

以下のすべての例では、**-od** パラメーターで指定するディレクトリーに複数のリソース・モニターをエクスポートします。各リソース・モニター定義が別々の XML ファイルに保存され、それぞれのファイル名が *agent name.monitor name.xml* という形式になります。

この例では、すべてのリソース・モニターを指定のディレクトリーにエクスポートします。

```
fteListMonitors -od /usr/mft/resmonbackup
```

この例では、AGENT1にあるすべてのリソース・モニターを指定のディレクトリーにエクスポートします。

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

ワイルドカード・マッチングを使用して、エクスポートするリソース・モニターを定義できます。アスタリスク文字 (*) を使用して、エージェント名とモニター名のいずれかまたは両方と突き合わせるパターンを指定できます。

この例では、AGENT1にある MON* というパターンに合致する名前のすべてのリソース・モニターを指定のディレクトリーにエクスポートします。

```
fteListMonitors -ma AGENT1 -mn MON* -od /usr/mft/resmonbackup
```

この例では、AGEN* というパターンに合致する名前のエージェントにあるすべてのリソース・モニターを指定のディレクトリーにエクスポートします。

```
fteListMonitors -ma AGEN* -od /usr/mft/resmonbackup
```

この例では、AGENT* というパターンに合致する名前のエージェントにある MON* というパターンに合致する名前のすべてのリソース・モニターを指定のディレクトリーにエクスポートします。

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

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

関連タスク

[170 ページの『MFT リソースのモニター』](#)

キューやディレクトリーなどの Managed File Transfer リソースをモニターできます。そのリソースで条件が満たされると、リソース・モニターがファイル転送などのタスクを開始します。**fteCreateMonitor** コマンドを使用するか、IBM MQ Explorer に対応した Managed File Transfer プラグインの「モニター」ビューを使用して、リソース・モニターを作成できます。

[193 ページの『MFT リソース・モニターのバックアップとリストア』](#)

リソース・モニターをバックアップしておけば、後で使用することができます。そのためには、リソース・モニターの定義を XML ファイルにエクスポートします。エクスポートした定義をインポートすれば、バックアップから新しいリソース・モニターを作成できます。

関連資料

413 ページの『[fteCreateMonitor: MFT リソース・モニターの作成](#)』

fteCreateMonitor コマンドは、コマンド行から新規リソース・モニターを作成して開始します。Managed File Transfer を使用してリソース (例えば、ディレクトリーの内容) をモニターして、トリガー条件が満たされたときに、指定したタスク (ファイル転送など) を開始することができます。

466 ページの『[fteDeleteMonitor: MFT リソース・モニターの削除](#)』

コマンド行で Managed File Transfer の既存のリソース・モニターを停止して削除する場合には、**fteDeleteMonitor** コマンドを使用します。リソース・モニター・エージェントに対してこのコマンドを発行します。

fteListScheduledTransfers

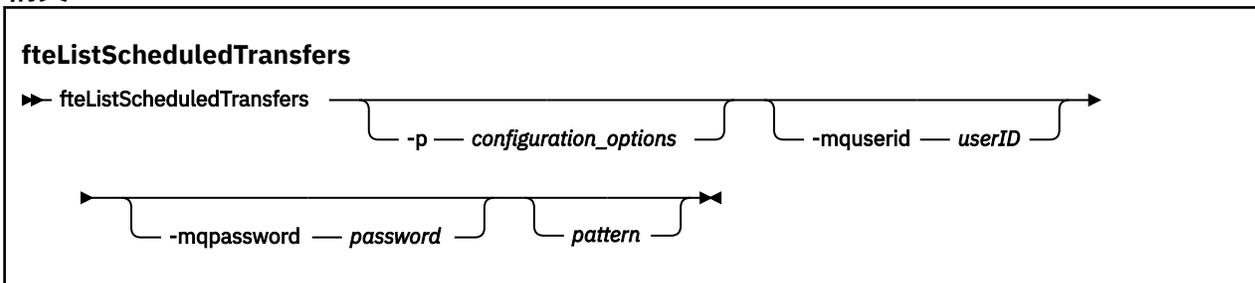
fteListScheduledTransfers コマンドは、コマンド行または IBM MQ Explorer を使用して以前に作成した Managed File Transfer のすべての転送をリストするために使用します。

目的

すべてのスケジュール済み転送は、ソース・エージェント名に基づいて、または調整キュー・マネージャーに基づいてリストできます。

オプションの **-p** パラメーターは、デフォルトと異なる構成オプションを使用する場合にのみこのコマンドに指定します。**-p** を指定しない場合、`installation.properties` で定義された構成オプションが使用されます。詳しくは、『57 ページの『[MFT 構成オプション \(Multiplatforms\)](#)』』を参照してください。

構文



Parameters

-p (configuration_options)

オプション。複数の調整キュー・マネージャーがある場合は、このパラメーターを使用して、スケジュール済み転送をリストするエージェントを明示的に指定します。このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられた構成オプションを使用します。

このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションが使用されます。

-mquserid (userID)

オプション。調整キュー・マネージャーで認証するユーザー ID を指定します。

-mqpassword (password)

オプション。調整キュー・マネージャーで認証するパスワードを指定します。**-mquserid** パラメーターも一緒に指定する必要があります。**-mquserid** を指定した場合に **-mqpassword** を指定しないと、関連付けるパスワードを入力するというプロンプトが出されます。パスワードは表示されません。

pattern

オプション。Managed File Transfer のスケジュール済み転送のリストをフィルターに掛けるために使用するパターン。このパターンはソース・エージェント名に対して突き合わせられます。アスタリスク (*) 文字は、ゼロ以上の文字と一致するワイルドカードとして解釈されます。

このパラメーターを指定しない場合、デフォルトで調整キュー・マネージャーに登録されているすべてのスケジュール済み転送がリストされます。

-? または -h

オプション。コマンド構文を表示します。

例

この例では、パターン *2 に一致するソース・エージェントが含まれるすべてのスケジュール済み転送がリストされます。

```
fteListScheduledTransfers "*2"
```

このコマンド例では、以下の出力が生成されます。スケジュールの開始時刻と次の転送時刻は協定世界時 (UTC) で表示されます。

```
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
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

関連タスク

[163 ページの『スケジュール済みファイル転送の作成』](#)

IBM MQ Explorer エクスプローラーまたはコマンド行のいずれかを使用して新規ファイル転送をスケジュールに入れられます。スケジュール済みの転送には、単一のファイルまたは1つのグループの複数のファイルを含めることができます。スケジュール済みファイル転送は、1回実行することも複数回転送を繰り返すこともできます。

関連資料

[468 ページの『fteDeleteScheduledTransfer』](#)

fteListTemplates

fteListTemplates コマンドを使用して、調整キュー・マネージャーで使用できる Managed File Transfer の転送テンプレートをリスト表示します。

目的

このコマンドでは、すべてのテンプレート名、またはフィルターで選択したテンプレート名をリスト表示します。リストの出力フォーマットは、以下のいずれかになります。

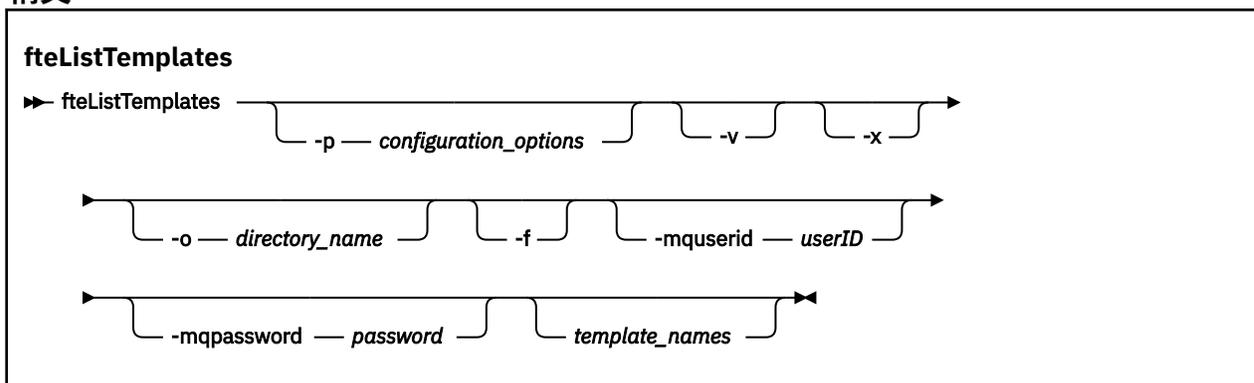
- テンプレート名のみ (デフォルトの動作)

- テンプレート名とテンプレートの要約 (冗長モード)
- テンプレートを記述した完全な XML メッセージ (**-x** パラメーターと **-o** パラメーター)

このコマンドは、`coordination.properties` ファイルを使用して、調整キュー・マネージャーに接続します。詳しくは、540 ページの『MFT `coordination.properties` ファイル』を参照してください。

オプションの **-p** パラメーターは、デフォルト・セットと異なる構成オプションのセットを使用する場合にのみこのコマンドに指定します。詳しくは、[構成オプション](#)を参照してください。

構文



Parameters

-p

オプション。このパラメーターは、テンプレートを削除するために使用する構成オプションのセットを決定します。慣例として、このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティ・ファイルのセットを使用します。

このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。

-v

オプション。冗長モードを指定して、一致する各テンプレートの短い要約を生成します。**-x** パラメーターも指定すると、このパラメーターは無視されます。

-v パラメーターを指定すると、各テンプレートの要約が記述されます。以下に例を示します。

```

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

```

-v パラメーターを指定しない場合は、デフォルトの出力モードとして、マッチングするテンプレート名がリスト表示されます。

-x

オプション。マッチングする各テンプレートの XML 形式のメッセージを表示します。このパラメーターは、**-o** パラメーターを指定した場合以外は無視されます。

-o (directory_name)

オプション。指定したディレクトリーにあるファイルに XML 形式のメッセージを送信します。テンプレートごとに 1 つのファイルが作成され、各ファイルの名前は、テンプレートと同じ名前に `.xml` という接尾部を付けた形式になります。このパラメーターは、**-x** パラメーターを指定した場合以外は無視されます。

-f

オプション。既存のすべての出力ファイルを強制的に上書きします。このパラメーターは、**-o** パラメーターを指定した場合以外は無視されます。**-f** を指定しないで、既存の出力ファイルの名前を指定した場合は、エラーを報告して処理を続行する、というのがデフォルトの動作になります。

-mquserid (userID)

オプション。調整キュー・マネージャーで認証するユーザー ID を指定します。

-mqpassword (password)

オプション。調整キュー・マネージャーで認証するパスワードを指定します。**-mquserid** パラメーターも一緒に指定する必要があります。**-mquserid** を指定した場合に **-mqpassword** を指定しないと、関連付けるパスワードを入力するというプロンプトが出されます。パスワードは表示されません。

(template_names)

オプション。表示する 1 つ以上のテンプレート名のリスト。テンプレート名では、ゼロ個以上の文字にマッチングするワイルドカードとしてアスタリスクを使用できます。オペレーティング・システムによっては、シェルによる展開を回避するために、ワイルドカード文字を組み込んだテンプレート名を二重引用符 (" ") または単一引用符 (') で囲まなければならない場合があります。シェルによる展開は、予期しない動作の原因になる可能性があります。

`template_names` に何も指定しない場合は、デフォルトですべてのテンプレートのリストが表示されます。

-? または -h

オプション。コマンド構文を表示します。

例

この例では、名前が ST で始まるすべてのテンプレートをリスト表示します。

```
fteListTemplates "ST*"
```

この例では、STANDBY テンプレートを XML 形式のメッセージとして現行ディレクトリーの `STANDBY.xml` ファイルに作成します。

```
fteListTemplates -x -o . STANDBY
```

このコマンドで作成される `STANDBY.xml` の出力は、以下のとおりです。

```
<?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>
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

fteMigrateAgent: FTE 7.0 エージェントの MQ 7.5 以降へのマイグレーション

既存のエージェントとそれに関連する構成を IBM WebSphere MQ File Transfer Edition 7.0 から IBM WebSphere MQ 7.5 以降にマイグレーションする場合は、**fteMigrateAgent** コマンドを使用してマイグレーションします。このコマンドを使用して、標準エージェント、Connect:Direct エージェント、またはプロトコル・ブリッジ・エージェントをマイグレーションすることができます。また、このコマンドでは、複数のエージェントを単一の要求でマイグレーションすることも可能です。

V 9.0.0 IBM MQ 9.0 から、Managed File Transfer は Web エージェントをサポートしなくなりました。**fteMigrateAgent** コマンドを使用して Web エージェントを以前のリリースから IBM MQ 9.0 にマイグレーションしようとする、Web エージェントのマイグレーションがサポートされていないことを示すエラー・メッセージが表示されます。

注：IBM WebSphere MQ File Transfer Edition 7.0 以降からマイグレーションしていて、FTE_CONFIG 環境変数を引き続き使用しようとしている場合には、FTE_CONFIG の値を変更せずにマイグレーションすることができます。標準マイグレーションを実行できませんが、BFG_DATA を設定してはならず、FTE_CONFIG を IBM WebSphere MQ 7.0 で使用されていたとおりに設定する必要があります。

重要： **ULW** IBM MQ for UNIX, Linux, and Windows では、IBM MQ 管理者 (および mqm グループのメンバー) であるユーザーのみが、このコマンドを実行できます。IBM MQ 管理者以外のユーザーとしてこのコマンドを実行しようすると、エラー・メッセージ BFGCL0502E: 要求された操作を実行する権限がありません。を受け取ります。コマンドは実行されません。

z/OS z/OS システムでは、マイグレーション・コマンドを実行するためには、ユーザーは以下の条件の (少なくとも) 1 つを満たす必要があります。

- mqm グループのメンバーである (システムに mqm グループが定義されている場合)。
- V 9.0.0.1** BFG_GROUP_NAME 環境変数で名前が指定されたグループのメンバーである (名前が指定されている場合)。
- V 9.0.0.1** コマンド実行時に BFG_GROUP_NAME 環境変数に値が設定されていない。

エージェントが Windows サービスとして実行されるように構成されている場合は、**fteModifyAgent** コマンドを使用して、Windows サービスでなくなるようにエージェントを再構成します。マイグレーションが完了した後、再度 **fteModifyAgent** コマンドを使用して、新しいエージェントが Windows サービスになるように構成します。代わりに -f パラメーター を含めると、コマンドは実行されますが、警告を出します。

fteMigrateAgent コマンドを実行する前に、**fteStopAgent** コマンドを使用して、マイグレーションするエージェントを停止する必要があります。

-f パラメーター を指定してこのコマンドを実行すると、そのエージェントに関する情報のみが最新表示されます。必要なファイルが見つからない場合、コマンドは失敗します。

具体的には、以下のプロパティ・ファイル、XML ファイル、およびそのエージェントに関連付けられているディレクトリーがマイグレーションされます。

表 37. <i>fteMigrateAgent</i> コマンドによってマイグレーションされるエージェント・ファイル	
fteMigrateAgent コマンドによってマイグレーションされる各エージェントのファイル名	情報
wmqfte.properties	wmqfte.properties ファイルの名前が、IBM WebSphere MQ 7.5 以降で installation.properties に変更されました。
command.properties	
coordination.properties	
coordination_queue_manager.mqsc	
agent_name_create.mqsc	
agent_name_delete.mqsc	
exits ディレクトリー	このコマンドは exits ディレクトリー内のすべてのファイルをコピーします。
標準エージェントにのみ適用:	
UserSandboxes.xml	
Connect:Direct ブリッジ・エージェントにのみ適用:	
ConnectDirectCredentials.xml	
ConnectDirectNodeProperties.xml	
ConnectDirectProcessDefinitions.xml	
プロトコル・ブリッジ・エージェントにのみ適用:	
ProtocolBridgeCredentials.xml	
ProtocolBridgeProperties.xml	このファイルは、IBM WebSphere MQ File Transfer Edition 7.0.4 以降の場合のみ存在します。

fteMigrateAgent コマンドは、インストール・キュー・マネージャー、調整キュー・マネージャー、およびコマンド・キュー・マネージャーのファイルをマイグレーションし、それらのファイルが IBM WebSphere MQ 7.5 以降にまだ存在しない場合は IBM WebSphere MQ 7.5 以降にコピーします。それらのファイルが既に存在している場合、それらはこのコマンドでコピーされません。

構文

<p>fteMigrateAgent</p> <p>▶ fteMigrateAgent — -agentName — <i>agent_name</i> — -config — <i>configuration_directory</i> →</p> <p>▶ — -credentialPath — <i>credentials_path</i> — — -f — — -p — <i>configuration_options</i> —▶</p>

Parameters

-agentName *agent_name*

必須。IBM WebSphere MQ 7.5 以降にマイグレーションするエージェントの名前。

-config configuration_directory

必須。エージェントのマイグレーション元となるインストール済み環境の構成ディレクトリーのパス。C:\Documents and Settings\All Users\Application Data\IBM\WMQFTE\config に例を示します

-credentialPath credentials_path

必須。資格情報のマイグレーション先の場所を定義します。このパラメーターは、既存の資格情報ファイルがあるディレクトリー・パスとするか、または新しい資格情報ファイルを受け取る新しい場所とすることができます。z/OS プラットフォームでは、これは前もって存在する区分データ・セット拡張 (PDSE) とすることができます。そこでは、既存のメンバーを更新することになるか、または既存のメンバーはなく、資格情報のための新規メンバーを組み込むことになります。

注：PDSE を使用する場合、それは可変ブロックである必要があります。

-f

オプション。通常マイグレーションされる構成ファイルの一部が既存の構成と競合する場合でも、エージェントを強制的にマイグレーションします。例えば、Managed File Transfer 上のプロパティール・ファイルと IBM WebSphere MQ 7.5 以降のプロパティール・ファイルの間に不一致がある場合、**-f** パラメーターを指定すると、この不一致は無視されます。

-p configuration_options

オプション。このパラメーターは、マイグレーション対象となる構成の位置指定に使用される構成オプションのセットを決定します。構成オプションのセット名を **-p** パラメーターの値として使用します。慣例により、これは調整キュー・マネージャーの名前です。このパラメーターを指定しない場合、デフォルトの構成オプションのセットが使用されます。詳しくは、57 ページの『MFT 構成オプション (Multiplatforms)』を参照してください。

-? または -h

オプション。コマンド構文を表示します。

例

次の例では、AGENT3 と /var/ibm/WMQFTE/config にあるその構成が、IBM WebSphere MQ 7.5 またはそれ以降にマイグレーションされます。

```
fteMigrateAgent -agentName AGENT3 -config /var/ibm/WMQFTE/config -credentialPath /home/user1/AGENT3
```

この例では、C:\Documents and Settings\All Users\Application Data\IBM\WMQFTE\config 内のすべてのエージェントとその構成が IBM WebSphere MQ 7.5 またはそれ以降にマイグレーションされます。Windows ファイル・パスは、二重引用符 (") で囲まれます。**-f** パラメーターは、マイグレーションを強制し、プロパティール・ファイルの不一致を無視するために指定されます。

```
fteMigrateAgent -agentName "*" -config "C:\Documents and Settings\All Users\Application Data\IBM\WMQFTE\config" -credentialPath "C:\Documents and Settings\user1\AGENT3" -p "configurationOption" -f
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

戻りコードについて詳しくは、338 ページの『MFT の戻りコード』を参照してください。

fteMigrateConfigurationOptions: FTE 7.0 構成の MQ 7.5 以降へのマイグレーション

fteMigrateConfigurationOptions コマンドは、IBM WebSphere MQ File Transfer Edition 7.0 の構成オプションのセットをマイグレーションして、IBM WebSphere MQ 7.5 以降にコピーします (それらのファ

イルがターゲット・バージョンにまだ存在していない場合)。それらのファイルが既存の場合、メッセージが出力され、コマンドは続行されません。

注：IBM WebSphere MQ File Transfer Edition 7.0 からマイグレーションして、FTE_CONFIG 環境変数を引き続き使用しようとしている場合には、FTE_CONFIG の値を変更せずにマイグレーションすることができます。標準マイグレーションを実行できますが、BFG_DATA を設定してはならず、FTE_CONFIG を IBM WebSphere MQ File Transfer Edition 7.0 で使用されていたとおりに設定する必要があります。

重要： **ULW** IBM MQ for UNIX, Linux, and Windows では、IBM MQ 管理者 (および mqm グループのメンバー) であるユーザーのみが、このコマンドを実行できます。IBM MQ 管理者以外のユーザーとしてこのコマンドを実行しようとすると、エラー・メッセージ BFGCL0502E: 要求された操作を実行する権限がありません。を受け取ります。コマンドは実行されません。

z/OS z/OS システムでは、マイグレーション・コマンドを実行するためには、ユーザーは以下の条件の (少なくとも) 1 つを満たす必要があります。

- mqm グループのメンバーである (システムに mqm グループが定義されている場合)。
- **V9.0.0.1** BFG_GROUP_NAME 環境変数で名前が指定されたグループのメンバーである (名前が指定されている場合)。
- **V9.0.0.1** コマンド実行時に BFG_GROUP_NAME 環境変数に値が設定されていない。

構文

fteMigrateConfigurationOptions

```
► fteMigrateConfigurationOptions — -config — configuration_directory — -credentialPath ►  
  
◄ credentials_path — -configurationOptionsName — configuration_options_name ◄
```

Parameters

-config (configuration_directory)

必須。マイグレーション元となるインストール済み環境の構成ディレクトリーのパス。

C:\Documents and Settings\All Users\Application Data\IBM\WMQFTE\config に例を示します

-credentialPath (credentials_path)

必須。資格情報のマイグレーション先の場所を定義します。このパラメーターは、既存の資格情報ファイルがあるディレクトリー・パスとするか、または新しい資格情報ファイルを受け取る新しい場所とすることができます。

このパラメーターは、agent.properties ファイル、coordination.properties ファイル、および command.properties ファイルにある SSL/TLS 鍵ストア・プロパティーおよびトラストストア・プロパティーのパスワード・プロパティーを、IBM WebSphere MQ 7.5 より前のバージョンの製品から 7.5 以降にマイグレーションするために使用されます。

IBM WebSphere MQ 7.5 より前のバージョンでは、パスワード・プロパティー (**coordinationSslTrustStorePassword** など) は coordination.properties ファイルに存在していましたが、IBM WebSphere MQ 7.5 内の MQMFTCredentials.xml ファイルに移動されました。パスワード・プロパティーを coordination.properties ファイルから MQMFTCredentials.xml ファイルに移動するために、**-credentialsPath** を使用します。このオプションを使用して **fteMigrateConfigurationOptions** コマンドを実行すると、coordination.properties ファイルで coordinationSslTrustStorePassword が検索され、そのプロパティーが存在すると、MQMFTCredentials.xml ファイルにマイグレーションされます。

z/OS z/OS プラットフォームでは、これは前もって存在する区分データ・セット拡張 (PDSE) とすることができます。そこでは、既存のメンバーを更新することになるか、または既存のメンバーはなく、資格情報のための新規メンバーを組み込むこととなります。

注：PDSE を使用する場合、それは可変ブロックである必要があります。

-configurationOptionsName (configuration_options_name)

必須。マイグレーションする構成オプション・セットの名前。ゼロ個以上の文字を表すアスタリスク文字 (*) を使用して、複数の構成オプション・セットをマイグレーションすることができます。アスタリスクはストリングと共に使用できます。例えば、名前が IBM で始まるすべての構成オプション・セットをマイグレーションするには、このパラメーターを `-configurationOptionsName IBM*` のように使用します。

例

次の例では、ディレクトリー `C:\Documents and Settings\All Users\Application Data\IBM\WMQFTE\config` ディレクトリー内にあるすべての構成がマイグレーションされます。ディレクトリー・パスは二重引用符で囲まれています。

```
fteMigrateConfigurationOptions -config "C:\Documents and Settings\All Users\Application Data\IBM\WMQFTE\config" -credentialPath "C:\Documents and Settings\user1\configurationoptions" -configurationOptionsName *
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

関連資料

13 ページの『[MFT のマイグレーションの考慮事項](#)』

Managed File Transfer のマイグレーションを計画している場合は、以下の情報を確認してください。

483 ページの『[fteMigrateAgent: FTE 7.0 エージェントの MQ 7.5 以降へのマイグレーション](#)』

既存のエージェントとそれに関連する構成を IBM WebSphere MQ File Transfer Edition 7.0 から IBM WebSphere MQ 7.5 以降にマイグレーションする場合は、**fteMigrateAgent** コマンドを使用してマイグレーションします。このコマンドを使用して、標準エージェント、Connect:Direct エージェント、またはプロトコル・ブリッジ・エージェントをマイグレーションすることができます。また、このコマンドでは、複数のエージェントを単一の要求でマイグレーションすることも可能です。

487 ページの『[fteMigrate ロガー: FTE 7.0 データベース・ロガーの MQ 7.5 以降へのマイグレーション](#)』

既存のスタンドアロン・データベース・ロガーの構成を IBM WebSphere MQ File Transfer Edition 7.0.1 以降から IBM WebSphere MQ 7.5 以降にマイグレーションする場合は、**fteMigrateLogger** コマンドを使用します。

fteMigrate ロガー: FTE 7.0 データベース・ロガーの MQ 7.5 以降へのマイグレーション

既存のスタンドアロン・データベース・ロガーの構成を IBM WebSphere MQ File Transfer Edition 7.0.1 以降から IBM WebSphere MQ 7.5 以降にマイグレーションする場合は、**fteMigrateLogger** コマンドを使用します。

このコマンドを使用して、Java EE データベース・ロガーをマイグレーションすることはできません。代わりに、[Java EE データベース・ロガーのマイグレーション](#)の情報を请使用してください。

注：IBM WebSphere MQ 7.0 以降からマイグレーションしていて、FTE_CONFIG 環境変数を引き続き使用しようとしている場合には、FTE_CONFIG の値を変更せずにマイグレーションすることができます。標準マイグレーションを実行できますが、BFG_DATA を設定してはならず、FTE_CONFIG を IBM WebSphere MQ 7.0 で使用されていたとおりに設定する必要があります。

重要: **ULW** IBM MQ for UNIX, Linux, and Windows では、IBM MQ 管理者 (および mqm グループのメンバー) であるユーザーのみが、このコマンドを実行できます。IBM MQ 管理者以外のユーザーとしてこのコマンドを実行しようとすると、エラー・メッセージ BFGCL0502E: 要求された操作を実行する権限がありません。を受け取ります。コマンドは実行されません。

z/OS z/OS システムでは、マイグレーション・コマンドを実行するためには、ユーザーは以下の条件の (少なくとも) 1 つを満たす必要があります。

- mqm グループのメンバーである (システムに mqm グループが定義されている場合)。
- **V9.0.0.1** BFG_GROUP_NAME 環境変数で名前が指定されたグループのメンバーである (名前が指定されている場合)。
- **V9.0.0.1** コマンド実行時に BFG_GROUP_NAME 環境変数に値が設定されていない。

スタンドアロンのデータベース・ロガーを Windows サービスとして実行するように構成した場合、そのロガーの構成を **fteMigrateLogger** コマンドを使用してマイグレーションすることはできません。Windows サービスとして実行するように構成されたロガーで **fteMigrateLogger** コマンドを実行すると、そのコマンドはエラーを発生し、続行されません。代わりに **-f** パラメーターを含めると、コマンドは実行されますが、警告を出します。

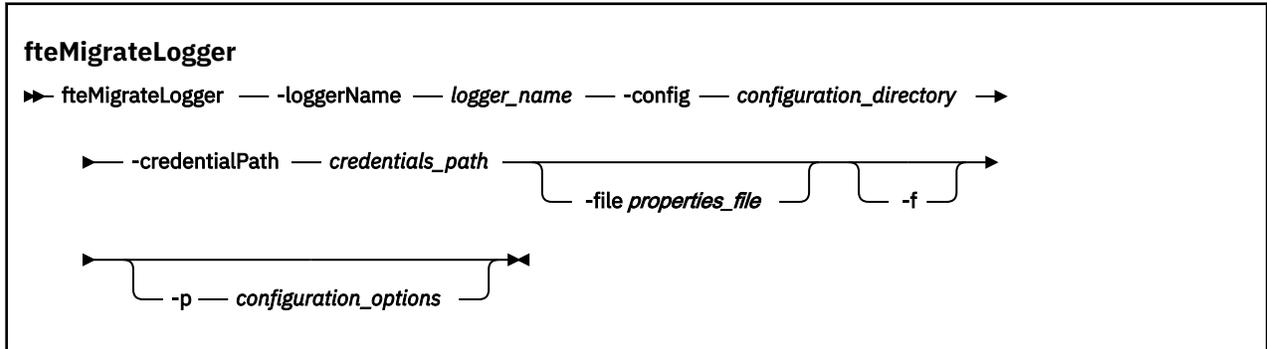
fteMigrateLogger コマンドを実行する前に、IBM WebSphere MQ File Transfer Edition 7.0 のマイグレーションする構成があるデータベース・ロガーを停止します。

-f パラメーターを指定してこのコマンドを実行すると、そのロガーに関する情報のみが最新表示されます。必要なファイルが見つからない場合、コマンドは失敗します。具体的には、ロガー構成に関連付けられている以下のプロパティ・ファイルと **.mqsc** ファイルがマイグレーションされます。

fteMigrateLogger コマンドによってマイグレーションされるファイルの名前	情報
wmqfte.properties	IBM WebSphere MQ 7.5 以降では、wmqfte.properties ファイルのベースになるのは installation.properties です。
command.properties	
coordination.properties	
coordination_queue_manager.mqsc	
databaselogger.properties または -file パラメーターを使用して指定されたその他のプロパティ・ファイル	IBM WebSphere MQ 7.5 以降では、databaselogger.properties を使用して logger.properties ファイルが作成されます。

fteMigrateLogger コマンドは、インストール・キュー・マネージャー、調整キュー・マネージャー、およびコマンド・キュー・マネージャーのファイルをマイグレーションし、それらのファイルがターゲット・バージョンにまだ存在しない場合は IBM WebSphere MQ 7.5 以降にコピーします。それらのファイルが既に存在している場合、それらはこのコマンドでコピーされません。

構文



Parameters

-loggerName *logger_name*

必須。IBM WebSphere MQ 7.5 以降において、マイグレーションされたロガー構成に付ける名前。IBM WebSphere MQ 7.5 の新機能であるロガー名について詳しくは、[logger_name](#) パラメーターを参照してください。

-config *configuration_directory*

必須。ロガー構成のマイグレーション元となっているインストール済み環境の構成ディレクトリーのパス。

-credentialPath *credentials_path*

必須。資格情報のマイグレーション先の場所を定義します。このパラメーターは、既存の資格情報ファイルがあるディレクトリー・パスとするか、または新しい資格情報ファイルを受け取る新しい場所とすることができます。z/OS プラットフォームでは、これは前もって存在する区分データ・セット拡張 (PDSE) とすることができます。そこでは、既存のメンバーを更新することになるか、または既存のメンバーはなく、資格情報のための新規メンバーを組み込むことになります。

注：PDSE を使用する場合、それは可変ブロックである必要があります。

-file *properties_file*

オプション。マイグレーションするデータベース・ロガー・プロパティ・ファイルを指定します。このパラメーターは、プロパティ・ファイルがデフォルトの名前およびパスである *configuration_directory/coordination_qmgr_name/databaselogger.properties* を使用しない場合のみ必須です。

-f

オプション。通常マイグレーションされる構成ファイルの一部が既存の構成と競合する場合でも、強制的にマイグレーションします。例えば、IBM WebSphere MQ File Transfer Edition 上のデータベース・ロガー・プロパティ・ファイルと IBM WebSphere MQ 7.5 以降上のプロパティ・ファイルとの間に不一致がある場合、**-f** パラメーターを指定すると、この不一致は無視されます。

-p *configuration_options*

オプション。このパラメーターは、マイグレーション対象となるロガー構成の位置指定に使用される構成オプションのセットを決定します。構成オプションのセット名を **-p** パラメーターの値として使用します。慣例により、これは調整キュー・マネージャーの名前です。このパラメーターを指定しない場合、デフォルトの構成オプションのセットが使用されます。詳しくは、57 ページの『MFT 構成オプション (Multiplatforms)』を参照してください。

-? または -h

オプション。コマンド構文を表示します。

例

次の例では、`/var/ibm/WMQFTE/config`にあるスタンドアロン・データベース・ロガーの構成が IBM WebSphere MQ 7.5 にマイグレーションされ、`FTELOGGER1`: という名前が付けられます。

```
fteMigrateLogger -loggerName FTELOGGER1 -config /var/ibm/WMQFTE/config  
-credentialPath /home/user1/FTELOGGER1
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

戻りコードについて詳しくは、[338 ページの『MFT の戻りコード』](#)を参照してください。

fteMigrateLogger コマンドの実行後

マイグレーションを検証するには、**fteMigrateLogger** コマンドを正常に実行した後、[530 ページの『fteStartLogger』](#) コマンドを使用して、IBM WebSphere MQ 7.5 以降で構成をマイグレーションしたデータベース・ロガーを開始します。

関連資料

[13 ページの『MFT のマイグレーションの考慮事項』](#)

Managed File Transfer のマイグレーションを計画している場合は、以下の情報を確認してください。

[483 ページの『fteMigrateAgent: FTE 7.0 エージェントの MQ 7.5 以降へのマイグレーション』](#)

既存のエージェントとそれに関連する構成を IBM WebSphere MQ File Transfer Edition 7.0 から IBM WebSphere MQ 7.5 以降にマイグレーションする場合は、**fteMigrateAgent** コマンドを使用してマイグレーションします。このコマンドを使用して、標準エージェント、Connect:Direct エージェント、またはプロトコル・ブリッジ・エージェントをマイグレーションすることができます。また、このコマンドでは、複数のエージェントを単一の要求でマイグレーションすることも可能です。

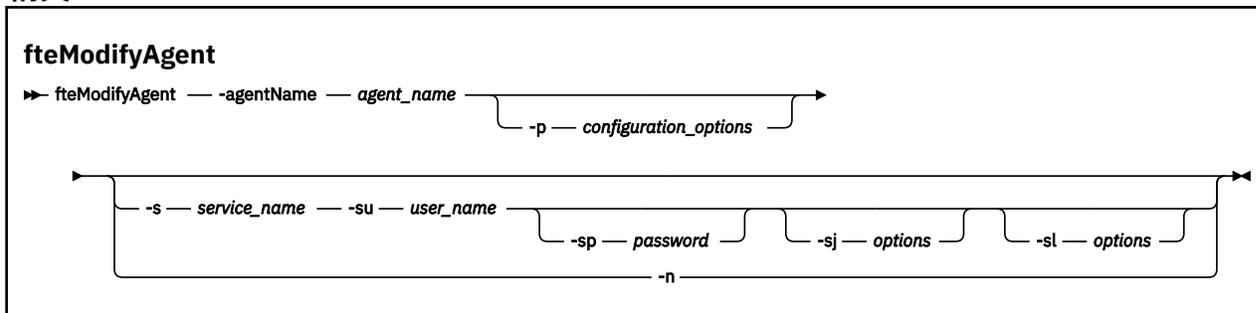
[485 ページの『fteMigrateConfigurationOptions: FTE 7.0 構成の MQ 7.5 以降へのマイグレーション』](#)

fteMigrateConfigurationOptions コマンドは、IBM WebSphere MQ File Transfer Edition 7.0 の構成オプションのセットをマイグレーションして、IBM WebSphere MQ 7.5 以降にコピーします (それらのファイルがターゲット・バージョンにまだ存在していない場合)。それらのファイルが既存の場合、メッセージが出力され、コマンドは続行されません。

Windows fteModifyAgent (Windows サービスとしての MFT エージェントの実行)

fteModifyAgent コマンドは、既存のエージェントを変更して、Windows サービスとして実行できるようにします。このコマンドは Windows でのみ使用可能であり、IBM MQ 管理者であり、mqm グループのメンバーであるユーザーが実行する必要があります。

構文



Parameters

-agentName agent_name

必須。変更するエージェントの名前。

-p configuration_options

オプション。このパラメーターは、エージェントを変更するために使用される構成オプションのセットを決定します。慣例として、このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、**fteModifyAgent** コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティ・ファイルのセットを使用します。

オプションの **-p** パラメーターは、デフォルトと異なる構成オプションを使用する場合にのみ指定します。このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。

-s service_name

オプション。エージェントを Windows サービスとして実行するように指示します。service_name を指定しないと、サービスの名前は mqmftAgentAGENTQMGR になります。AGENT はエージェント名、QMGR はエージェント・キュー・マネージャー名です。

Windows の「サービス」ウィンドウの「名前」列に示されるサービスの表示名は、常に「**Managed File Transfer Agent AGENT@QMGR**」です。

注：再配布可能エージェントが Windows サービスとして実行される場合、このサービスを機能させるには、システム環境で **BFG_DATA** 環境変数を設定する必要があります。

-su user_name

オプション。エージェントが Windows サービスとして実行される場合、このパラメーターは、サービスが実行されるアカウントの名前を指定します。Windows ドメイン・ユーザー・アカウントを使用してエージェントを実行するには、DomainName\UserName の形式で値を指定します。ローカル組み込みドメインからのアカウントを使用してサービスを実行するには、値を UserName の形式で指定します。

-su パラメーターを使用して指定する Windows ユーザー・アカウントには、**Log on as a service** 権限が必要です。この権限を付与方法については、[MFT エージェントまたはロガーを Windows サービスとして実行するためのガイダンス](#)を参照してください。

このパラメーターは、**-s** が指定された場合に必須です。

-sp password

オプション。このパラメーターは、**-s** が指定された場合にのみ有効です。

-sj options

オプション。エージェントを Windows サービスとして開始する場合は、このパラメーターによって、-D または -X という形で、Java 仮想マシン (JVM) に渡すオプションのリストを定義します。オプションは、番号記号 (#) またはセミコロン (;) 文字を使用して区切られます。# または ; 文字を組み込む必要がある場合は、それらを単一引用符で囲みます。

このパラメーターは、**-s** が指定された場合にのみ有効です。

fteModifyAgent コマンドが JVM オプションに対する更新の検証を処理する方法について詳しくは、[エージェントまたはロガーの JVM オプションを更新するためのガイダンス](#)を参照してください。

-sl options

オプション。Windows サービスのログ・レベルを設定します。有効なオプションは、error、info、warn、debug です。デフォルトは info です。このオプションは、Windows サービスに関して問題が発生した場合に便利です。これを debug に設定すると、より詳細な情報がサービス・ログ・ファイルに記録されます。

このパラメーターは、**-s** が指定された場合にのみ有効です。

-n
オプション。エージェントを通常のプロセスとして実行するように指示します。これは **-s** オプションと同時に指定できません。**-s** オプションも **-n** オプションも指定しない場合、エージェントは通常の Windows プロセスとして構成されます。

-? または -h
オプション。コマンド構文を表示します。

例

この例では、AGENT1 が Windows サービスとして実行されるように変更されています。

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

この例では、AGENT1 が Windows サービスを削除するように変更されています。

```
fteModifyAgent -agentName AGENT1
```

fteModifyAgent コマンドを実行する前に、[fteStopAgent](#) コマンドを使用して、変更するエージェントを停止する必要があります。

戻りコード

- 0**
コマンドは正常に完了しました。
- 1**
コマンドは失敗しました。

関連資料

388 ページの『[fteCreateAgent \(MFT エージェントの作成\)](#)』

fteCreateAgent コマンドは、Managed File Transfer Agent とその関連構成を作成します。

492 ページの『[fteModifyLogger \(Windows サービスとしての MFT ロガーの実行\)](#)』

Managed File Transfer ロガーを Windows サービスとして実行できるように変更するには、**fteModifyLogger** コマンドを使用します。このコマンドは Windows プラットフォームでのみ使用できます。IBM MQ 管理者であり、mqm グループのメンバーであるユーザーが実行する必要があります。まず、**fteStopLogger** コマンドを使用してロガーを停止する必要があります。

関連情報

[Windows サービスとしての MFT エージェントの開始](#)

[MFT エージェントまたはロガーを Windows サービスとして実行するためのガイダンス](#)

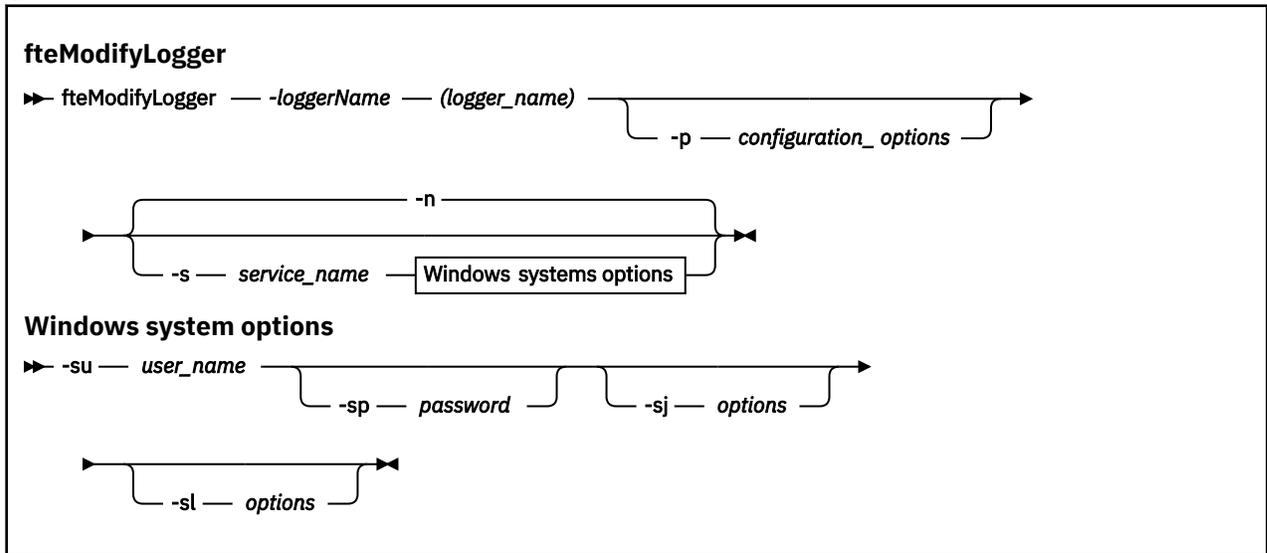
Windows fteModifyLogger (Windows サービスとしての MFT ロガーの実行)

Managed File Transfer ロガーを Windows サービスとして実行できるように変更するには、**fteModifyLogger** コマンドを使用します。このコマンドは Windows プラットフォームでのみ使用できます。IBM MQ 管理者であり、mqm グループのメンバーであるユーザーが実行する必要があります。まず、**fteStopLogger** コマンドを使用してロガーを停止する必要があります。

目的

スタンドアロン・ロガーは、ファイルの場合もデータベースの場合も、**サービス** アプリケーションの **名前** 列に "プロパティ・セット `logger_name@logger_qm` の Managed File Transfer ロガー" と表示されます。`logger_qm` の値は、ロガーのコマンド・キュー・マネージャーの名前です。

構文



Parameters

-loggerName (logger_name)

必須。変更する Managed File Transfer ロガーの名前。

-p configuration_options

オプション。このパラメーターは、ロガーを変更するために使用される構成オプションのセットを決定します。慣例として、このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、**fteModifyLogger** コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティ・ファイルのセットを使用します。

オプションの **-p** パラメーターは、デフォルトと異なる構成オプションを使用する場合にのみ指定します。このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。

-s service_name

オプション。ロガーを Windows サービスとして実行するように指示します。service_name を指定しない場合、サービスの名前は、mqmftLoggerLOGGERQMGR となります。ここで、LOGGER はロガーの名前であり、QMGR はロガー・キュー・マネージャーの名前です。

Windows の「サービス」ウィンドウの「名前」列に示されるサービスの表示名は、常に「**Managed File Transfer Logger LOGGER@QMGR**」です。

-su user_name

-s が指定されている場合は必須。Windows サービスを実行するアカウント名を指定します。

Windows ドメイン・ユーザー・アカウントを使用してエージェントを実行するには、DomainName\UserName の形式で値を指定します。ローカル組み込みドメインからのアカウントを使用してサービスを実行するには、値を UserName の形式で指定します。

-su パラメーターを使用して指定する Windows ユーザー・アカウントには、サービスとしてログオンする権限が必要です。この権限を付与する方法については、[MFT エージェントまたはロガーを Windows サービスとして実行するためのガイダンス](#)を参照してください。

-sp password

オプション。 **-s** が指定された場合のみ有効。 **-su** パラメーターによって設定されたユーザー・アカウントのパスワード。

-s パラメーターを指定する場合にこのパラメーターを指定しないと、サービスを正常に開始するには、あらかじめ Windows サービス・ツールを使用してパスワードを設定する必要があることを示す警告が出されます。

-sj options

オプション。 **-s** が指定された場合のみ有効。 ロガーが Windows サービスとして開始される場合は、このパラメーターは JVM に渡されるオプションのリストを **-D** または **-X** の形式で定義します。 オプションは、番号記号 (#) またはセミコロン (;) 文字を使用して区切られます。 # または ; 文字を組み込む必要がある場合は、それらを単一引用符 (') で囲みます。

fteModifyLogger コマンドが JVM オプションに対する更新の検証を処理する方法について詳しくは、[エージェントまたはロガーの JVM オプションを更新するためのガイダンス](#)を参照してください。

-sl options

オプション。 **-s** が指定された場合のみ有効。 Windows サービスのログ・レベルを設定します。 有効なオプションは、**error**、**info**、**warn**、**debug** です。 デフォルトは **info** です。 このオプションは、Windows サービスに関して問題が発生した場合に便利です。 これを **debug** に設定すると、より詳細な情報がサービス・ログ・ファイルに記録されます。

-n

オプション。 ロガーを通常のプロセスとして実行するように指示します。 これは **-s** オプションと同時に指定できません。 **-s** オプションも **-n** オプションも指定されなかった場合は、ロガーは通常の Windows プロセスとして構成されます。

-? または -h

オプション。 コマンド構文を表示します。

例

fteModifyLogger コマンドを実行する前に、[fteStopLogger](#) コマンドを使用してロガーを停止しておく必要があります。

この例では、**logger1** というロガーが以前に作成されています。 次のコマンドは、Windows サービスとして実行されるようにロガーを変更する方法を示しています。

```
fteModifyLogger -loggerName logger1 -s -su fteuser -sp ftepassword
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

関連資料

530 ページの『[fteStartLogger](#)』

fteStartLogger コマンドは、Managed File Transfer ロギング・アプリケーションを開始します。

533 ページの『[fteStopLogger](#)』

fteStopLogger コマンドは Managed File Transfer ロガーを停止します。

関連情報

[Windows サービスとしての MFT エージェントの開始](#)

[MFT エージェントまたはロガーを Windows サービスとして実行するためのガイダンス](#)

fteObfuscate: 機密データの暗号化

fteObfuscate コマンドは、資格情報ファイル内の機密データを暗号化します。 これにより、他のユーザーが資格情報ファイルにアクセスした際に、内容が読み取られるのを防ぐことができます。

目的

資格情報ファイル内のユーザー名とパスワードのプロパティは難読化できます。これらのプロパティは、暗号接尾部を付加すると、新しい関連プロパティに変換されます。以下に例を示します。

```
<!--
  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>
```

構文

fteObfuscate

▶ fteObfuscate — **-credentialsFile** — *credentials_file_name* ◀

パラメーター

-credentialsFile

必須。内容を難読化する資格情報ファイルの名前。

-? または -h

オプション。コマンド構文を表示します。

例

この例では、MQMFTCredentials.xml の内容が難読化されます。

```
fteObfuscate -credentialsFile /home/fteuser/MQMFTCredentials.xml
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

関連資料

42 ページの『MFT と IBM MQ の接続認証』

接続認証では、指定されたユーザー ID とパスワードを使用してアプリケーションを認証するようキュー・マネージャーを構成できます。関連付けられたキュー・マネージャーのセキュリティーが使用可能に設定されており、資格情報の詳細 (ユーザー ID とパスワード) が必要な場合、キュー・マネージャーと正常に接続するには、その前に接続認証フィーチャーを使用可能にしておく必要があります。接続認証は互換モードでも、MQCSP 認証モードでも実行できます。

810 ページの『MFT 資格情報ファイル・フォーマット』

MQMFTCredentials.xml ファイルには、ユーザー ID とパスワードに関する重要な情報が含まれています。MQMFTCredentials.xml ファイル内のエレメントは MQMFTCredentials.xsd スキーマに準拠する必要があります。資格情報ファイルのセキュリティーはユーザーの責任です。

ftePingAgent: MFT エージェントがアクティブかどうかの確認

ftePingAgent コマンドは、Managed File Transfer エージェントを ping して、そのエージェントはアクセス可能かどうか、可能な場合はシンプルな照会に応答できるかどうかを判別します。

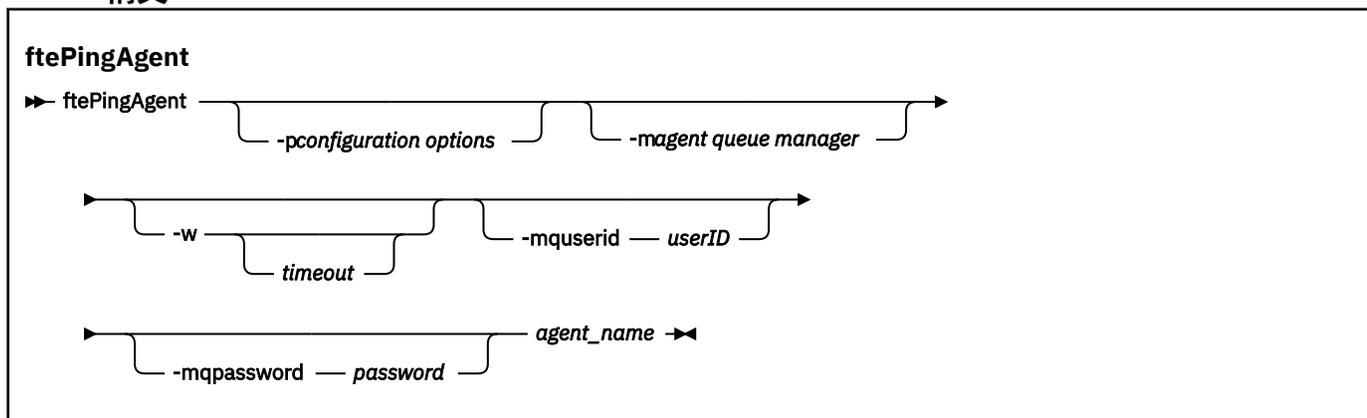
目的

ftePingAgent コマンドは、Managed File Transfer エージェントがアクセス可能かどうか、可能な場合は「are you there?」のやり取りに沿ってシンプルな照会に応答できるかどうかを確認するために使用します。このコマンドの出力例を以下に示します。

```
C:\> ftePingAgent AGENT86
5724-H72 Copyright IBM Corp. 2008, 2023. All Rights Reserved
BFGPR0127W: No credentials file has been specified to connect to IBM MQ. Therefore, the
assumption is that IBM MQ authentication has been disabled.
BFGCL0212I: Issuing ping request to agent AGENT86
BFGCL0213I: agent AGENT86 responded to ping in 0.094 seconds.
```

オプションの **-p** パラメーターは、デフォルト・セットと異なる構成オプションのセットを使用する場合にのみこのコマンドに指定します。詳しくは、[構成オプション](#)を参照してください。

構文



Parameters

-p (configuration options)

オプション。このパラメーターは、エージェントを ping する要求を発行するために使用される構成オプションのセットを決定します。このパラメーターの入力には、デフォルト調整キュー・マネージャ

ー以外の名前を使用します。これで、コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティ・ファイルのセットを使用します。このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。詳しくは、[構成オプション](#)を参照してください。

-m (queue manager)

オプション。ping 対象のエージェントが接続されているキュー・マネージャーの名前。-m パラメーターを指定しない場合、使用されるキュー・マネージャーは、使用中の構成オプションのセットから決定されます。

-w (timeout)

オプション。コマンドがエージェントの応答を *timeout* で示される秒数まで待機することを指定します。timeout を指定しない場合、または timeout 値に -1 を指定した場合、コマンドはエージェントの応答を無期限に待機します。このオプションを指定しない場合、デフォルトではエージェントの応答を 5 秒まで待機します。

timeout を指定すると、**ftePingAgent** コマンド・メッセージは、指定された送達不能キューに入れられずに、timeout 値の 2 倍の時間の経過後にタイムアウトになります。コマンドが無期限に待機するよう設定されている場合、コマンド・メッセージはタイムアウトになりません。

-mquserid (user ID)

オプション。コマンド・キュー・マネージャーで認証するユーザー ID を指定します。

-mqpassword (password)

オプション。コマンド・キュー・マネージャーで認証するパスワードを指定します。**-mquserid** パラメーターも一緒に指定する必要があります。**-mquserid** を指定した場合に **-mqpassword** を指定しないと、関連付けるパスワードを入力するというプロンプトが出されます。パスワードは表示されません。

(agent name)

必須。ping する Managed File Transfer エージェントの名前。

-? または -h

オプション。コマンド構文を表示します。

例

この例では、コマンドは、QM_MERCURY に接続されたエージェント AGENT1 を ping します。コマンドは、AGENT1 の応答を 40 秒まで待機してから戻ります。

```
ftePingAgent -m QM_MERCURY -w 40 AGENT1
```

戻りコード

0

コマンドは正常に完了しました。エージェントはアクティブで、転送を処理できます。

1

コマンドは失敗しました。コマンドはメッセージをエージェントに送信できませんでした。

2

コマンドはタイムアウトになりました。コマンドはメッセージをエージェントに送信しましたが、エージェントは時間内に応答しませんでした。

関連資料

473 ページの『[fteListAgents: 調整キュー・マネージャーの MFT エージェントのリスト](#)』

fteListAgents コマンドは、特定の調整キュー・マネージャーに登録されているすべての Managed File Transfer のエージェントをリストするために使用します。

518 ページの『[fteShowAgentDetails](#)』

fteShowAgentDetails コマンドは、特定の Managed File Transfer Agent の詳細を表示するために使用します。これらは、エージェントの Managed File Transfer 調整キュー・マネージャーによって保管される詳細です。

314 ページの『ファイル転送が停止したと思われる場合の対処法』

負荷の重いシステムであったり、ソース・エージェントと宛先エージェントとの間でネットワーク問題があったりする場合には、転送が待機状態またはリカバリー状態で停止しているように見えることがあります。この原因となるいくつかの要因があります。

fteRAS: MFT のトラブルシューティング情報の収集

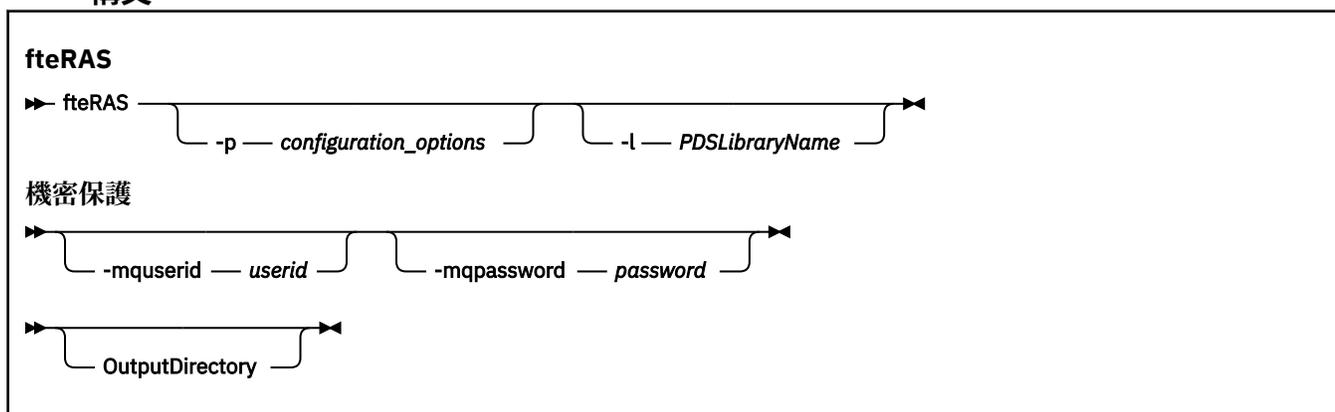
fteRAS コマンドは、Managed File Transfer のトラブルシューティング情報 (MustGather データ) を収集します。**fteRAS** が収集する情報は、プログラムが実行されているシステム上の Managed File Transfer インストール済み環境に固有のものです。

目的

Managed File Transfer エージェント、データベース・ロガー、またはその他のコマンドが問題を報告しているか、正しく機能しない場合に解決策を見つけるために使用するトラブルシューティング情報を収集する必要がある場合は、**fteRAS** コマンドを使用して、信頼性、可用性、および保守性情報 (RAS) 収集ツールを実行します。

fteRAS コマンドを実行する時には、結果を保管するアーカイブ (.zip) ファイルを書き込む出力ディレクトリとして、デフォルトの場所を使用することも、自分で選択したディレクトリを指定することもできます。

構文



Parameters

-p configuration_options

オプション。トラブルシューティング情報 (エージェントのリストなど) の収集に使用する構成オプションのセットを決定します。構成オプションのセット名を **-p** パラメーターの値として使用します。慣例により、この名前は調整キュー・マネージャーの名前です。このパラメーターを指定しない場合、デフォルトの構成オプションのセットが使用されます。

z/OS -l

オプション。z/OS のみ。特定のエージェントまたはロガーを対象にして MQMFT コマンドを呼び出す JCL スクリプトが入っている PDS ライブラリーの名前を指定します。コマンド PDS ライブラリーの BFGRAS JCL スクリプトからこのコマンドを実行して、PDS ライブラリーのすべてのメンバーを出力ディレクトリに取り込む場合は、常にこのオプションを設定します。

-mquserid user id

オプション。コマンド・キュー・マネージャーで認証するユーザー ID を指定します。

-mqpassword password

オプション。コマンド・キュー・マネージャーで認証するパスワードを指定します。 **-mquserid** パラメーターも一緒に指定する必要があります。 **-mquserid** を指定する際に一緒に **-mqpassword** を指定しないと、関連パスワードを指定するようプロンプトが表示されます。パスワードは画面には表示されません。

OutputDirectory

オプション。RAS データを収集するときに使用するディレクトリー。データ収集が成功すると、ここに出力ファイル (例えば **fteRAS.zip**) が保管されます。ディレクトリーが存在しない場合は、作成されます。デフォルトの場所は **mqft** の **logs** ディレクトリーです。

-? または -h

オプション。コマンド構文を表示します。

例

Linux **UNIX** UNIX および Linux で出力ファイル **fteRAS.zip** を **/var/mqm/errors** ディレクトリーに保管する場合は、以下の例のようにして **fteRAS** を実行します。

```
fteRAS /var/mqm/errors
```

コマンドが正常に完了したことを確認する以下のようなメッセージが表示されます。

BFGCL0604I: fteRAS コマンドが正常に完了しました。 出力は /var/mqm/errors/fteRAS.zip に保管されます。

Windows Windows で出力ファイル **fteRAS.zip** を IBM MQ の新しいインストール環境のデフォルトの **errors** ディレクトリーに保管する場合は、以下の例のようにして **fteRAS** を実行します。

```
fteRAS "C:\ProgramData\IBM\MQ\errors"
```

コマンドが正常に完了したことを確認する以下のようなメッセージが表示されます。

BFGCL0604I: fteRAS コマンドが正常に完了しました。 出力は C:\ProgramData\IBM\MQ\errors\fteRAS.zip に保管されます。

注: IBM MQ 8.0 以降の場合は、対象バージョンの製品の新しいインストール環境でなければ、お使いのシステムでは **errors** ディレクトリーの場所は異なる可能性があります。詳細については、[プログラム・ディレクトリーとデータ・ディレクトリーの場所 \(Windows\)](#) を参照してください。

IBM i IBM i で出力ファイルを **/QIBM/UserData/mqm/errors** にコピーする場合は、以下の例のようにして **Qshell** から **fteRAS** コマンドを実行します。

```
/QIBM/ProdData/mqm/bin/fteRAS /QIBM/UserData/mqm/errors
```

コマンドが正常に完了したことを確認する以下のようなメッセージが表示されます。

BFGCL0604I: fteRAS コマンドが正常に完了しました。 出力は /QIBM/UserData/mqm/errors/fteRAS.zip に保管されません。

関連資料

270 ページの『[トラブルシューティング MFT](#)』

以下の参照情報を、Managed File Transfer のエラーの診断に役立ててください。

V 9.0.0 fteSetAgentLogLevel: MFT プロトコル・ブリッジ・エージェントのログ・レベルの設定

fteSetAgentLogLevel コマンドを使用して、Managed File Transfer プロトコル・ブリッジ・エージェントと FTP/SFTP/FTPS ファイル・サーバーの間のファイル転送に関する診断情報のロギングを有効または無効にします。

目的

このコマンドを使用して、Managed File Transfer プロトコル・ブリッジ・エージェントと FTP/SFTP/FTPS ファイル・サーバーの間のファイル転送に関する診断情報の収集のログギングを有効または無効にします。Managed File Transfer プロトコル・ブリッジ・エージェントのログ記録は、より軽量で、トレースとは分離されていますが、トレースと並行して実行できます。

ファイル転送中に発生する可能性のある問題を簡単に診断できるように、プロトコル・ブリッジ・エージェントから FTP/SFTP/FTPS ファイル・サーバーに送信されるコマンドと受信する応答がログ・ファイルに書き込まれます。

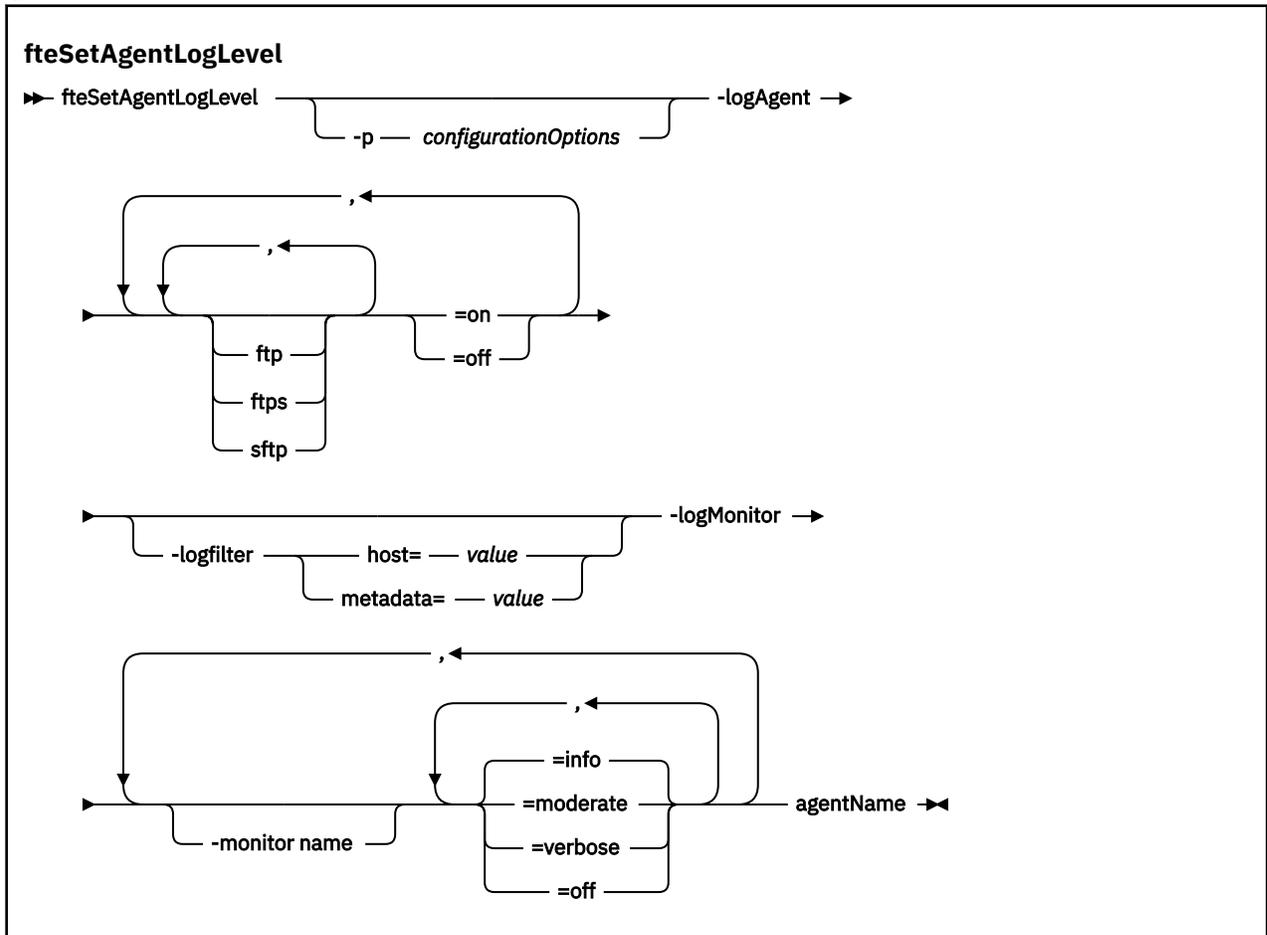
以下の 2 つの方法で、必要なログギングを有効にしたり、無効にしたり、レベルを設定したりすることができます。

- **fteSetAgentLogLevel** コマンドを使用して、プロトコル・ブリッジ・エージェントの実行中にログギングを有効または無効にします。ログギングを開始するためにプロトコル・ブリッジ・エージェントを再始動する必要はありません。
- **agent.properties** ファイルを使用して、エージェントの開始時にログギングを有効または無効にします。この方式を使用してログギングを有効または無効にする場合、エージェントが実行中のときには、ログギングを開始するためにエージェントを再始動する必要があります。詳しくは、[agent.properties ファイル](#)を参照してください。

プロトコル・ブリッジ・エージェントとファイル・サーバーの間の対話はイベントとして扱われ、各イベントは `agentevent?.log` ファイルに書き込まれます。「?」ログ名が数値を表しています。名前に 0 (ゼロ) が含まれるファイルには、キャプチャーされたメッセージの最新のセットが入っています。ログ・ファイルは、`MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/` フォルダに格納されます。

その他のログギング・プロパティを、例えばログ・ファイルのサイズや保持するログ・ファイルの数を制限するために、[agent.properties ファイル](#)に設定することができます。

構文



Parameters

-logAgent component=operation

必須。

V 9.0.3

重要: logAgent と **logMonitor** のうちいずれか1つしか選択できません。両方のオプションを選択しようとすると、エラー・メッセージが表示されます。

ロギングを有効または無効にするプロトコル・ブリッジ・エージェントの名前。プロトコル・ブリッジ・エージェントのロギングは、以下の3つのコンポーネントで有効または無効にすることができます。ftpサーバー、ftpsサーバー、およびsftpサーバー。可能な3つのサーバー・タイプのいずれかを指定して、操作値を追加することにより、接続先のプロトコル・ブリッジ・エージェントのロギングをoffまたはonにします。

V 9.0.3

コンポーネントが正符号(+)で始まる場合は、正符号に続くコンポーネントのリストが、現在ログに記録されている既存のログ・コンポーネントに追加されます。

有効なログ・レベルの操作オプションは以下のとおりです。

off

プロトコル・ブリッジ・エージェントのすべてのロギングを無効にします。これはデフォルトのオプションです。

```
fteSetAgentLogLevel -logAgent =off PBA1
```

プロトコル・ブリッジ・エージェントが接続しているコンポーネントを指定してロギングを無効にするには、以下のコマンドを使用します。

```
fteSetAgentLogLevel -logAgent ftp=off PBA1
```

```
fteSetAgentLogLevel -logAgent ftps=off PBA1
```

```
fteSetAgentLogLevel -logAgent sftp=off PBA1
```

ON

プロトコル・ブリッジ・エージェントが接続している可能性がある3つのファイル・サーバー・コンポーネントすべてでロギングを有効にするには、次のコマンドを使用します。

```
fteSetAgentLogLevel -logAgent =on PBA1
```

プロトコル・ブリッジ・エージェントが接続しているコンポーネントを指定してロギングを有効にするには、以下のコマンドを使用します。

```
fteSetAgentLogLevel -logAgent ftp=on PBA1
```

```
fteSetAgentLogLevel -logAgent ftps=on PBA1
```

```
fteSetAgentLogLevel -logAgent sftp=on PBA1
```

その他の構成オプションについては、[504 ページの『例 1』](#)および[504 ページの『例 2』](#)を参照してください。

-agentName

必須。ロギングを有効または無効にするプロトコル・ブリッジ・エージェントの名前。

-logFilter filter=value

オプション。ロギングを指定のフィルター基準に制限するために使用します。**host** またはユーザー指定のファイル転送 **metadata** のいずれかの値を指定する必要があります。

host

次の3つの値のいずれかをフィルター基準として指定します。ホスト名、コンマ区切りのホスト名のリストまたはIPアドレス。FTPサーバー `ftpprod.ibm.com` に対するコマンドとそこからの応答をログに記録するには、次のコマンドを使用します。

```
fteSetAgentLogLevel -logAgent ftp=on -logFilter host=ftpprod.ibm.com PBA1
```

IPアドレスが `9.182.*` で始まるすべてのSFTPサーバーに対するコマンドとそこからの応答をログに記録するには、次のコマンドを使用します。

```
fteSetAgentLogLevel -logAgent sftp=on -logFilter host=9.182.* PBA1
```

metadata

転送の作成中にユーザーによって定義されたテキストを `key=value` のフォーマットで指定します。例えば、**metadata="BANK=WORLD BANK"** のように指定します。プロトコル・ブリッジ・エージェント PBA1 に接続するFTPサーバーに対するロギングを有効にし、ファイル転送をフィルターに

掛けて **metadata** がキーとして含まれ "BANK=WORLD BANK" が値として含まれる場合だけを含むようにするためには、次のコマンドを使用します。

```
fteSetAgentLogLevel -logAgent ftp=on metadata="BANK=WORLD BANK" PBA1
```

注: メタデータに基づいてフィルターに掛けるためには、ファイル転送の一部としてフィルター基準となる値を **-md** パラメーターの下に指定する必要があります。詳しくは、[437 ページの『fteCreateTransfer: 新規ファイル転送の開始』](#)を参照してください。

V 9.0.3 **-logMonitor monitor name=log level**

必須。

重要: logAgent と **logMonitor** のうちいずれか1つしか選択できません。両方のオプションを選択しようとする、エラー・メッセージが表示されます。

ロギングを有効または無効にするリソース・モニターのコマ区切りのリスト。各項目の意味は次のとおりです。

monitor name

オプション。ロギングを有効または無効にするリソース・モニター名、またはリソース・モニター名のリスト。名前を指定しない場合、ロギングはエージェントのすべてのリソース・モニターで有効になります。

log level

必須。以下のいずれかです。

info

情報レベルのロギングをオンにします。これはデフォルト値です。

エージェント AGENT1 のモニター MON1 で info レベルのロギングを有効にするには、次のように指定します。

```
fteSetAgentLogLevel -logMonitor MON1=info AGENT1
```

moderate

普通レベルのロギングをオンにします。

エージェント AGENT1 のモニター MON1 および MON2 で moderate レベルのロギングを有効にするには、次のように指定します。

```
fteSetAgentLogLevel -logMonitor MON1, MON2=moderate AGENT1
```

詳細

詳細レベルのロギングをオンにします。

エージェント AGENT1 のすべてのモニターで verbose レベルのロギングを有効にするには、次のように指定します。

```
fteSetAgentLogLevel -logMonitor =verbose AGENT1
```

off

ロギングをオフにします。

エージェント AGENT1 のモニター MON1 および MON2 でロギングをオフにするには、次のように指定します。

```
fteSetAgentLogLevel -logMonitor MON1, MON2=off AGENT1
```

エージェント AGENT1 で、モニター MON1 のロギングをオフにして、モニター MON2 の info レベルのロギングを有効にするには、次のように指定します。

```
fteSetAgentLogLevel -logMonitor MON1=off, MON2=info AGENT1
```

エージェント AGENT1 のすべてのモニターでログングをオフにするには、次のように指定します。

```
fteSetAgentLogLevel -logMonitor =off AGENT1
```

V 9.0.3 **-p configurationOptions**

オプション。エージェント・ログ・レベルを設定するために使用される構成オプションのセットを決定します。構成オプションのセット名を **-p** パラメーターの値として使用します。

慣例により、これは調整キュー・マネージャーの名前です。このパラメーターを指定しない場合、デフォルトの構成オプションのセットが使用されます。

-? または -h

オプション。コマンド構文を表示します。

V 9.0.3 コマンドで同じリソース・モニター名が繰り返し指定されると、最後に出現するコンポーネントが有効と見なされます。以下に例を示します。

```
fteSetAgentLogLevel -logMonitor MON1=info, MON2=off, MON3=off AGENT1
```

この例では、リソース・モニター MON1 のログングがオフになります。

V 9.0.3



重要: 存在しないリソース・モニター名をコマンドで指定した場合、コンソールにエラーは表示されません。

例 1

この例では、コマンド区切りグループを使用して、複数のコンポーネントを 1 つのコマンドで指定しています。PBA1 プロトコル・ブリッジ・エージェントが接続している FTP および SFTP サーバーでログングが有効になりますが、FTPS サーバーでは無効になります。

```
fteSetAgentLogLevel -logAgent ftp=on,ftps=off,sftp=on PBA1
```

複数のコンポーネントをコンマで区切っても、同じ結果になります。次に例を示します。

```
fteSetAgentLogLevel -logAgent ftp,sftp=on,ftps=off PBA1
```

例 2

この例では、1 つのコマンド内で同じコンポーネントが繰り返されます。 *component=operation* ペアの最後のインスタンスが有効と見なされます。この例では、PBA1 プロトコル・ブリッジ・エージェントが接続している FTP サーバーでのログングが無効になります。

```
fteSetAgentLogLevel -logAgent ftp=on,ftp=off PBA1
```

直前の例は、次の例と同じになります。

```
fteSetAgentLogLevel -logAgent ftp=off PBA1
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

関連概念

214 ページの『[プロトコル・ブリッジ](#)』

プロトコル・ブリッジを使用すれば、Managed File Transfer (MFT) ネットワークから、MFT ネットワークの外部（ローカル・ドメインとリモート・ロケーションの両方）にあるファイル・サーバーに格納されているファイルにアクセスできます。このファイル・サーバーでは、FTP、FTPS、または SFTP ネットワーク・プロトコルを使用できます。それぞれのファイル・サーバーで少なくとも 1 つの専用エージェントが必要です。この専用エージェントは、プロトコル・ブリッジ・エージェントとして知られています。ブリッジ・エージェントは、複数のファイル・サーバーと相互作用できます。

関連資料

550 ページの『[MFT agent.properties ファイル](#)』

各 Managed File Transfer Agent には、独自のプロパティ・ファイル `agent.properties` があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。`agent.properties` ファイルには、エージェントの動作を変更するプロパティを含めることもできます。

437 ページの『[fteCreateTransfer: 新規ファイル転送の開始](#)』

fteCreateTransfer コマンドは、コマンド行から新規のファイル転送を作成して開始します。このコマンドは、ファイル転送の即時開始、将来の日時におけるファイル転送のスケジュール、ファイル転送の 1 回以上の繰り返し、および特定の条件に基づくファイル転送の起動を実行できます。

fteSetAgentTraceLevel

fteSetAgentTraceLevel コマンドを使用して、エージェントの現在のトレース・レベルを動的に変更することができます。

目的

エージェント・トレースのオンとオフを切り替えるとき、または設定されているエージェント・トレースのレベルを変更するときに、このコマンドを使用します。**fteSetAgentTraceLevel** コマンドを使用すると、トレース・レベルを変更するために、エージェントをシャットダウンして再始動する必要はありません。生成されるトレース・ファイルは、`MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/trace%PID%/trace%PID%.txt`（ここで、`%PID%` はエージェント・インスタンスのプロセス ID）に置かれます。



重要:

Multi マルチプラットフォームで IBM WebSphere MQ 7.5 以降を使用する場合、エージェント・プロセスを実行しているユーザーのみが **fteSetAgentTraceLevel** コマンドを実行できます。

z/OS V 9.0.2 z/OS の場合、Continuous Delivery では、IBM MQ 9.0.2 以降、以下のいずれかの方法で **fteSetAgentTraceLevel** コマンドを実行できます。

- エージェント・プロセスを実行しているものと同じユーザー ID。
- エージェント・プロパティ **adminGroup** で指定されたグループのメンバー。

V 9.0.0.1 これは、IBM MQ 9.0.0 Fix Pack 1 以降の Long Term Support リリースにも適用されません。

詳しくは、**adminGroup** プロパティ（550 ページの『[MFT agent.properties ファイル](#)』）を参照してください。

IBM WebSphere MQ 7.5 以降では、**fteSetAgentTraceLevel** コマンドは、エージェント・プロセス・コントローラーのトレースの書き込みも行います。生成されるトレース・ファイルは、`MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/pctrace%PID%/pctrace%PID%.txt`（ここで、`%PID%` はエージェント・インスタンスのプロセス ID）に置かれます。

このコマンドを使用して、エージェント・プロセスで Javacore を生成することもできます。エージェントによって、`MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name` ディレクトリ内に Javacore ファイルが生成されます。

トレースを実行すると、パフォーマンスに重大な影響を及ぼすおそれがあり、また大量のトレース・データが生成される可能性もあるため、トレースの実行は注意して、必要な場合にのみ実施してください。通常は、IBM サービス担当員からトレースを依頼された場合に限り、トレースを有効にします。



重要:

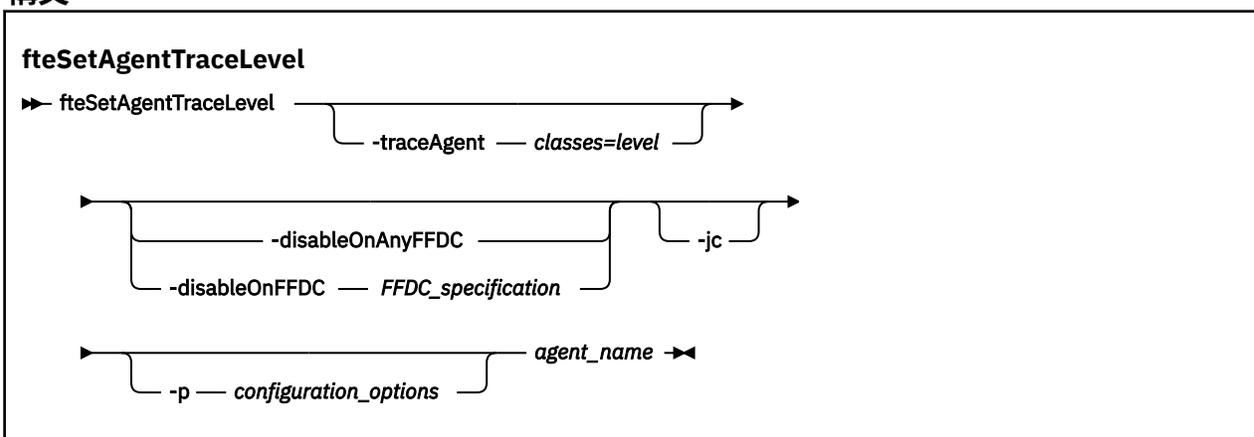
1. このコマンドは、エージェントが実行されているシステムで実行する必要があります。
2. トレースとログは、エージェントの再始動後は持続しません。

エージェントが終了し、プロセス・コントローラー・プロセスによって再始動された場合、必要なトレース・プロパティとログ・プロパティを含むように `agent.properties` ファイルが更新されるまで、動的トレースとログは有効になりません。

トレース・ファイルのサイズや保持するトレース・ファイルの数などの他のトレース・プロパティを `agent.properties` ファイルに設定することができます。これらのプロパティについては、『[拡張エージェント・プロパティ](#)』で説明されています。

オプションの `-p` パラメーターは、デフォルト・セットと異なる構成オプションのセットを使用する場合にのみこのコマンドに指定します。詳しくは、『[550 ページの『MFT agent.properties ファイル』](#)』を参照してください。

構文



Parameters

-traceAgent classes=level

必須。エージェント・トレースを設定するレベルおよびトレースの適用対象クラス。

V 9.0.5 IBM MQ 9.0.5 以降では、コロンで区切られたクラス仕様のリストを指定できます。このオプションを使用すると、さまざまなクラスを異なるレベルでトレースするように設定できます。以下に例を示します。

```
fteSetAgentTraceLevel -traceAgent com.ibm.wmqfte.agent=all:com.ibm.wmqfte.cmdhandler=moderate AGENT1
```

このレベルのトレースを適用する対象となる、コンマで区切られたクラス仕様のリストは引き続き指定できます。このパラメーターが指定されない場合、トレース・レベルはすべてのエージェント・クラスに適用されます。次の形式を使用します。

```
classes=level
```

以下に例を示します。

```
com.ibm.wmqfte=all
```

classes を Managed File Transfer パッケージ名に置換すると、特定のパッケージのみをトレースできます。ただし、このオプションでキャプチャーされるのはエージェントの動作のサブセットのみであるため、通常はパッケージのフィルタリングを使用しないことをお勧めします。

(*classes*) が正符号 (+) で始まる場合、現在トレースされている既存のトレース・クラスすべてに、正符号に続くトレース・クラスのリストが追加されます。

有効なトレース・レベル・オプションは以下のとおりです。トレース・ファイルのサイズと詳細度の昇順にリストしています。

off

エージェント・トレースをオフに切り替えますが、ログ・ファイルへの情報の書き込みは続行されます。これはデフォルト・オプションです。

flow

エージェント内でプロセッシング・フローに関連付けられたトレース・ポイントのデータを収集します。

moderate

トレースであまり多くない量の診断情報を収集します。

詳細

トレースで詳細な診断情報を収集します。

all

すべてのエージェント・クラスでエージェント・トレースを実行するように設定します。

エージェントの完全トレースを開始するには、次のコマンドを実行します。

```
fteSetAgentTraceLevel -traceAgent =all AGENT_NAME
```

エージェントの完全トレースを停止するには、次のコマンドを実行します。

```
fteSetAgentTraceLevel -traceAgent =off AGENT_NAME
```

-disableOnAnyFFDC

オプション。このパラメーターを指定すると、First Failure Data Capture (FFDC) ファイルを生成するときに、エージェントでトレースが無効になります。

-disableOnAnyFFDC および **-disableOnFFDC** パラメーターのいずれか一方のみ指定できます。

-disableOnFFDC FFDC_specification

オプション。このパラメーターを指定すると、*FFDC_specification* に合致する First Failure Data Capture (FFDC) ファイルを生成するときに、エージェントでトレースが無効になります。*FFDC_specification* は、値のコンマ区切りリストです。値のフォーマットは以下のいずれかです。

class_name

FFDCが発生したクラスの名前。例えば `com.ibm.wmqfte.classA` です。

class_name:probe_ID

クラスの名前と、FFDCの発生元のクラスにあるロケーションのプロブ ID。例えば `com.ibm.wmqfte.classB:1` です。

-disableOnAnyFFDC および **-disableOnFFDC** パラメーターのいずれか一方のみ指定できます。

-jc

オプション。エージェントが Javacore ファイルを生成するように要求します。IBM サービス・チームが問題診断のために、このパラメーターを指定してコマンドを実行するように求める場合があります。このパラメーターは、**-p** 以外のパラメーターと一緒に使用することはできません。

-p configuration_options

オプション。このパラメーターは、エージェント・トレース・レベルを設定するために使用される構成オプションのセットを決定します。このパラメーターの入力には、デフォルト調整キュー・マネー

ジャー以外の名前を使用します。これで、コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティ・ファイルのセットを使用します。

このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。

agent_name

必須。トレース・レベルを設定する Managed File Transfer Agent の名前。

-? または -h

オプション。コマンド構文を表示します。

例

この例では、AGENT1 のすべてのクラスについて、トレース・レベルが all に設定されています。

```
fteSetAgentTraceLevel -traceAgent com.ibm.wmqfte=all AGENT1
```

この例では、AGENT1 の `com.ibm.wmqfte.agent.Agent` と `com.ibm.wmqfte.cmdhandler` クラスについて、トレース・レベルが all に設定されています。

```
fteSetAgentTraceLevel -traceAgent com.ibm.wmqfte.agent.Agent,com.ibm.wmqfte.cmdhandler=moderate AGENT1
```

以下の例では、**-traceLevel** パラメーターが off に設定されているため、サブクラスがトレースから除外されます。com.ibm.outer.inner で始まるクラスを除いて、com.ibm.outer で始まるクラスはすべて、verbose レベルでトレースされます。

```
fteSetAgentTraceLevel -traceAgent com.ibm.outer=verbose AGENT1
fteSetAgentTraceLevel -traceAgent +com.ibm.outer.inner=off AGENT1
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

fteSetLoggerTraceLevel

fteSetLoggerTraceLevel コマンドを使用して、Managed File Transfer ロガーの現在のトレース・レベルを動的に変更することができます。

目的

ロガーのオンとオフを切り替えるか、設定されているロガー・トレースのレベルを変更するために、このコマンドを使用します。**fteSetLoggerTraceLevel** コマンドを使用すると、トレース・レベルを変更するために、ロガーをシャットダウンして再始動する必要はありません。生成されるトレース・ファイルは、`MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/trace%PID%/trace%PID%.txt` (ここで、%PID% はロガー・インスタンスのプロセス ID) に置かれます。

IBM WebSphere MQ 7.5 以降では、**fteSetLoggerTraceLevel** コマンドは、ロガー・プロセス・コントローラーのトレースの書き込みも行います。生成されるトレース・ファイルは、`MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs/pctrace%PID%/pctrace%PID%.txt` (ここで、%PID% はロガー・インスタンスのプロセス ID) に置かれます。

このコマンドを使用すると、ロガー・プロセスで Javacore を生成することもできます。ロガーによって、`MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name` ディレクトリー内に Javacore ファイルが生成されます。

トレースを実行すると、パフォーマンスに重大な影響を及ぼすおそれがあり、また大量のトレース・データが生成される可能性もあるため、トレースの実行は注意して、必要な場合にのみ実施してください。通常は、IBM サービス担当員からトレースを依頼された場合に限り、トレースを有効にします。



重要:

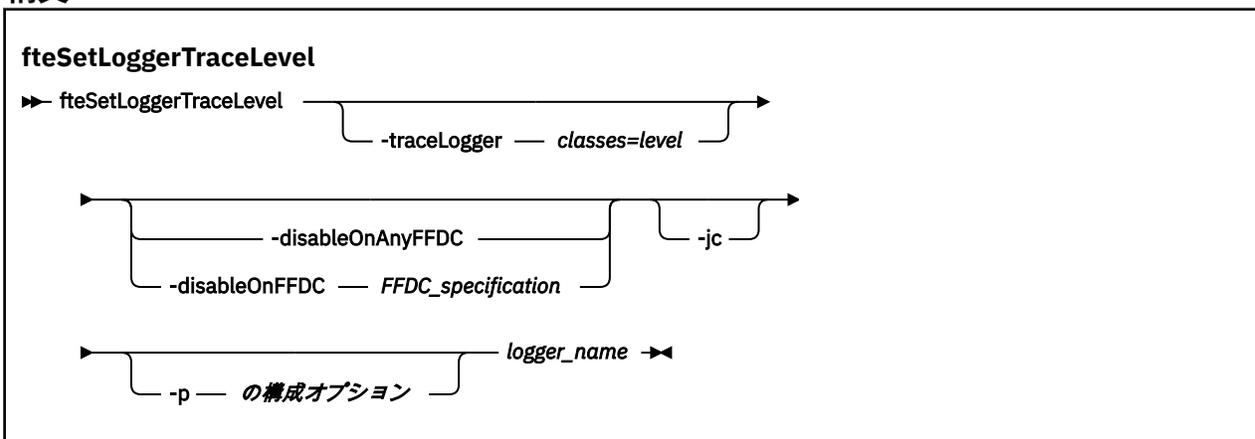
1. このコマンドは、エージェントが実行されているシステムで実行する必要があります。
2. トレースとログギングは、エージェントの再始動後は持続しません。

エージェントが終了し、プロセス・コントローラー・プロセスによって再始動された場合、必要なトレース・プロパティとログ・プロパティを含むように `agent.properties` ファイルが更新されるまで、動的トレースとログは有効になりません。

`logger.properties` ファイルでは、トレース・ファイルのサイズや保持するトレース・ファイルの数など、その他のトレース・プロパティを設定することができます。これらのプロパティについては、[ロガー・プロパティ](#)で説明しています。

オプションの `-p` パラメーターは、デフォルト・セットと異なる構成オプションのセットを使用する場合にのみこのコマンドに指定します。詳しくは、[118 ページの『MFT ロガーの構成プロパティ』](#)を参照してください。

構文



Parameters

`-traceLogger classes=level`

必須。ロガー・トレースを設定するレベルおよびトレースの適用対象クラス。

V 9.0.5 IBM MQ 9.0.5 以降では、コロンで区切られたクラス仕様のリストを指定できます。このオプションを使用すると、さまざまなクラスを異なるレベルでトレースするように設定できます。以下に例を示します。

```
fteSetLoggerTraceLevel -traceLogger com.ibm.wmqfte.logger=all:com.ibm.wmqfte.cmdhandler=moderate  
LOGGER1
```

このレベルのトレースを適用する対象となる、コンマで区切られたクラス仕様のリストは引き続き指定できます。このパラメーターが指定されない場合、トレース・レベルはすべてのエージェント・クラスに適用されます。次の形式を使用します。

```
classes=level
```

以下に例を示します。

```
com.ibm.wmqfte=all
```

トレースのレベルの適用対象の、コンマで区切られたクラス仕様のリストを指定します。このパラメーターが指定されない場合、トレース・レベルはすべてのロガー・クラスに適用されます。

(*classes*) が正符号 (+) で始まる場合、現在トレースされている既存のトレース・クラスすべてに、正符号に続くトレース・クラスのリストが追加されます。

有効なトレース・レベル・オプションは以下のとおりです。トレース・ファイルのサイズと詳細度の昇順にリストしています。

off

ロガー・トレースをオフに切り替えますが、ログ・ファイルへの情報の書き込みは続行されます。これはデフォルト・オプションです。

flow

ロガー内でプロセッシング・フローに関連付けられたトレース・ポイントのデータを収集します。

moderate

トレースであまり多くない量の診断情報を収集します。

詳細

トレースで詳細な診断情報を収集します。

all

すべてのロガー・クラスでロガー・トレースを実行するように設定します。

-disableOnAnyFFDC

オプション。このパラメーターを指定すると、First Failure Data Capture (FFDC) ファイルを生成するときに、ロガーでトレースが無効になります。

-disableOnAnyFFDC および **-disableOnFFDC** パラメーターのいずれか一方のみ指定できます。

-disableOnFFDC *FFDC_specification*

オプション。このパラメーターを指定すると、*FFDC_specification* に合致する First Failure Data Capture (FFDC) ファイルを生成するときに、ロガーでトレースが無効になります。*FFDC_specification* は、値のコンマ区切りリストです。以下のいずれかの形式の値を使用できます。

class_name

FFDCが発生したクラスの名前。例えば `com.ibm.wmqfte.classA` です。

class_name:probe_ID

クラスの名前と、FFDCの発生元のクラスにあるロケーションのプローブ ID。例えば `com.ibm.wmqfte.classB:1` です。

-disableOnAnyFFDC および **-disableOnFFDC** パラメーターのいずれか一方のみ指定できます。

-jc

オプション。ロガーが Javacore ファイルを生成するように要求します。IBM サービス・チームが問題診断のために、このパラメーターを指定してコマンドを実行するように求める場合があります。**-jc** パラメーターは他のパラメーターと一緒に使用することはできません。

-p *configuration_options*

オプション。このパラメーターは、ロガー・トレース・レベルを設定するために使用される構成オプションのセットを決定します。このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティ・ファイルのセットを使用します。

このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。

logger_name

必須。トレース・レベルを設定する Managed File Transfer Logger の名前。

-? または -h

オプション。コマンド構文を表示します。

例

この例では、LOGGER1 のすべてのクラスについて、トレース・レベルが **all** に設定されています。

```
fteSetLoggerTraceLevel -traceLogger com.ibm.wmqfte=all LOGGER1
```

この例では、LOGGER1 の `com.ibm.wmqfte.logger.logger` と `com.ibm.wmqfte.cmdhandler` クラスについて、トレース・レベルが **all** に設定されています。

```
fteSetLoggerTraceLevel -traceLogger com.ibm.wmqfte.logger.logger,com.ibm.wmqfte.cmdhandler=moderate  
LOGGER1
```

以下の例では、**-traceLevel** パラメーターが **off** に設定されているため、サブクラスがトレースから除外されます。 `com.ibm.outer.inner` で始まるクラスを除いて、 `com.ibm.outer` で始まるクラスはすべて、 **verbose** レベルでトレースされます。

```
fteSetLoggerTraceLevel -traceLogger com.ibm.outer=verbose LOGGER1  
fteSetLoggerTraceLevel -traceLogger +com.ibm.outer.inner=off LOGGER1
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

z/OS V 9.0.3 fteSetProductId: z/OS SCRT レコーディング ID の設定

fteSetProductId は、インストール済み環境における Managed File Transfer 使用率の記録の対象となる製品タイプを設定する際に使用します。このコマンドは、z/OS でのみ有効です。

目的

このコマンドは、1つ以上の調整キュー・マネージャーが定義されるか、インストール済み環境用の `installation.properties` ファイルが作成された後にいつでも実行できます。

製品使用記録の詳細については、[製品情報のレポート](#)を参照してください。

構文



Parameters

使用率レコーディングの対象となる製品タイプ:

以下のいずれかを指定します。

MFT

スタンドアロン Managed File Transfer 製品として使用率が記録されます (製品 ID は 5655-MF9)。

ADVANCED

IBM MQ Advanced for z/OS 製品の一部として使用率が記録されます (製品 ID は 5655-AV9)。

ADVANCEDVUE

IBM MQ Advanced for z/OS, Value Unit Edition 製品の一部として使用率が記録されます (製品 ID は 5655-AV1)。

戻りコード

0

コマンドは正常に完了しました。

1

コマンドが正常に終了しなかったか、製品タイプが要求された値に設定されていません。

関連タスク

98 ページの『MFT 用の調整キュー・マネージャーの構成』

fteSetupCoordination コマンドを実行したら、`MQ_DATA_PATH/mqft/config/coordination_qmgr_name` ディレクトリー内の `coordination_qmgr_name.mqsc` スクリプトを実行して、調整キュー・マネージャーに関する必要な構成を行います。ただし、この構成を手動で行う場合は、調整キュー・マネージャーに対して以下の手順を実行します。

fteSetupCommands: MFT command.properties ファイルの作成

fteSetupCommands コマンドは、Managed File Transfer `command.properties` ファイルを作成します。このプロパティー・ファイルは、コマンドの発行時に IBM MQ ネットワークに接続するキュー・マネージャーの詳細を指定します。

重要: **ULW** IBM MQ for UNIX, Linux, and Windows では、IBM MQ 管理者 (および `mqm` グループのメンバー) であるユーザーのみが、このコマンドを実行できます。IBM MQ 管理者以外のユーザーとしてこのコマンドを実行しようとすると、エラー・メッセージ `BFGCL0502E: 要求された操作を実行する権限がありません。` を受け取ります。コマンドは実行されません。

z/OS z/OS システムでは、マイグレーション・コマンドを実行するためには、ユーザーは以下の条件の (少なくとも) 1 つを満たす必要があります。

- `mqm` グループのメンバーである (システムに `mqm` グループが定義されている場合)。
- **V9.0.0.1** `BFG_GROUP_NAME` 環境変数で名前が指定されたグループのメンバーである (名前が指定されている場合)。
- **V9.0.0.1** コマンド実行時に `BFG_GROUP_NAME` 環境変数に値が設定されていない。

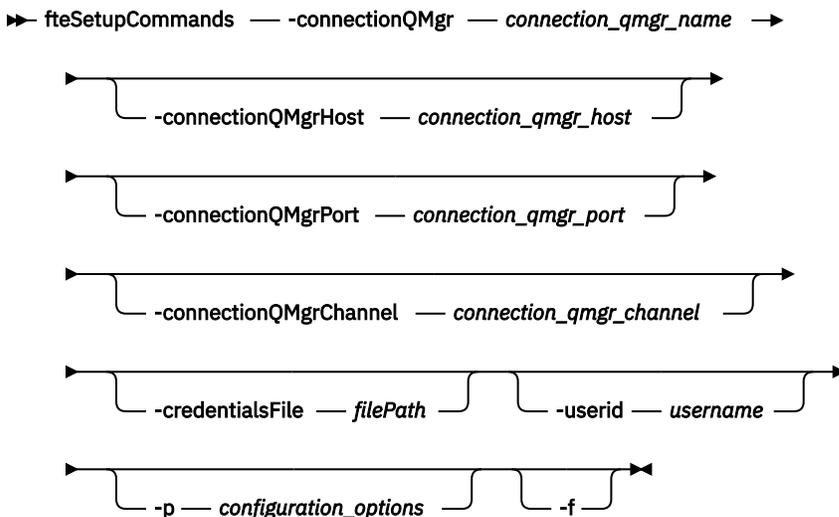
目的

fteSetupCommands コマンドは、調整キュー・マネージャー構成ディレクトリーに `command.properties` ファイルを作成するために使用します。このコマンドは `MQ_DATA_PATH` 環境変数および `installation.properties` ファイルを使用して、`command.properties` ファイルを見つける場所を判別します。**fteSetupCommands** コマンドを発行する前に、調整キュー・マネージャーを既に作成および構成してあることを確認してください。

プロパティー・ファイルについて詳しくは、546 ページの『MFT `command.properties` ファイル』を参照してください。

構文

fteSetupCommands



Parameters

-connectionQMGr (*connection_qmgr_name*)

必須。IBM MQ ネットワークに接続してコマンドを発行するために使用されるキュー・マネージャーの名前。

-connectionQMGrHost (*connection_qmgr_host*)

オプション。接続キュー・マネージャーのホスト名または IP アドレス。

-connectionQMGrHost パラメーターを指定しなかった場合、バインディング・モードでの接続が想定されます。このため、クライアント・モードでの接続を使用する場合、このパラメーターは必須です。

-connectionQMGrHost パラメーターに値を指定し、**-connectionQMGrPort** および **-connectionQMGrChannel** プロパティーには値を指定しない場合、ポート番号 1414 およびチャンネル SYSTEM.DEF.SVRCONN がデフォルトで使用されます。

-connectionQMGrPort (*connection_qmgr_port*)

オプション。クライアント・モードの接続キュー・マネージャーとの接続に使用されるポート番号。

-connectionQMGrPort パラメーターを指定した場合には、**-connectionQMGrHost** パラメーターも指定する必要があります。

-connectionQMGrChannel (*connection_qmgr_channel*)

オプション。接続キュー・マネージャーとの接続に使用されるチャンネル名。

-connectionQMGrChannel パラメーターを指定した場合には、**-connectionQMGrHost** パラメーターも指定する必要があります。

-p (*configuration_options*)

オプション。このパラメーターは、コマンド・キュー・マネージャーをセットアップするために使用される構成オプションのセットを決定します。このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、**fteSetupCommands** コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティー・ファイルのセットを使用します。

このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。

-credentialsFile (filePath)

オプション。IBM MQ 認証の詳細の追加先となる、既存または新規の資格情報ファイルの絶対ファイル・パス。

このコマンドは、指定した Managed File Transfer 資格情報ファイルに IBM MQ 認証の詳細のセットを追加する操作をサポートします。このコマンドは、IBM MQ 接続認証を使用可能にしたときに使用してください。既存の詳細を更新した場合は、強制パラメーター **-f** を使用する必要があります。

-userid (username)

オプション。資格情報の詳細を関連付けるために使用するユーザー ID。ユーザー ID を指定しない場合、資格情報の詳細はすべてのユーザーに適用されます。**-credentialsFile** パラメーターも一緒に指定する必要があります。

-f

オプション。既存の `command.properties` ファイルを、このコマンドで指定された詳細で強制的に上書きします。

-? または -h

オプション。コマンド構文を表示します。

例

```
fteSetupCommands -connectionQMGr QM_NEPTUNE -connectionQMGrHost 9.146.157.241  
-connectionQMGrPort 1414 -connectionQMGrChannel SYSTEM.DEF.SVRCONN
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

関連資料

546 ページの『MFT command.properties ファイル』

`command.properties` ファイルは、コマンド発行時に接続先となるコマンド・キュー・マネージャーと、そのキュー・マネージャーに接続するために Managed File Transfer が必要とする情報を指定します。

514 ページの『fteSetupCoordination』

fteSetupCoordination コマンドは、Managed File Transfer のプロパティ・ファイルと調整キュー・マネージャー・ディレクトリーを作成します。

fteSetupCoordination

fteSetupCoordination コマンドは、Managed File Transfer のプロパティ・ファイルと調整キュー・マネージャー・ディレクトリーを作成します。

重要: **ULW** IBM MQ for UNIX, Linux, and Windows では、IBM MQ 管理者 (および `mqm` グループのメンバー) であるユーザーのみが、このコマンドを実行できます。IBM MQ 管理者以外のユーザーとしてこのコマンドを実行しようとする、エラー・メッセージ `BFGCL0502E: 要求された操作を実行する権限がありません。` を受け取ります。コマンドは実行されません。

z/OS z/OS システムでは、マイグレーション・コマンドを実行するためには、ユーザーは以下の条件の (少なくとも) 1 つを満たす必要があります。

- `mqm` グループのメンバーである (システムに `mqm` グループが定義されている場合)。
- **V9.0.0.1** `BFG_GROUP_NAME` 環境変数で名前が指定されたグループのメンバーである (名前が指定されている場合)。
- **V9.0.0.1** コマンド実行時に `BFG_GROUP_NAME` 環境変数に値が設定されていない。

目的

fteSetupCoordination コマンドは、以下の Managed File Transfer オブジェクトを作成するために使用します。

- 調整キュー・マネージャーのディレクトリー
- データ・ディレクトリー mqft (存在しない場合)
- installation.properties ファイル
- coordination.properties ファイル

さらに、このコマンドは以下の MQSC コマンドを提供します。これを調整キュー・マネージャーに対して実行し、Managed File Transfer を構成する必要があります。MQSC コマンドは、トピック、トピック・ストリング、SYSTEM.FTE キュー、およびデフォルト・データベース・ロガーのキューを作成します。また、これらのコマンドは名前リストを更新して、調整キュー・マネージャーの PSMODE 属性を ENABLED に設定します。

z/OS 調整キュー・マネージャーが z/OS 上にある場合、これらの MQSC コマンドを実行する前に、以下の必須オブジェクトが既に存在することを確認する必要があります。

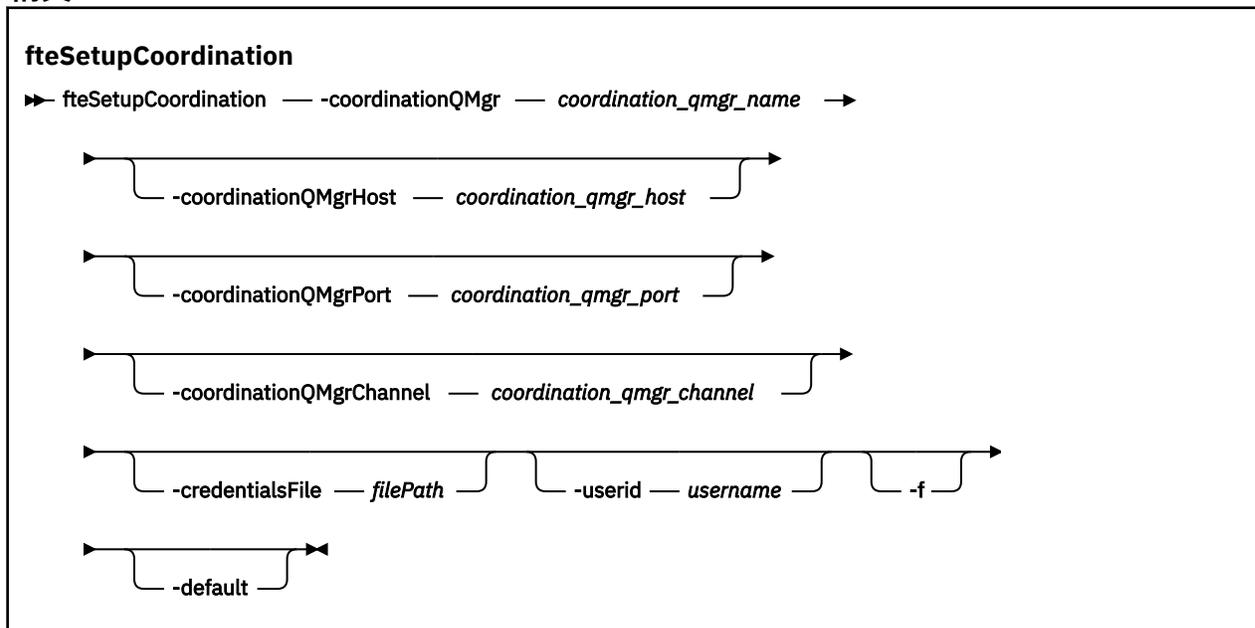
- SYSTEM.BROKER.DEFAULT.STREAM キュー
- SYSTEM.QPUBSUB.QUEUE.NAMELIST 名前リスト
- SYSTEM.BROKER.DEFAULT.STREAM および SYSTEM.BROKER.ADMIN.STREAM ストリーム

```
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 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)
```

プロパティ・ファイルについて詳しくは、[構成オプション](#)を参照してください。

z/OS z/OS を使用している場合は、Managed File Transfer コマンド・テンプレート PDSE ライブラリー・データ・セットから生成したスクリプトを使用して、JCL から **fteSetupCoordination** コマンドやその他のコマンドを発行することができます。詳しくは、[63 ページの『MFT エージェントまたはロガーのコマンド・データ・セットの作成』](#)を参照してください。

構文



Parameters

-coordinationQMgr (coordination_qmgr_name)

必須。調整キュー・マネージャーの名前。このキュー・マネージャーは IBM WebSphere MQ 7.0 以降のキュー・マネージャーでなければなりません。

-coordinationQMgrHost (coordination_qmgr_host)

オプション。調整キュー・マネージャーのホスト名または IP アドレス。

-coordinationQMgrHost パラメーターを指定しなかった場合、バインディング・モードでの接続が想定されます。

-coordinationQMgrHost パラメーターに値を指定し、**-coordinationQMgrPort** および **-coordinationQMgrChannel** パラメーターには値を指定しない場合、ポート番号 1414 およびチャンネル SYSTEM.DEF.SVRCONN がデフォルトで使用されます。

-coordinationQMgrPort (coordination_qmgr_port)

オプション。調整キュー・マネージャーとのクライアント接続で使用されるポート番号。

-coordinationQMgrPort パラメーターを指定した場合には、**-coordinationQMgrHost** パラメーターも指定する必要があります。

-coordinationQMgrChannel (coordination_qmgr_channel)

オプション。調整キュー・マネージャーとの接続に使用されるチャンネル名。

-coordinationQMgrChannel パラメーターを指定した場合には、**-coordinationQMgrHost** パラメーターも指定する必要があります。

-credentialsFile (filePath)

オプション。IBM MQ 認証の詳細の追加先となる、既存または新規の資格情報ファイルの絶対ファイル・パス。

このコマンドは、指定した Managed File Transfer 資格情報ファイルに IBM MQ 認証の詳細のセットを追加する操作をサポートします。このコマンドは、IBM MQ 接続認証を使用可能にしたときに使用してください。既存の詳細を更新した場合は、強制パラメーター **-f** を使用する必要があります。

-userid (username)

オプション。資格情報の詳細を関連付けるために使用するユーザー ID。ユーザー ID を指定しない場合、資格情報の詳細はすべてのユーザーに適用されます。 **-credentialsFile** パラメーターも一緒に指定する必要があります。

-f

オプション。既存の調整キュー・マネージャー構成を、このコマンドで指定された詳細で強制的に上書きします。

-デフォルト

オプション。デフォルト構成オプションを、このコマンドで指定された調整キュー・マネージャーに関連付けられているオプションに更新します。

-? または -h

オプション。コマンド構文を表示します。

例

以下の例では、必要なオブジェクトが、クライアント・モードで接続される QM_SATURN という調整キュー・マネージャー用にセットアップされます。

```
fteSetupCoordination -coordinationQMGr QM_SATURN
-coordinationQMGrHost myhost.ibm.com -coordinationQMGrPort 1415
-coordinationQMGrChannel SYSTEM.DEF.SVRCONN
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

関連概念

[57 ページの『MFT 構成オプション \(Multiplatforms\)』](#)

Managed File Transfer は、セットアップに関する重要な情報が含まれた、操作に必要ないくつかのプロパティ・ファイルを提供します。これらのプロパティ・ファイルは、製品のインストール時に定義される構成ディレクトリーにあります。

[70 ページの『z/OS での MQMFTCredentials.xml の構成』](#)

Managed File Transfer (MFT) がセキュリティーを有効にして構成されている場合、接続認証では、ユーザー ID とパスワードの資格情報を提供するために、すべての MFT エージェント、およびキュー・マネージャーに接続するコマンドが必要になります。

関連タスク

[98 ページの『MFT 用の調整キュー・マネージャーの構成』](#)

fteSetupCoordination コマンドを実行したら、`MQ_DATA_PATH/mqft/config/coordination_qmgr_name` ディレクトリー内の `coordination_qmgr_name.mqsc` スクリプトを実行して、調整キュー・マネージャーに関する必要な構成を行います。ただし、この構成を手動で行う場合は、調整キュー・マネージャーに対して以下の手順を実行します。

関連資料

[550 ページの『MFT agent.properties ファイル』](#)

各 Managed File Transfer Agent には、独自のプロパティ・ファイル `agent.properties` があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。 `agent.properties` ファイルには、エージェントの動作を変更するプロパティを含めることもできます。

関連情報

[607 ページの『coordination.properties ファイルの SSL プロパティ』](#)

fteShowAgentDetails

fteShowAgentDetails コマンドは、特定の Managed File Transfer Agent の詳細を表示するために使用します。これらは、エージェントの Managed File Transfer 調整キュー・マネージャーによって保管される詳細です。

目的

fteShowAgentDetails コマンドは、調整キュー・マネージャーに接続できる任意のシステムから実行できます。このコマンドは、`coordination.properties` ファイルを使用して、調整キュー・マネージャーに接続します。

オプションの **-p** パラメーターは、デフォルト・セットと異なる構成オプションのセットを使用する場合にのみこのコマンドに指定します。詳しくは、[57 ページの『MFT 構成オプション \(Multiplatforms\)』](#)を参照してください。

このコマンドで生成されるエージェント状況情報は、エージェントが SYSTEM.FTE トピックにパブリッシュする状況メッセージから生成されます。これらのメッセージについては、[734 ページの『MFT エージェント状況メッセージ・フォーマット』](#)で説明しています。**fteShowAgentDetails** コマンドで生成される状況情報を調べれば、最後の状況メッセージがパブリッシュされた時点でのエージェントの状況を確認できます。それらの状況メッセージの頻度は、**agentStatusPublishRateLimit** プロパティの値によって決まります。詳しくは、[550 ページの『MFT agent.properties ファイル』](#)を参照してください。

IBM WebSphere MQ 7.5.0 Fix Pack 1 以降の場合、ローカル・エージェントに関する診断情報を確認するには、このコマンドのオプションの **-d** パラメーターを指定します。この情報には、現行の転送、スケジュール済み転送、モニター、およびエージェント・キュー項目数が含まれます。これらの情報を使用して、ローカル・エージェントの正常性と状況を判断できます。

 z/OS の場合、IBM MQ 9.0.2 および IBM MQ 9.0.0 Fix Pack 1 以降では、**fteShowAgentDetails** コマンドが以下によって実行される場合にのみ、**-d** パラメーターを指定できます。

- エージェント・プロセスを実行しているものと同じユーザー ID。
- エージェント・プロパティ **adminGroup** で指定されたグループのメンバー。

詳しくは、**adminGroup** プロパティ ([550 ページの『MFT agent.properties ファイル』](#)) を参照してください。

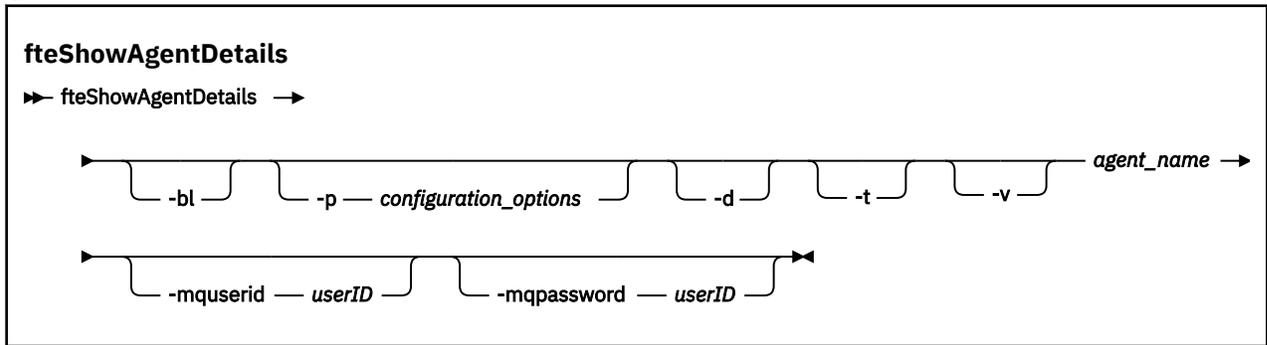
IBM WebSphere MQ 7.5 以降の場合、このコマンドをエージェントと同じシステムで実行すると、エージェント・プロセス・コントローラーおよびキュー・マネージャーの状況を確認できます。これらの情報は、問題判別に役立てることができます。また、このコマンドをエージェントと同じシステムで実行すると、エージェントが予期せずに終了した場合に、より詳細なエージェント状況情報を確認できます。

有効なエージェント状況値とそれぞれの意味をまとめたリストについては、[624 ページの『MFT エージェントの状況値』](#)を参照してください。

エージェント・プロセス・コントローラーの有効な状況値とそれぞれの意味をまとめたリストについては、[627 ページの『MFT エージェント・プロセス・コントローラーの状況値』](#)を参照してください。

エージェントのトレースの値および FFDC 仕様とそれぞれの意味をまとめたリストについては、[273 ページの『fteSetAgentTraceLevel』](#)を参照してください。

構文



パラメーター

-bl

オプション。追加でエージェントの製品ビルド・レベルを出力します。

-p (*configuration_options*)

オプション。このパラメーターは、エージェントの詳細を表示する要求を発行するために使用される構成オプションのセットを決定します。このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティ・ファイルのセットを使用します。

このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。

-d

オプション。 *agent_name* の診断情報を収集することを指定します。

診断情報はコンソールに出力され、

MQ_DATA_PATH\mqft\logs\coordination_qmgr_name\agents\agent_name\logs ディレクトリーの *diagnostics.<yyyyMMdd>.<HHmmss>.<ssss>.<number>.properties* というファイルに書き出されます。エージェントの診断情報が含まれている履歴・ファイルが最大で 5 個作成されます。 **-d** パラメーターを指定して **fteShowAgentDetails** コマンドを実行したときに、エージェントの履歴・ファイルが既に 5 個作成されている場合は、一番古い履歴・ファイルが削除され、そのエージェントの最新の診断情報を含む新しいファイルに置き換えられます。

このパラメーターは、エージェントがローカル・システムで実行されている場合のみ使用できます。

-t

オプション。簡潔モードを指定します。IBM MQ 9.0.2 では、**Status Age** 情報がデフォルトで出力に含まれます。この情報を表示しないようにするには、 **-t** パラメーターを指定してコマンドを発行します。詳しくは、304 ページの『エージェントが UNKNOWN 状態にあると表示される場合の処理』を参照してください。

-v

オプション。冗長モードを指定します。これにより、エージェントに関する追加出力が生成されます。これには、ホスト名、製品バージョン、製品ビルド・レベル、トレース・レベル、初期障害データ・キャプチャー機能 (FFDC) 仕様、および現在のソース転送と宛先転送ごとの転送状態のリストが含まれます。

現在の転送情報は、734 ページの『MFT エージェント状況メッセージ・フォーマット』で説明されている、エージェント状況のパブリケーションから取得されます。したがって、この転送情報が正確なのは、 *agentStatusPublishRateLimit* プロパティの値の範囲内に限られます。このプロパティに関して詳しくは、550 ページの『MFT agent.properties ファイル』を参照してください。

agent_name

必須。表示する Managed File Transfer Agent の名前。

-mquserid (userID)

オプション。調整キュー・マネージャーで認証するユーザー ID を指定します。

-mqpassword (password)

オプション。調整キュー・マネージャーで認証するパスワードを指定します。 **-mquserid** パラメーターも一緒に指定する必要があります。 **-mquserid** を指定した場合に **-mqpassword** を指定しないと、関連付けるパスワードを入力するというプロンプトが出されます。パスワードは表示されません。

-? または -h

オプション。コマンド構文を表示します。

例

次の例では、バインディング・エージェントが実行中であり、そのエージェントに対してローカルに **fteShowAgentDetails** コマンドが発行されています。

```
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
  414d51204d49414f5720202020202020202020822c5b4a648c0b20 progress
  414d51204d49414f5720202020202020202020822c5b4a346c0b20 progress
```

次の例では、QMGR1 は、構成オプションに対する入力として使用されるデフォルト以外の調整キュー・マネージャーであり、診断情報は **-d** パラメーターによって要求されます。 **fteShowAgentDetails** コマンドは、ローカル・エージェントを持つ IBM WebSphere MQ 7.5.0 Fix Pack 1 システムで発行されます。

```
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) -
414d5120514d43414e444520202020202079e20f5064230010
  Last Internal Command Time: 2012-07-30T14:17:10.506Z
  Last External Command Type: New Monitor Request
  Last External Command Time: 2012-07-30T14:10:57.751Z
  Diagnostic Properties File name: C:\Program Files (x86)\IBM\WebSphere
MQ\mqft\logs\MUNGEE\agents\MUNGEE\logs\di
agnostics.20121031.083420.0477.1.properti
es

Command Handler Worker Thread 0 Diagnostics:
  Status:              Waiting

Command Handler Worker Thread 1 Diagnostics:
  Status:              Waiting

Command Handler Worker Thread 2 Diagnostics:
  Status:              Waiting

Command Handler Worker Thread 3 Diagnostics:
  Status:              Waiting

Command Handler Worker Thread 4 Diagnostics:
  Status:              Waiting

File Transfer Diagnostics:
  Source Transfers:    1
  Destination Transfers: 2

File Transfer 0 Diagnostics:
  Transfer Id:         414d5120514d43414e4445202020202079e20f5064230010
  Role:                SOURCE
  State:               ReSynchronisingTransfer
  Status:              INACTIVE
  Start Time:         Not started
```

```

Retry Count: 0
CheckPoint Index: 0
CheckPoint Position: 0

File Transfer 1 Diagnostics:
  Transfer Id: 414d5120514d43414e44452020202020c8fbd54f144f0d20
  Role: DESTINATION
  State: RunningTransfer
  CheckPoint Index: 0
  CheckPoint Position: 0
  Write Index: 0
  Write Position: 0

File Transfer 2 Diagnostics:
  Transfer Id: 414d5120514d43414e4445202020202079e20f5086020010
  Role: DESTINATION
  State: RunningTransfer
  CheckPoint Index: 9
  CheckPoint Position: 0
  Write Index: 3
  Write Position: 140923

Monitor 0 Diagnostics:
  Name: MONITOR1
  Status: STARTED
  Resource Type: directory
  Resource: /tmp/monitor
  Poll Interval: 1 minutes
  Batch Size: 2
  Condition: Match
  Pattern: * (wildcard)
  Executing: false
  Last Execute Start Time: 2012-04-04T16:19:01.852Z
  Last Execute End Time: 2012-04-04T16:19:01.852Z
  Last Execute Match Count: 0

Schedule 1 Diagnostics:
  Id: 1
  Next Trigger Time: 2012-07-17T16:00+0100
  Occurrences So Far: 14
  Repeat Interval: hours
  Repeat Frequency: 5
  Source Agent: AGCANDE
  Destination Agent: AGCANDE
  Source File: /tmp/source/a.txt, ...
  Destination File: /tmp/dest/a.txt, ...

```

次の例では、バインディング・エージェントが停止し、そのエージェントからリモートに **fteShowAgentDetails** コマンドが発行されています。

```

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.
```

次の例では、エージェント・キュー・マネージャーが停止した状態で、バインディング・エージェントは再始動を待機しています。Total Agent Restart Count: 1 より前にエージェントが既に1回再始動されています。これは、前のエージェント・キュー・マネージャーの再始動が原因である可能性があります。

注: キュー・マネージャー情報については、Last Status Reported に対する Last Error MQRC。この情報は、キュー・マネージャーが使用可能になっても残ります。

```
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.
```

次の例では、クライアント・モード・エージェントが予期せずに終了し、エージェント・プロセス・コントローラーが maxRestartDelay エージェント・プロパティー値で指定された遅延の後で再始動することにより、状態の回復を試みます。デフォルトの maxRestartDelay エージェント・プロパティー値は -1 です。この値によって、エージェント・プロセス・コントローラーが終了します。そのため、この例では、maxRestartDelay プロパティー値に 0 より大きい値が設定されている必要があります。Current Agent Restart Count: 4 は、maxRestartInterval エージェント・プロパティーの期間中に 4 回の再始動があったことを意味します。maxRestartCount エージェント・プロパティーが 4 である場合には、maxRestartInterval 内で 4 回の再始動があった後で、エージェント・プロセス・コントローラーは maxRestartDelay 秒待機してからエージェントを再起動します。これがこの例に該当します。Total Agent restart Count: 8 によって、これが以前にも起こっていることが示唆されます。この例は典型的なケースではありません。エージェントの予期しない終了は、エージェントがメモリー不足になった場合や、カスタム・ユーザー出口で何らかのランタイム・エラーが発生した場合にのみ発生することが予想されます。エージェントが予期せずに終了する理由についての詳細は、各エージェントの output0.log ファイルに示されます。

```
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

```

次の例では、Connect:Direct ブリッジ・エージェントの結果が表示されます。

```

fteShowAgentDetails AG_CD1
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
Agent Information:
Name: AG_CD1
Type: Connect:Direct bridge
Description:
Connect:Direct Node Name: CDNODE
Connect:Direct Node Host: localhost:1363
Operating System: Windows Server 2003
Time Zone: Greenwich Mean Time

Agent Controller Information:
Controller type: MQMFT Process Controller
Status: UNKNOWN
Status Details: Information about the agent controller
is not available, either because the
agent is not running or the agent is
running on a different system.

Agent Restarts within Interval: 0
Total Agent Restart Count: 0

Agent Availability Information:
Status: STOPPED
Status Details: The agent has been stopped. It was shut
down in a controlled manner.

Queue Manager Information:
Name: QM_JUPITER
Transport: Bindings
Last Status Reported: UNKNOWN
Status Details: Information about the queue manager is
not available, either because the agent
is not running or the agent is running
on a different system.

```

z/OS 次の例では、z/OS 上で稼働するエージェントを自動リスタート・マネージャー (ARM) に登録します。

```

fteShowAgentDetails AGENTZ
5724-H72 Copyright IBM Corp. 2008, 2023. ALL RIGHTS RESERVED
Agent Information:
Name: AGENTZ
Type: Standard
Description:
Operating System: z/OS
Time Zone: Greenwich Mean Time

Agent Controller Information:
Controller Type: z/OS Automatic Restart Manager (ARM)
Agent registered with ARM: Yes (ELEMTYPE: SYSBFGAG, ELEMENT: AGENTZ)
Agent Restarted: No

Agent Availability Information:
Status: READY
Status Details: The agent is running and is publishing
its status at regular intervals. The last

```

```
update was received within the expected
time period. The agent is ready to
process transfers, but none are currently
in progress.
```

Queue Manager Information:

```
Name: ZQM
Transport: Bindings
Last Status Reported: AVAILABLE
Status Details: The queue manager is available.
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

関連資料

[473 ページの『fteListAgents: 調整キュー・マネージャーの MFT エージェントのリスト』](#)

fteListAgents コマンドは、特定の調整キュー・マネージャーに登録されているすべての Managed File Transfer のエージェントをリストするために使用します。

[624 ページの『MFT エージェントの状況値』](#)

fteListAgents と **fteShowAgentDetails** のいずれかのコマンドを実行すると、エージェント状況情報が生成されます。この状況として生成される可能性がある値は、以下のとおりです。

[627 ページの『MFT エージェント・プロセス・コントローラーの状況値』](#)

fteShowAgentDetails コマンドは、エージェント・プロセス・コントローラーの状況情報を生成します。この状況として生成される可能性がある値は、以下のとおりです。

fteShowLoggerDetails

fteShowLoggerDetails コマンドは、特定の Managed File Transfer ロガーの詳細を表示するために使用します。

目的

fteShowLoggerDetails コマンドは、ロガーと同じシステム上で実行する必要があります。これにより、ロガー・プロセス・コントローラーおよびロガー・キュー・マネージャーの状況が表示されます。これは、問題判別に役立ちます。**fteShowLoggerDetails** コマンドは、特定の Managed File Transfer ロガーに関する以下の詳細をリストします。

- ロガー・コントローラーの状況。
- 特定の間隔におけるロガーの再開数
- ロガーの再開の合計数
- ロガーの可用性状況
- ロガーのキュー・マネージャー名
- ロガーのキュー・マネージャー・トランスポート・タイプ
- ロガー・キュー・マネージャーの最後に報告された状況 (バインディング・トランスポート・モードにのみ適用されます)

オプションの **-p** パラメーターは、デフォルト・セットと異なる構成オプションのセットを使用する場合にのみこのコマンドに指定します。詳しくは、[『57 ページの『MFT 構成オプション \(Multiplatforms\)』』](#)を参照してください。

有効なロガー状況値とそれぞれの意味をまとめたリストについては、[628 ページの『MFT ロガーの状況値』](#)を参照してください。

ロガー・プロセス・コントローラーの有効な状況値とそれぞれの意味をまとめたリストについては、[628 ページの『MFT ロガー・プロセス・コントローラーの状況値』](#)を参照してください。

V 9.0.4 IBM MQ 9.0.4 以降では、コマンドの出力にはロガーがキュー・マネージャーへの接続に使用している接続情報が表示されます。

V 9.0.4 ロガーがクライアント・モードで接続されている場合、各出力は以下のとおりです。

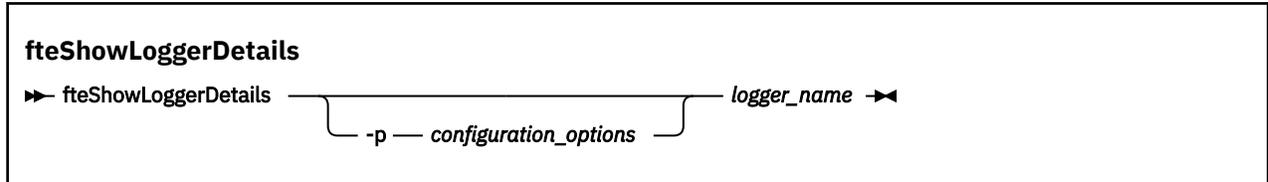
報告された最後の状況

UNKNOWN として表示されます

状況の詳細

Information about the queue manager is not available because the logger has a client connection to queue manager. として表示されます

構文



パラメーター

-p configuration_options

オプション。このパラメーターは、ロガーの詳細を表示する要求を発行するために使用される構成オプションのセットを決定します。このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティ・ファイルのセットを使用します。

このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。

logger_name

必須。表示する Managed File Transfer ロガーの名前。

-? または -h

オプション。コマンド構文を表示します。

例

次の例では、開始されたロガーに対してローカルで **fteShowLoggerDetails** コマンドが発行されています。

```
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.
```

次の例では、キュー・マネージャーが使用不可であることが原因で待機しているログーに対して、ローカルで **fteShowLoggerDetails** コマンドが発行されています。

```
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.
```

z/OS における次の例では、実行中のログー (ARM に未登録) を示します。

```
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.
```

z/OS における次の例では、実行されていない、または別のシステムで実行中のログーを示します。

```
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.
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

関連資料

628 ページの『MFT ロガーの状況値』

fteShowLoggerDetails コマンドは、ロガーの状況情報を生成します。この状況として生成される可能性がある値は、以下のとおりです。

628 ページの『MFT ロガー・プロセス・コントローラーの状況値』

fteShowLoggerDetails コマンドは、ロガー・プロセス・コントローラーの状況情報を生成します。この状況として生成される可能性がある値は、以下のとおりです。

fteStartAgent

fteStartAgent コマンドは、Managed File Transfer のエージェントをコマンド行から開始します。

目的

fteStartAgent コマンドは、Managed File Transfer エージェントを開始するために使用します。エージェントを使用してファイル転送を実行するには、まずエージェントを開始する必要があります。

fteStartAgent コマンドは、コマンドを発行するシステムのエージェントを開始します。リモート・システムのエージェントを開始することはできません。

IBM WebSphere MQ 7.5 以降の場合、エージェント・プロセス・コントローラーがエージェントの始動を管理します。ただし、エージェント・プロセス・コントローラーは、エージェントの障害が高い比率で発生している場合などには、一定期間待機してから、エージェントの始動を再試行します。IBM MQ 管理者は、**fteStartAgent** コマンドを使用してこの待機を無効にし、エージェントの始動を開始できます。キュー・マネージャーが使用可能になるのをエージェント・プロセス・コントローラーが待機している場合は、このコマンドによって、エージェント・プロセス・コントローラーによるキュー・マネージャーへの再接続の試みも開始されます。

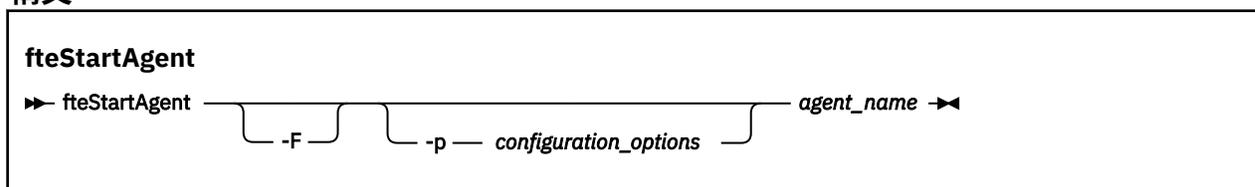
Windows

fteCreateAgent または **fteModifyAgent** コマンドを使用して Windows サービスとして実行するようにエージェントを構成した場合、**fteStartAgent** コマンドを実行すると Windows サービスが開始されます。

エージェントが開始しない、または既に開始済みの場合、このコマンドはエラーを戻します。エージェントは、`agent.properties` ファイルに定義されている値に基づいてキュー・マネージャーと通信します。

オプションの **-p** パラメーターは、デフォルト・セットと異なる構成オプションのセットを使用する場合にのみ、このコマンドに指定します。詳しくは、『550 ページの『MFT agent.properties ファイル』』を参照してください。

構文



パラメーター

-F

オプション。このパラメーターは、エージェント・デーモンをフォアグラウンド・プロセスとして実行します。デフォルトで、エージェント・デーモンはバックグラウンドで実行されます。

Windows 上で作業中であり、**fteCreateAgent** または **fteModifyAgent** コマンドを使用して、エージェントを Windows サービスとして実行するように構成した場合、**-F** パラメーターはこの構成をオーバーライドします。

-p configuration_options

オプション。このパラメーターは、エージェントを開始する要求を発行するために使用される構成オプションのセットを決定します。このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティ・ファイルのセットを使用します。

このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。

agent_name

必須。開始する Managed File Transfer エージェントの名前。

-? または -h

オプション。コマンド構文を表示します。

例

この例では、AGENT2 がフォアグラウンドで開始および実行されます。

```
fteStartAgent -F AGENT2
```

以下の例 (UNIX および Linux システムの場合) では、AGENT2 が非デフォルト調整キュー・マネージャー QM_SATURN により開始されます。

```
./fteStartAgent -p QM_SATURN AGENT2
```

このコマンドは、**fteStartAgent** へのパスを次のように指定しても実行することができます。

```
path/fteStartAgent agentname
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

応答

場合によっては、**fteStartAgent** コマンドを実行した後、以下のようなエラー・メッセージが表示されます。

- **fteStartAgent** コマンドを実行して以下のエラー・メッセージが表示された場合、Managed File Transfer と競合する他のライブラリー・パスが、ご使用の環境に含まれている可能性があります。

BFGCL0001E: 内部エラーが発生しました。 例外: 'CC=2;RC=2495;AMQ8568:
ネイティブ JNI ライブラリー「mqjbnd」が見つかりませんでした。 [3=mqjbnd]

LD_LIBRARY_PATH または LIBPATH 環境変数に、64 ビット版ライブラリーを 32 ビット版ライブラリーより先に参照するように設定している場合に、(現在、ほとんどのプラットフォームで使用されている) 32 ビット版の Java を使用してエージェントを実行すると、このエラーが発生します。

この問題を解決するには、Managed File Transfer エージェントのプロパティ `javaLibraryPath` を、ライブラリーの正しい場所を参照するように設定します。例えば、AIX の mqjbnd の場合、`/usr/mqm/java/lib` と設定します。Linux の mqjbnd の場合、`/opt/mqm/java/lib` に設定します。

関連タスク

[155 ページの『Windows サービスとしての MFT エージェントの開始』](#)

エージェントを Windows サービスとして開始することにより、Windows からログオフしても、引き続きエージェントを実行し、ファイル転送を受け取ることができます。

[212 ページの『MFT エージェントのリスト』](#)

特定のキュー・マネージャーに登録された Managed File Transfer エージェントは、コマンド行または IBM MQ Explorer を使用してリストできます。

[212 ページの『MFT エージェントの停止』](#)

Managed File Transfer エージェントはコマンド行から停止できます。エージェントを停止するときには、停止する前にエージェントを静止させて、エージェントが現行のファイル転送を完了するようにします。さらに、コマンド行で **-i** パラメーターを指定して、エージェントをただちに停止することもできます。エージェントが停止してしまうと、再始動するまでそのエージェントを使用してファイルを転送することはできません。

関連資料

[159 ページの『z/OS での MFT エージェントの開始』](#)

z/OS でエージェントを実行するには、UNIX System Services セッションから **fteStartAgent** コマンドを実行する方法に加えて、JCL から開始済みタスクとして開始する方法もあり、この場合には対話式セッションを必要としません。

fteStartLogger

fteStartLogger コマンドは、Managed File Transfer ロギング・アプリケーションを開始します。

目的

fteStartLogger コマンドは、ロガーを開始するために使用します。調整キュー・マネージャーと同じシステム上で実行されるファイル・アプリケーションまたはデータベース・アプリケーションをロガーにすることができます。詳しくは、[106 ページの『MFT ロガーの構成』](#)のトピックを参照してください。IBM WebSphere MQ 7.5 以降の場合、ロガー・プロセス・コントローラーがロガーの始動を管理します。ただし、ロガー・プロセス・コントローラーは、ロガーの障害が高い比率で発生している場合などには、一定期間待機してから、ロガーの始動を再試行します。IBM MQ 管理者は、**fteStartLogger** コマンドを使用してこの待機を無効にし、ロガーの始動を開始できます。キュー・マネージャーが使用可能になるのをロガー・プロセス・コントローラーが待機している場合は、このコマンドによって、ロガー・プロセス・コントローラーによるキュー・マネージャーへの再接続の試みも開始されます。

fteModifyLogger コマンドを使用して Windows サービスとして実行するようにロガーを構成した場合、**fteStartLogger** コマンドを実行すると、Windows サービスが開始されます。

ロガーが開始しない、または既に開始済みの場合、このコマンドはエラーを戻します。ロガーは、`logger.properties` ファイルに定義されている値に基づいてキュー・マネージャーと通信します。

-p パラメーターは、デフォルトと異なる構成オプションのセットを使用する場合にのみこのコマンドに指定します。ロガー・プロパティについて詳しくは、[118 ページの『MFT ロガーの構成プロパティ』](#)を参照してください。

構文

fteStartLogger

```
►► fteStartLogger -p configuration_options -F logger_name ◄◄
```

Parameters

logger_name

必須。開始する Managed File Transfer ロガーの名前。

-p configuration_options

オプション。このパラメーターは、ロガーを開始する要求を発行するために使用される構成オプションのセットを決定します。このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、**fteStartLogger** は、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティ・ファイルのセットを使用します。

このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。

-F

オプション。ロガーをフォアグラウンド・プロセス (デフォルトのバックグラウンド・プロセスではない) として実行します。**fteModifyLogger** コマンドを使用して、ロガーを Windows サービスとして実行するように構成した場合、**-F** パラメーターはこの構成をオーバーライドします。

-? または -h

オプション。コマンド構文を表示します。

例

この例では、ロガーは前もって `logger1` という名前で作成されています。次のコマンドは、このロガーをフォアグラウンド・プロセスとして開始する方法を示しています。

```
fteStartLogger -F logger1
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

関連概念

[106 ページの『MFT ロガーの構成』](#)

関連資料

[492 ページの『fteModifyLogger \(Windows サービスとしての MFT ロガーの実行\)』](#)

Managed File Transfer ロガーを Windows サービスとして実行できるように変更するには、**fteModifyLogger** コマンドを使用します。このコマンドは Windows プラットフォームでのみ使用できます。IBM MQ 管理者であり、mqm グループのメンバーであるユーザーが実行する必要があります。まず、**fteStopLogger** コマンドを使用してロガーを停止する必要があります。

[533 ページの『fteStopLogger』](#)

fteStopLogger コマンドは Managed File Transfer ロガーを停止します。

[331 ページの『MFT ロガーのエラー処理とリジェクト』](#)

Managed File Transfer ロガーは、メッセージごとのエラーと一般エラーの 2 つのタイプのエラーを特定します。

fteStopAgent

fteStopAgent コマンドは、Managed File Transfer のエージェントを制御された仕方で停止させるか、必要であれば **-i** パラメーターを使用してエージェントを即時に停止させるために使用します。

目的

fteStopAgent コマンドを使用してエージェントを停止すると、停止する前にエージェントが現行のファイル転送を完了することや、エージェントが現在ファイルを転送中であってもエージェントを即時に停止することができます。エージェントが停止してしまうと、再始動するまでそのエージェントを使用してファイルを転送することはできません。

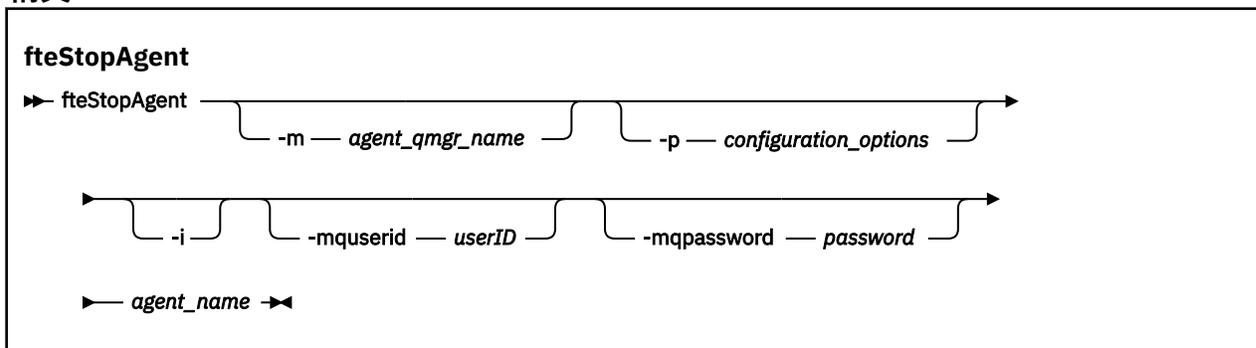
停止するエージェントが IBM MQ ネットワークに接続されている場合は、IBM MQ ネットワークに接続してそのエージェントのキュー・マネージャーまで経路指定できる任意のシステムから、**fteStopAgent** コマンドを実行できます。具体的には、このコマンドを実行する場合、このシステムに Managed File Transfer コンポーネント (Service またはエージェント) をインストールし、IBM MQ ネットワークと通信するように構成しておく必要があります。接続の詳細を使用できない場合は、ローカル・システムのデフォルト・キュー・マネージャーに対してバインディング・モードで接続されます。command.properties が存在しない場合は、エラーが生成されます。

停止するエージェントが IBM MQ ネットワークに接続されていない場合 (例えば、IBM MQ ネットワークが現在は使用不可の場合など) は、そのエージェントが実行されているシステムからのみ **fteStopAgent** コマンドを実行できます。IBM MQ ネットワークに接続されていないエージェントを停止するには、そのエージェントを実行しているユーザーが **fteStopAgent** コマンドを実行する必要があります。または、そのエージェントが Windows システムで実行されている場合には、このコマンドを管理者として実行することもできます。

オプションの **-p** パラメーターは、デフォルト・セットと異なる構成オプションのセットを使用する場合にのみこのコマンドに指定します。詳しくは、『550 ページの『MFT agent.properties ファイル』』を参照してください。

エージェントが Windows サービスとして実行中の場合、**fteStopAgent** コマンドを実行すると、Windows サービスが停止します。詳しくは、155 ページの『Windows サービスとしての MFT エージェントの開始』を参照してください。

構文



Parameters

-m (agent_qmgr_name)

オプション。停止するエージェントの接続先のキュー・マネージャーの名前。

エージェントがリモート・システム上にある場合、またはエージェントがローカル・システムにあってもエージェントを始動したユーザーでない場合は、適切な権限を持った上で **-m** パラメーターを使用する必要があります。権限について詳しくは、353 ページの『MFT 固有リソースのグループ権限の制限』を参照してください。

-p (configuration_options)

オプション。このパラメーターは、エージェントを停止する要求を発行するために使用される構成オプションのセットを決定します。このパラメーターの入力には、デフォルト調整キュー・マネージャー以外の名前を使用します。これで、コマンドは、デフォルト以外の調整キュー・マネージャーと関連付けられたプロパティ・ファイルのセットを使用します。

このパラメーターを指定しない場合、デフォルトの調整キュー・マネージャーに基づいた構成オプションのセットが使用されます。

-i

オプション。即時にエージェントを停止します。エージェントは、現在進行中の転送を完了しません。

-i パラメーターを指定しなかった場合、エージェントは現在進行中の転送は完了しますが、新規転送は開始しません。

-mquserid (userID)

オプション。コマンド・キュー・マネージャーで認証するユーザー ID を指定します。

-mqpassword (password)

オプション。コマンド・キュー・マネージャーで認証するパスワードを指定します。 **-mquserid** パラメーターも一緒に指定する必要があります。 **-mquserid** を指定した場合に **-mqpassword** を指定しないと、関連付けるパスワードを入力するというプロンプトが出されます。パスワードは表示されません。

agent_name

必須。停止する Managed File Transfer エージェントの名前。

-? または -h

オプション。コマンド構文を表示します。

例

この例では、キュー・マネージャー QM_JUPITER 上のエージェント AGENT2 が停止します。AGENT2 が接続するこのキュー・マネージャーは、構成オプションのセットで指定されているものと異なるので、**-m** パラメーターが使用されます。

```
fteStopAgent -m QM_JUPITER AGENT2
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

関連タスク

212 ページの『[MFT エージェントの停止](#)』

Managed File Transfer エージェントはコマンド行から停止できます。エージェントを停止するときには、停止する前にエージェントを静止させて、エージェントが現行のファイル転送を完了するようにします。さらに、コマンド行で **-i** パラメーターを指定して、エージェントをただちに停止することもできます。エージェントが停止してしまうと、再始動するまでそのエージェントを使用してファイルを転送することはできません。

関連資料

528 ページの『[fteStartAgent](#)』

fteStartAgent コマンドは、Managed File Transfer のエージェントをコマンド行から開始します。

212 ページの『[z/OS 上の MFT エージェントの停止](#)』

JCL からの開始タスクとして z/OS 上で Managed File Transfer Agent を実行している場合、エージェントは、**fteStopAgent** コマンドに加えて、z/OS オペレーター・コマンド **MODIFY** および **STOP** を受け入れます。

fteStopLogger

fteStopLogger コマンドは Managed File Transfer ロガーを停止します。

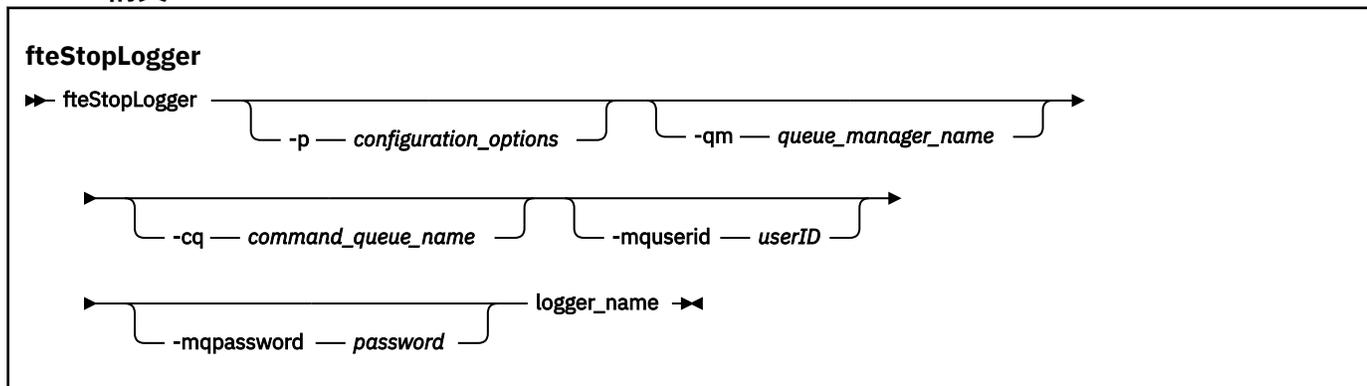
目的

fteStopLogger コマンドは、ロガーを停止するために使用します。ロガーは、管理対象ファイル転送アクティビティの履歴をファイルに記録するファイル・ロガーか、履歴をデータベースに記録するデータベース・ロガーのいずれかになります。

スタンドアロン・ロガー・プロセスの停止に関する追加の注記

ロガーが Windows サービスとして実行中の場合、**fteStopLogger** コマンドを実行すると、Windows サービスが停止します。

構文



Parameters

-p (configuration_options)

オプション。ロガーの停止に使用される構成オプションのセットを決定します。構成オプションのセット名を **-p** パラメーターの値として使用します。慣例として、この値は調整キュー・マネージャーの名前です。このパラメーターを指定しない場合、デフォルトの構成オプションのセットが使用されます。

-qm (queue_manager_name)

オプション。デフォルトでは、ロガーのコマンド・キューは、**-p** パラメーター (またはそのデフォルト) によって指定された調整キュー・マネージャー上にあるという前提です。別の場所にあるコマンド・キューにロガー・コマンドを送信する場合は、**-qm** パラメーターを使用して代替宛先を指定します。すべてのケースにおいて、メッセージの最終的な宛先に関係なく、このコマンドは、**-p** パラメーターによって指定されたコマンド・キュー・マネージャーに接続します。

-cq (command_queue_name)

オプション。停止メッセージの送信先のコマンド・キューを指定します。ほとんどの場合、ロガーはデフォルトのキュー名を使用するので、このパラメーターは必要ありません。

-mquserid (userID)

オプション。コマンド・キュー・マネージャーで認証するユーザー ID を指定します。

-mqpassword (password)

オプション。コマンド・キュー・マネージャーで認証するパスワードを指定します。**-mquserid** パラメーターも一緒に指定する必要があります。**-mquserid** を指定した場合に **-mqpassword** を指定しないと、関連付けるパスワードを入力するというプロンプトが出されます。パスワードは表示されません。

logger_name

必須。停止する Managed File Transfer ロガーの名前。

-? または -h

オプション。コマンド構文を表示します。

例

この例では、ロガーは前もって `logger1` という名前で作成されており、現在実行中です。次のコマンドは、このロガーを停止する方法を示しています。

```
fteStopLogger logger1
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

関連概念

[106 ページの『MFT ロガーの構成』](#)

関連資料

[492 ページの『fteModifyLogger \(Windows サービスとしての MFT ロガーの実行\)』](#)

Managed File Transfer ロガーを Windows サービスとして実行できるように変更するには、**fteModifyLogger** コマンドを使用します。このコマンドは Windows プラットフォームでのみ使用できます。IBM MQ 管理者であり、mqm グループのメンバーであるユーザーが実行する必要があります。まず、**fteStopLogger** コマンドを使用してロガーを停止する必要があります。

[530 ページの『fteStartLogger』](#)

fteStartLogger コマンドは、Managed File Transfer ログイン・アプリケーションを開始します。

MFT 構成リファレンス

MFT プロパティーでの環境変数の使用

IBM WebSphere MQ 7.5 以降、ファイルまたはディレクトリーの場所を表す Managed File Transfer プロパティーで環境変数を使用できます。これにより、製品の一部の実行時に使用されるファイルまたはディレクトリーの場所を、環境の変更に合わせて変えることができます。例えば、プロセスを実行するユーザーの変更などです。

以下のプロパティーは、ファイルまたはディレクトリーの場所を受け入れるため、環境変数を含めることができます。

- 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`

例

Windows システムで、ユーザー `fteuser` が次のように環境変数 `USERPROFILE` を使用したとします。

```
wmqfte.database.credentials.file=%USERPROFILE%\logger\mqmftcredentials.xml
```

これは次のファイル・パスに解決されます。

```
C:\Users\fteuser\logger\mqmftcredentials.xml
```

UNIX システムで、ユーザー `fteuser` が次のように環境変数 `HOME` を使用したとします。

```
transferRoot=$HOME/fte/
```

これは次のファイル・パスに解決されます。

```
/home/fteuser/fte/
```

MFT installation.properties ファイル

`installation.properties` ファイルは、構成オプションのデフォルト・セットの名前を指定します。この項目は、Managed File Transfer を、使用する構成が含まれる一連の構造化されたディレクトリーおよびプロパティ・ファイルに向けます。一般に、構成オプションのセット名は、関連付けられている調整キュー・マネージャーの名前です。

このファイルはインストーラーによって作成され、`fteChangeDefaultConfigurationOptions` コマンドを使用して変更できます。

`installation.properties` ファイルは `MQ_DATA_PATH` ディレクトリーにあります。例えば、Windows の場合、デフォルトのファイルの場所は `MQ_DATA_PATH\mqft\installations\installation_name` で、UNIX および Linux システムの場合、デフォルトのファイルの場所は `/var/mqm/mqft/installations/installation_name` です。

V 9.0.1 Redistributable Managed File Transfer Agent については、データ・パスは `fteCreateEnvironment` コマンドの実行時に設定されます。このコマンドを実行するときに、選択した場所を `-d` パラメーターで指定すると、その場所がデータ・パスに設定されます。

`fteCreateEnvironment` コマンドで場所を指定しない場合は、Redistributable Managed File Transfer Agent が解凍された場所をルート・ディレクトリーとして、その下にディレクトリー `mftdata` が作成されます。Redistributable Managed File Transfer Agent の `installation.properties` ファイルは、`MQ_DATA_PATH\mqft\installations\MFTZipInstall` ディレクトリーにあります。

`installation.properties` ファイルには以下の値が含まれます。

表 39. 基本プロパティー

プロパティー名	説明	デフォルト値
commandMessagePriority	<p>fteStopAgent、fteCancelTransfer、およびftePingAgent コマンドに関する内部メッセージとコマンド・メッセージの両方の優先順位を設定します。</p> <p>例えば、多数の小さいファイルを転送するための大量の転送要求を立て続けに送信する場合、新しい転送要求をソース・エージェントのコマンド・キューで待機状態にすることができます。外部メッセージと内部メッセージにはデフォルトの IBM MQ メッセージ優先順位が設定されていて、内部メッセージは新しい転送要求によってブロックされるようになっていきます。これにより、転送ネゴシエーション時間を超過して、転送がリカバリー状態になることがあります。</p> <p>また、commandMessagePriority プロパティーを使用すると、内部確認応答メッセージと確認応答が必要なメッセージの優先順位を設定することもできます。</p> <p>内部 Managed File Transfer メッセージに新しい転送要求より高い優先順位を設定するには、このプロパティーを 1 (最低の優先順位) から 9 (最高の優先順位) までの値に設定します。</p> <p>commandMessagePriority プロパティーのデフォルト値は 8 です。これは、エージェント・コマンド・キューの IBM MQ 属性 DEFPRTY (デフォルト優先順位) が 7 以下の場合の場合に、内部のネゴシエーション・メッセージが新しい転送要求よりも優先されるという意味です。DEFPRTY 属性の値が 8 または 9 のいずれかに設定されている場合、commandMessagePriority プロパティーの有効性を維持するには、DEFPRTY プロパティーまたは commandMessagePriority プロパティーのいずれかを変更する必要があります。</p>	<p>IBM MQ 9.0.0.0 の場合、デフォルト値は 8 です。</p> <p>以前のリリースと APAR IT06213 より前では、デフォルト値は MQPRI_PRIORITY_AS_Q_DEF 定数です。これは -1 の値です。</p>

表 39. 基本プロパティ (続き)

プロパティ名	説明	デフォルト値
defaultProperties	構成オプションのデフォルト・セットの名前。この値は、構成ディレクトリーにあるディレクトリーの名前です。このディレクトリーには複数のディレクトリーと、構成情報を指定するプロパティ・ファイルが含まれています。	デフォルトなし
enableFunctionalFixPack	<p>使用可能にするフィックスパック機能のレベル。デフォルトでは、フィックスパックに組み込まれている新しい機能は使用可能になっていません。このプロパティをバージョン ID に設定すると、そのバージョンで使用できる新しいフィーチャーを使用可能にすることができます。</p> <p>バージョン ID は、ピリオド文字 (.) を使用しても使用しなくても指定できます。例えば、IBM MQ 8.0.0 Fix Pack 2 で使用できる機能を使用するには、このプロパティを 8002 または 8.0.0.2 に設定してください。</p>	デフォルトなし

表 39. 基本プロパティ (続き)

プロパティ名	説明	デフォルト値
<p> V 9.0.2 V 9.0.0.2 V 9.0.2 V 9.0.0.2 messagePublicationFormat </p>	<p> MFT エージェントがその状況 XML メッセージ用に使用するメッセージ・パブリケーション形式を指定できます。このプロパティは、以下の値に設定できます。 </p> <p> messagePublicationFormat=mixed メッセージは MQMD FORMAT なしでパブリッシュされます (MQFMT_NONE)。ただし、/LOG トピック・ツリーの下で発行されるメッセージは例外で、MQFMT_STRING の MQMD 形式でパブリッシュされます。 </p> <p> messagePublicationFormat=MQFMT_NONE メッセージは MQMD FORMAT なしでパブリッシュされます。 </p> <p> messagePublicationFormat=MQFMT_STRING メッセージはストリング形式でパブリッシュされます。 </p> <p> V 9.0.2 IBM MQ 9.0.2 には、コーディング・エラーがあります。このバージョンでは、messagePublicationFormat=mixed を指定すると、その結果が messagePublicationFormat=MQFMT_NONE を指定した場合と同じになります。 V 9.0.3 このエラーは IBM MQ 9.0.3 で修正されています。 </p> <p> IBM MQ 8.0 より前のバージョンでは、MFT エージェントは XML 状況メッセージを SYSTEM.FTE トピックにストリング形式 (MQFMT_STRING) でパブリッシュしていました。可能な場合、以前に IBM WebSphere MQ 7.5 を使用していたアプリケーションを、IBM MQ 8.0 以降の形式でメッセージを処理するように更新してください。アプリケーションを変更できない場合は、messagePublicationFormat プロパティを MQFMT_STRING に設定して、IBM WebSphere MQ 7.5 の動作に戻します。 </p>	<p>messagePublicationFormat=mixed</p>

表 39. 基本プロパティ (続き)

プロパティ名	説明	デフォルト値
  z/OS 固有:		
    productId	<p>MFT 使用率の記録の対象となる製品タイプ。</p> <ul style="list-style-type: none"> • スタンドアロン Managed File Transfer 製品。(MFT が productId)。 • IBM MQ Advanced 製品の一部。(ADVANCED が productId)。 • IBM MQ Advanced for z/OS, Value Unit Edition 製品の一部。(ADVANCEDVUE が productId)。 <p>製品使用記録の詳細については、製品情報のレポートを参照してください。</p> <p> マルチプラットフォームでは、このプロパティは無視されます。</p>	MFT

次のテキストは、`installation.properties` ファイルの内容の例です。

```
defaultProperties=ERIS
```

ERIS は、`installation.properties` ファイルと同じディレクトリーにあるディレクトリーの名前です。ディレクトリー ERIS には複数のディレクトリーと、一連の構成オプションを記述するプロパティファイルが含まれています。

関連概念

57 ページの『[MFT 構成オプション \(Multiplatforms\)](#)』

Managed File Transfer は、セットアップに関する重要な情報が含まれた、操作に必要ないくつかのプロパティファイルを提供します。これらのプロパティファイルは、製品のインストール時に定義される構成ディレクトリーにあります。

関連資料

383 ページの『[fteChangeDefaultConfigurationOptions](#)』

fteChangeDefaultConfigurationOptions コマンドは、Managed File Transfer で使用するデフォルト構成オプションを変更するために使用します。構成オプションの値によって、Managed File Transfer で使用するプロパティファイルのグループを定義します。

MFT coordination.properties ファイル

`coordination.properties` ファイルは、調整キュー・マネージャーへの接続の詳細を指定します。複数の Managed File Transfer インストール済み環境が同じ調整キュー・マネージャーを共有する可能性があるため、共有ドライブ上の共通 `coordination.properties` ファイルへのシンボリック・リンクを使用できます。

`coordination.properties` ファイルは、インストーラーまたは **fteSetupCoordination** コマンドにより作成されます。**fteSetupCoordination** コマンドを、**-f** フラグと共に使用して、このファイルの基本調整キュー・マネージャー・プロパティを変更できます。拡張調整キュー・マネージャー・プロパティを変更または追加するには、テキスト・エディターでファイルを編集する必要があります。

coordination.properties ファイルは `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` ディレクトリーにあります。

coordination.properties ファイルには以下の値が含まれます。

プロパティー名	説明	デフォルト値
coordinationQMgr	調整キュー・マネージャーの名前。	デフォルトなし
coordinationQMgrHost	調整キュー・マネージャーのホスト名または IP アドレス。	デフォルトなし
coordinationQMgrPort	調整キュー・マネージャーとのクライアント接続で使用されるポート番号。	1414
coordinationQMgrChannel	調整キュー・マネージャーとの接続に使用される SVRCONN チャンネル名。	SYSTEM.DEF.SVRCONN

coordinationQMgrHost プロパティーの値を指定しない場合は、デフォルトでバインディング・モードが使用されます。

coordinationQMgrHost プロパティーに値を指定し、coordinationQMgrPort および coordinationQMgrChannel プロパティーには値を指定しない場合、ポート番号 1414 およびチャンネル SYSTEM.DEF.SVRCONN がデフォルトで使用されます。

プロパティー名	説明	デフォルト値
エージェントのプロパティー:		

表 41. 拡張調整キュー・マネージャーのプロパティ (続き)

プロパティ名	説明	デフォルト値
agentStatusJitterTolerance	<p>エージェントの状況メッセージのパブリッシュが遅れてもよい最大時間。その時間を超えると、メッセージは延滞と見なされます。この値は、ミリ秒単位で指定します。</p> <p>状況メッセージの経過時間は、調整キュー・マネージャーでパブリッシュされた時点が基点になります。ただし、メッセージがエージェントから送出されるのは、調整キュー・マネージャーでこのメッセージが受信される少し前です。これは、メッセージが IBM MQ ネットワークを移動するために必要な時間を許容するためです。この移動にかかる時間が常に同じであれば、実際の移動時間に関わらず、60 秒間隔で作成されるメッセージは、60 秒間隔でパブリッシュされます。ところが、その移動時間がメッセージごとに異なる場合は、60 秒間隔で作成されたメッセージが、例えば 61 秒、59 秒、58 秒、62 秒などの間隔でパブリッシュされる可能性があります。この場合に発生する 60 秒からの最大偏差のことをジッターといいます (この例では 2 秒です)。このプロパティでは、ジッターによる遅れの最大時間を指定します (その時間を超えると、メッセージは延滞と見なされます)。</p>	3000
コード・ページ・プロパティ:		
coordinationCcsid	<p>コマンドが、調整キュー・マネージャーに接続するとき使用するコード・ページ。エージェントによる調整キュー・マネージャーへのすべてのパブリケーションも、このコード・ページを使用して実行されます。coordinationCcsid の値を指定する場合は、coordinationCcsidName の値も指定する必要があります。</p>	1208

表 41. 拡張調整キュー・マネージャーのプロパティ (続き)		
プロパティ名	説明	デフォルト値
coordinationCcsidName	coordinationCcsid の Java 表記。coordinationCcsidName の値を指定する場合は、coordinationCcsid の値も指定する必要があります。	UTF8
接続プロパティ:		
javaLibraryPath	Managed File Transfer では、バインディング・モードでキュー・マネージャーに接続するときに、IBM MQ Java バインディング・ライブラリーへのアクセス権限が必要になります。Managed File Transfer は、デフォルトでは、IBM MQ で定義されているデフォルトの場所でバインディング・ライブラリーを検索します。バインディング・ライブラリーが別の場所にある場合は、このプロパティを使用して、バインディング・ライブラリーの場所を指定してください。	MQ_INSTALLATION_PATH/ java/lib
マルチ・インスタンス・キュー・マネージャーのプロパティ:		
coordinationQMGrStandby	coordinationQMGr プロパティによって定義されている、複数インスタンスの調整キュー・マネージャーのスタンバイ・インスタンスに、クライアント接続するために使用する ホスト名およびポート番号 (IBM MQ の CONNAME 形式で指定)。例えば、 <i>host_name(port_number)</i> のように指定します。	デフォルトなし
キュー・プロパティ:		

表 41. 拡張調整キュー・マネージャーのプロパティ (続き)

プロパティ名	説明	デフォルト値
dynamicQueuePrefix	<p>このプロパティは、一時キュー名を生成するために使用する IBM MQ 接頭辞を定義します。</p> <p>dynamicQueuePrefix プロパティの形式は、IBM MQ MQOD 構造の DynamicQName フィールドの形式に従います。詳細については、『動的キューの作成』を参照してください。</p> <p>エージェントからの応答を必要とするコマンドにより生成される一時応答キューに、特定の IBM MQ 接頭辞を使用する場合は、command.properties ファイルでこのプロパティを定義することもできます。</p>	WMQFTE.*
modelQueueName	<p>このプロパティは、一時キューを生成するために使用する IBM MQ MQ モデル・キューを定義します。</p> <p>エージェントからの応答を必要とするコマンドにより生成される一時応答キューに、特定の IBM MQ モデル・キューを使用する場合は、command.properties ファイルでこのプロパティを定義することもできます。詳しくは、546 ページの『MFT command.properties ファイル』を参照してください。</p>	SYSTEM.DEFAULT.MODEL.QUEUE
セキュリティ・プロパティ:		
userIdForClientConnect	<p>IBM MQ へのクライアント接続を介してフローされるユーザー ID。java が指定されている場合、JVM によって報告されるユーザー名が、IBM MQ 接続要求の一部として送られます。このプロパティの値は None または java のいずれかになります。</p>	なし

表 41. 拡張調整キュー・マネージャーのプロパティ (続き)		
プロパティ名	説明	デフォルト値
coordinationQMGrAuthenticationCredentialsFile	調整キュー・マネージャーに接続するための MQ 接続資格情報が入ったファイルへのパス。	<p>z/OS 認証資格情報ファイルの作成に関する詳細は、MFT 資格情報ファイルの作成を参照してください</p> <p>ULW このファイルの場所およびアクセス権については、MQMFTCredentials.xml の構成を参照してください。</p> <p>ULW 認証資格情報ファイルの作成について詳しくは、MFT と IBM MQ の接続認証を参照してください。</p>
サブスクリプション・プロパティ:		
coordinationSubscriptionTopic	<p>IBM MQ ネットワークの状況に関するパブリケーションを取得するために、サブスクライブ先の SYSTEM.FTE 以外のトピックを指定する場合に、このプロパティを使用してください。すべてのツールは引き続き SYSTEM.FTE トピックにパブリッシュしますが、IBM MQ トポロジーを変更することにより、それらのパブリケーションを内容に応じて異なるトピックに配布することができます。それから、この機能を使用して、他のトピックのいずれかにサブスクライブするよう、ツールを強制することができます。</p> <p>IBM WebSphere MQ 7.5 およびそれ以降のフィックスパックの場合、プロパティが IBM MQ Explorer および fteListMonitors コマンドによって認識されるためには、APAR IC96850 の暫定修正が必要です。</p>	SYSTEM.FTE

次のテキストは、coordination.properties ファイルの内容の例です。

```
coordinationQMGr=ERIS
coordinationQMGrHost=kuiper.example.com
coordinationQMGrPort=2005
coordinationQMGrChannel=SYSTEM.DEF.SVRCONN
```

ERIS は、システム kuiper.example.com 上にある IBM MQ キュー・マネージャーの名前です。キュー・マネージャー ERIS は、Managed File Transfer のログ情報の送信先であるキュー・マネージャーです。

関連概念

57 ページの『[MFT 構成オプション \(Multiplatforms\)](#)』

Managed File Transfer は、セットアップに関する重要な情報が含まれた、操作に必要ないくつかのプロパティ・ファイルを提供します。これらのプロパティ・ファイルは、製品のインストール時に定義される構成ディレクトリーにあります。

関連資料

514 ページの『[fteSetupCoordination](#)』

fteSetupCoordination コマンドは、Managed File Transfer のプロパティ・ファイルと調整キュー・マネージャー・ディレクトリーを作成します。

605 ページの『[MFT の SSL プロパティ](#)』

SSL または TLS を IBM MQ および Managed File Transfer とともに使用して、エージェントとキュー・マネージャーとの間の許可されない接続を防止し、エージェントとキュー・マネージャーとの間のメッセージ・トラフィックを暗号化します。

MFT command.properties ファイル

command.properties ファイルは、コマンド発行時に接続先となるコマンド・キュー・マネージャーと、そのキュー・マネージャーに接続するために Managed File Transfer が必要とする情報を指定します。

command.properties ファイルは、インストーラーまたは **fteSetupCommands** コマンドにより作成されます。**fteSetupCommands** コマンドを **-f** フラグと共に使用して、このファイルで基本コマンド・キュー・マネージャー・プロパティを変更できます。拡張コマンド・キュー・マネージャー・プロパティを変更または追加するには、テキスト・エディターでファイルを編集する必要があります。

一部の Managed File Transfer のコマンドは、コマンド・キュー・マネージャーの代わりにエージェント・キュー・マネージャーまたは調整キュー・マネージャーに接続します。どのコマンドがどのキュー・マネージャーに接続するかについては、[367 ページの『MFT のコマンドおよびプロセスと、その接続先のキュー・マネージャー』](#)を参照してください。

command.properties ファイルは `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` ディレクトリーにあります。

command.properties ファイルには以下の値が含まれます。

プロパティ名	説明	デフォルト値
connectionQMgr	IBM MQ ネットワークとの接続に使用されるキュー・マネージャーの名前。	デフォルトなし
connectionQMgrHost	接続キュー・マネージャーのホスト名または IP アドレス。	デフォルトなし
connectionQMgrPort	クライアント・モードの接続キュー・マネージャーとの接続に使用されるポート番号。	1414
connectionQMgrChannel	接続キュー・マネージャーとの接続に使用される SVRCONN チャンネル名。	SYSTEM.DEF.SVRCONN

connectionQMgrHost プロパティの値を指定しない場合は、デフォルトでバインディング・モードが使用されます。

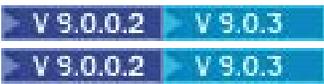
connectionQMgrHost プロパティに値を指定し、connectionQMgrPort および connectionQMgrChannel プロパティには値を指定しない場合、ポート番号 1414 およびチャンネル SYSTEM.DEF.SVRCONN がデフォルトで使用されます。

表 43. 拡張コマンド・キュー・マネージャーのプロパティ		
プロパティ名	説明	デフォルト値
コード・ページ・プロパティ:		
connectionCcsid	コマンドが、コマンド・キュー・マネージャーに接続するときに使用するコード・ページ。 connectionCcsid の値を指定する場合は、connectionCcsidName の値も指定する必要があります。	1208
connectionCcsidName	connectionCcsid の Java 表記。 connectionCcsidName の値を指定する場合は、connectionCcsid の値も指定する必要があります。	UTF8
マルチ・インスタンス・キュー・マネージャーのプロパティ:		
connectionQMGrStandby	connectionQMGr プロパティによって定義されている、複数インスタンスのコマンド・キュー・マネージャーのスタンバイ・インスタンスに、クライアント接続するために使用するホスト名およびポート番号 (IBM MQ の CONNAME 形式で指定)。例えば、 <i>host_name(port_number)</i> のように指定します。	デフォルトなし
セキュリティ・プロパティ:		
userIdForClientConnect	IBM MQ へのクライアント接続を介してフローされるユーザー ID。 java が指定されている場合、JVM によって報告されるユーザー名が、IBM MQ 接続要求の一部として送られます。このプロパティの値は None または java のいずれかになります。	なし
connectionQMGrAuthenticationCredentialsFile	コマンド・キュー・マネージャーに接続するための MQ 接続資格情報が入ったファイルへのパス。	<p>Z/OS 認証資格情報ファイルの作成に関する詳細は、MFT 資格情報ファイルの作成を参照してください</p> <p>ULW このファイルの場所およびアクセス権については、MQMFTCredentials.xml の構成を参照してください。</p> <p>ULW 認証資格情報ファイルの作成について詳しくは、MFT と IBM MQ の接続認証を参照してください。</p>
キュー・プロパティ:		

表 43. 拡張コマンド・キュー・マネージャーのプロパティ (続き)

プロパティ名	説明	デフォルト値
dynamicQueuePrefix	<p>エージェントからの応答を必要とするコマンドの場合、このプロパティは、一時応答キュー名生成のために使用する IBM MQ 接頭部を定義します。</p> <p>dynamicQueuePrefix プロパティの形式は、IBM MQ MQOD 構造の DynamicQName フィールドの形式に従います。詳細については、『動的キューの作成』を参照してください。</p> <p>WMQFTE により生成される一時キューに特定の IBM MQ 接頭辞を使用する場合は、<code>coordination.properties</code> ファイルでこのプロパティを定義することもできます。</p>	WMQFTE.*
modelQueueName	<p>エージェントからの応答を必要とするコマンドの場合、このプロパティは、一時応答キュー生成のために使用する IBM MQ モデル・キューを定義します。</p> <p>WMQFTE により生成される一時キューに特定の IBM MQ モデル・キューを使用する場合は、<code>coordination.properties</code> ファイルでこのプロパティを定義することもできます。詳しくは、540 ページの『MFT <code>coordination.properties</code> ファイル』を参照してください。</p>	SYSTEM.DEFAULT.MODEL.QUEUE
接続プロパティ:		
javaLibraryPath	<p>Managed File Transfer では、バインディング・モードでキュー・マネージャーに接続するときに、IBM MQ Java バインディング・ライブラリーへのアクセス権限が必要になります。Managed File Transfer は、デフォルトでは、IBM MQ で定義されているデフォルトの場所でバインディング・ライブラリーを検索します。バインディング・ライブラリーが別の場所にある場合は、このプロパティを使用して、バインディング・ライブラリーの場所を指定してください。</p>	/opt/mqm/java/lib

表 43. 拡張コマンド・キュー・マネージャーのプロパティ (続き)

プロパティ名	説明	デフォルト値
 legacyXMLMessageMQMDFormat	<p>Managed File Transfer コマンドの XML メッセージは、ブランクの MQMD 形式フィールドと共に、キューに送信されるようになりました。以前のバージョンの製品では、MQMD 形式フィールドを MQSTR (テキスト・メッセージ・ストリング) に設定していました。このプロパティを true に設定すると、Managed File Transfer コマンドの XML メッセージを、MQSTR の MQMD 形式フィールドと共にキューに送信することが可能になります。</p> <p>MQMD 形式フィールドが MQSTR に設定されている場合、データ変換が使用可能になっている MQ ネットワーク内のチャンネルが存在すると、Managed File Transfer コマンドの XML メッセージが壊れる可能性があります。</p>	false
エージェントのプロパティ		
 failCleanAgentWithNoArguments	<p>デフォルトでは、このプロパティの値は true に設定されているため、エージェント名のパラメーターのみが指定されている場合、fteCleanAgent コマンドの実行が失敗します。このプロパティを false に設定すると、エージェント名のパラメーターのみが設定されている場合に、fteCleanAgent コマンドの動作は -all パラメーターを指定したときの動作と同じになります。</p>	true

次のテキストは、command.properties ファイルの内容の例です。

```
connectionQMGr=PLUTO
connectionQMGrHost=kuiper.example.com
connectionQMGrPort=1930
connectionQMGrChannel=SYSTEM.DEF.SVRCONN
```

PLUTO は、システム kuiper.example.com 上にある IBM MQ キュー・マネージャーの名前です。キュー・マネージャー PLUTO は、Managed File Transfer のコマンドの接続先のキュー・マネージャーです。

関連概念

57 ページの『MFT 構成オプション (Multiplatforms)』

Managed File Transfer は、セットアップに関する重要な情報が含まれた、操作に必要ないくつかのプロパティ・ファイルを提供します。これらのプロパティ・ファイルは、製品のインストール時に定義される構成ディレクトリにあります。

関連資料

605 ページの『MFT Java システム・プロパティ』

いくつかの Managed File Transfer コマンド・プロパティとエージェント・プロパティは Java システム・プロパティとして定義する必要があります。これは、コマンド・プロパティおよびエージェント・プロパティのメカニズムを使用できない初期の関数用の構成を定義するためです。

605 ページの『MFT の SSL プロパティ』

SSL または TLS を IBM MQ および Managed File Transfer とともに使用して、エージェントとキュー・マネージャーとの間の許可されない接続を防止し、エージェントとキュー・マネージャーとの間のメッセージ・トラフィックを暗号化します。

512 ページの『fteSetupCommands: MFT command.properties ファイルの作成』

fteSetupCommands コマンドは、Managed File Transfer `command.properties` ファイルを作成します。このプロパティ・ファイルは、コマンドの発行時に IBM MQ ネットワークに接続するキュー・マネージャーの詳細を指定します。

384 ページの『fteCleanAgent: MFT エージェントのクリーンアップ』

fteCleanAgent コマンドを使用して、Managed File Transfer Agent が使用する永続的および非永続的キューからメッセージを削除することにより、エージェントが使用するキューをクリーンアップします。エージェントの開始に問題があり、エージェントが使用するキューに情報が残っていることが原因である可能性がある場合、**fteCleanAgent** コマンドを使用します。

MFT agent.properties ファイル

各 Managed File Transfer Agent には、独自のプロパティ・ファイル `agent.properties` があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。`agent.properties` ファイルには、エージェントの動作を変更するプロパティを含めることもできます。

`agent.properties` ファイルは、インストーラーまたは **fteCreateAgent**、**fteCreateBridgeAgent**、または **fteCreateCDAgent** コマンドによって作成されます。これらのコマンドのいずれかを **-f** フラグと共に使用して、基本エージェント・キュー・マネージャー・プロパティおよび、作成しているエージェントのタイプに関連付けられている拡張エージェント・プロパティを変更できます。拡張エージェント・プロパティを変更または追加するには、テキスト・エディターでファイルを編集する必要があります。

エージェントの `agent.properties` ファイルは、`MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` ディレクトリーにあります。

z/OS z/OS では、`agent.properties` ファイルの場所は `$BFG_CONFIG variable/mqft/config/coordination_qmgr_name/agents/agent_name` です。

`agent.properties` ファイルを変更した場合は、エージェントを再始動してその変更が反映されるようにする必要があります。

IBM WebSphere MQ 7.5 以降の場合は、ファイルまたはディレクトリーの場所を表す Managed File Transfer プロパティの一部で環境変数を使用できます。これにより、製品の一部の実行時に使用されるファイルまたはディレクトリーの場所を、環境の変更 (プロセスを実行しているユーザーなど) に合わせて変えることができます。詳しくは、535 ページの『MFT プロパティでの環境変数の使用』を参照してください。

エージェントの基本プロパティ

各 `agent.properties` ファイルには、以下の基本プロパティが含まれています。

プロパティ名	説明	デフォルト値
agentName	エージェントの名前。エージェントの名前は IBM MQ オブジェクト命名規則に準拠していなければなりません。詳しくは、623 ページの『MFT オブジェクト命名規則』を参照してください。	デフォルトなし

表 44. エージェントの基本プロパティ (続き)		
プロパティ名	説明	デフォルト値
agentDesc	エージェントの説明 (説明の作成を選択した場合)。	デフォルトなし
agentQMgr	エージェントのキュー・マネージャーの名前。	デフォルトなし
agentQMgrHost	エージェントのキュー・マネージャーのホスト名または IP アドレス。	デフォルトなし
agentQMgrPort	エージェントのキュー・マネージャーとのクライアント接続で使用されるポート番号。	1414
agentQMgrChannel	エージェントのキュー・マネージャーとの接続に使用される SVRCONN チャンネル名。	SYSTEM.DEF.SVRCONN
agentType	エージェントのタイプ。 ・ 標準非ブリッジ・エージェント (STANDARD) ・ プロトコル・ブリッジ・エージェント (BRIDGE) ・ Connect:Direct ブリッジ・エージェント (CD_BRIDGE) ・ IBM Integration Bus によって使用される組み込みエージェント (EMBEDDED) ・ Sterling File Gateway 組み込みエージェント (SFG)	STANDARD

agentQMgrHost プロパティの値を指定しない場合は、デフォルトでバインディング・モードが使用されます。

agentQMgrHost プロパティに値を指定し、agentQMgrPort および agentQMgrChannel プロパティには値を指定しない場合、ポート番号 1414 およびチャンネル SYSTEM.DEF.SVRCONN がデフォルトで使用されます。

拡張エージェント・プロパティ

Managed File Transfer は、エージェントの構成に役立つ拡張エージェント・プロパティも備えています。以下のプロパティのいずれかを使用する場合、手動で agent.properties ファイルを編集し、必要な拡張プロパティを追加します。Windows のファイル・パスを指定する場合は、区切り文字のバックスラッシュ (¥) が 2 つのバックスラッシュ (¥¥) で入力されていること、つまり、バックスラッシュ (¥) がエスケープされていることを確認してください。あるいは、単一のスラッシュ文字 (/) を区切り文字として使用することもできます。Java プロパティ・ファイルでの文字のエスケープ・シーケンスについては、Oracle の資料 [Javadoc for the Properties class](#) を参照してください。

- ・ [エージェント・サイズ・プロパティ](#)
- ・ [コード・ページ・プロパティ](#)
- ・ [コマンド・プロパティ](#)
- ・ [接続プロパティ](#)
- ・ [Connect:Direct ブリッジ・プロパティ](#)
- ・ [ファイルからメッセージへの転送とメッセージからファイルへの転送に関するエージェント・プロパティ](#)
- ・ [汎用エージェント・プロパティ](#)
- ・ [入出力プロパティ](#)
- ・ [マルチチャンネル・サポート・プロパティ](#)
- ・ [マルチ・インスタンス・プロパティ](#)
- ・ [プロセス・コントローラーのプロパティ](#)

- [プロトコル・ブリッジ・プロパティ](#)
- **V 9.0.0** [プロトコル・ブリッジ・エージェントのログ・プロパティ](#)
- [キュー・プロパティ](#)
- [リソース・モニター・プロパティ](#)
- [ルート・ディレクトリー・プロパティ](#)
- [スケジューラー・プロパティ](#)
- [セキュリティー・プロパティ](#)
- [タイムアウト・プロパティ](#)
- **V 9.0.1** [転送リカバリー・タイムアウト・プロパティ](#)
- [トレースとロギングのプロパティ](#)
- [転送制限プロパティ](#)
- [ユーザー出口ルーチン・プロパティ](#)
- [IBM MQ クライアント圧縮プロパティ](#)
- **z/OS** [z/OS 固有のプロパティ](#)
- [その他のプロパティ](#)

表 45. 拡張エージェント・プロパティ		
プロパティ名	説明	デフォルト値
エージェント・サイズ・プロパティ:		
agentCheckpointInterval	<p>チェックポイントがリカバリー目的で設定される完全なデータ・フレーム単位での間隔。これは拡張プロパティであり、ほとんどの Managed File Transfer 構成では、このプロパティの値を変更する必要はありません。</p> <p>何らかの問題によって転送がリカバリー状態になる場合、転送をリカバリーできるのは、チェックポイントの境界までだけです。したがって、この値を大きくすると (agentChunkSize、agentWindowSize、および agentFrameSize にも大きな値を設定)、エージェントが転送をリカバリーするのに必要な時間が長くなります。転送がリカバリー状態になることがほとんどない信頼性のある Managed File Transfer ネットワークでは、この値を増やすと、全体的なパフォーマンスの改善に役立つ可能性があります。</p>	1
agentChunkSize	<p>ファイル・データのトランスポートの各転送チャンクのサイズしたがって、ソース・エージェントと宛先エージェントとの間で転送される IBM MQ メッセージの最大サイズを意味します。これは拡張プロパティであり、ほとんどの Managed File Transfer 構成では、このプロパティの値を変更する必要はありません。</p> <p>この値は、ソース・エージェントと宛先エージェントの間でネゴシエーションされ、両者の値のうち大きい方が使用されます。このプロパティの値を変更する場合は、ソース・エージェント側と宛先エージェント側の両方で値を変更してください。</p> <p>agentChunkSize は整数値です。例えば、agentChunkSize=10240 とすると、チャンク・サイズは 10 KB に設定されます。</p>	262144 バイト (256 KB 相当)
agentFrameSize	<p>転送フレームのウィンドウの数。これは拡張プロパティであり、ほとんどの Managed File Transfer 構成では、このプロパティの値を変更する必要はありません。</p> <p>待ち時間の長いネットワークでは、この値を増やすとエージェントが同時にアクティブにできるメッセージ・チャンクが増えるため、全体的なパフォーマンスの改善に役立つ可能性があります。</p> <p>このプロパティの値に agentWindowSize を乗算し、さらに agentChunkSize を乗算した値は、転送ごとのエージェントのメモリー消費量の上限を意味します。例えば、262144 バイトのチャンク $\times 10 \times 5 = 12.5$ MB (転送ごと) となります。</p> <p>注: 単一の転送で転送されるファイルのサイズが 12.5 MB より小さい場合、このプロパティの値を増やしても、転送のパフォーマンスに与える効果はありません。</p>	5

表 45. 拡張エージェント・プロパティ (続き)

プロパティ名	説明	デフォルト値
agentWindowSize	<p>各ウィンドウのチャンクの数。これは拡張プロパティであり、ほとんどの Managed File Transfer 構成では、このプロパティの値を変更する必要はありません。</p> <p>待ち時間の長いネットワークでは、この値を増やすと、全体的なパフォーマンスの改善に役立つ可能性があります。その理由は、エージェントが同時にアクティブにできるメッセージ・チャンクが増えて、確認応答メッセージがソース・エージェントに返信される頻度が減るためです。</p> <p>このプロパティの値に agentFrameSize を乗算し、さらに agentChunkSize を乗算した値は、転送ごとのエージェントのメモリー消費量の上限、および宛先エージェントのコマンド・キューでの IBM MQ メッセージ・データの上限を意味します。例えば、262144 バイトのチャンク $\times 10 \times 5 =$ 上限 12.5 MB (転送ごと) となります。</p> <p>注: 単一の転送で転送されるファイルのサイズが 12.5 MB より小さい場合、このプロパティの値を増やしても、転送のパフォーマンスに与える効果はありません。</p>	10
コード・ページ・プロパティ:		
agentCcsid	<p>エージェントがそのエージェント・キュー・マネージャーに接続するときに使用するコード・ページ。agentCcsid の値を指定する場合は、agentCcsidName の値も指定する必要があります。JVM にとって既知のコード・ページを表示する方法については、fteCreateBridgeAgent コマンドの <code>-hsc</code> パラメーターを参照してください。</p>	1208
agentCcsidName	<p>agentCcsid の Java 表記。agentCcsidName の値を指定する場合は、agentCcsid の値も指定する必要があります。</p>	UTF8
コマンド・プロパティ:		
maxCommandHandlerThreads	<p>初期構文解析および転送コマンド・メッセージの処理に使用可能なスレッド数を制御します。アクティブである場合には、スレッドはキュー・マネージャーへの接続を必要としますが、アイドルである場合には、スレッドは接続を解放します。</p>	5
maxCommandOutput	<p>コマンド出力用に保管される最大バイト数。このプロパティは、管理対象呼び出し用に指定されたコマンド、および、管理対象転送用の preSource、postSource、preDestination、および postDestination コマンドに適用されます。この値によって、SYSTEM.FTE トピックの転送ログに書き込まれるコマンド出力の長さが制限されます。</p>	10240
maxCommandRetries	<p>エージェントが許可するコマンド再試行の最大回数。このプロパティは、管理対象呼び出し用に指定されたコマンド、および、管理対象転送用の preSource、postSource、preDestination、および postDestination コマンドに適用されます。</p>	9
maxCommandWait	<p>エージェントが許可する各再試行間の最大待ち時間 (秒単位)。このプロパティは、管理対象呼び出し用に指定されたコマンド、および、管理対象転送用の preSource、postSource、preDestination、および postDestination コマンドに適用されます。</p>	60
immediateShutdownTimeout	<p>エージェントの即時シャットダウンの場合は、このプロパティを使用して、シャットダウンを強制する前にエージェントがその転送の完了を待つ最大時間を秒単位で指定することができます。</p> <p>注: このプロパティの値をデフォルト (10 秒) 未満に変更してはなりません。エージェントの即時シャットダウンには、外部プロセスをすべて終了するための十分な時間が必要です。このプロパティの値が低すぎる場合は、プロセスが実行されたままになる可能性があります。</p> <p>このプロパティに 0 の値が指定されると、エージェントは、未解決の転送がすべて停止するのを待ちます。このプロパティに無効値が指定されると、デフォルト値が使用されます。</p>	10
接続プロパティ:		
javaLibraryPath	<p>Managed File Transfer では、バインディング・モードでキュー・マネージャーに接続するときに、IBM MQ Java バインディング・ライブラリーへのアクセス権限が必要になります。Managed File Transfer は、デフォルトでは、IBM MQ で定義されているデフォルトの場所でバインディング・ライブラリーを検索します。バインディング・ライブラリーが別の場所にある場合は、このプロパティを使用して、バインディング・ライブラリーの場所を指定してください。</p>	なし
Connect:Direct ブリッジ・プロパティ		

表 45. 拡張エージェント・プロパティ (続き)		
プロパティ名	説明	デフォルト値
cdNode	Connect:Direct ブリッジを使用する場合に必要なプロパティ。 Connect:Direct ブリッジ・エージェントから宛先の Connect:Direct ノードにメッセージを転送するために使用する Connect:Direct ノードの名前。このノードは、Connect:Direct ブリッジの一部であって、転送のソースまたは宛先であるリモート・ノードではありません。詳しくは、230 ページの『Connect:Direct ブリッジ』を参照してください。	デフォルトなし
cdNodeHost	Connect:Direct ブリッジ・エージェントから宛先ノードにファイルを転送するときに使用する Connect:Direct ノード (Connect:Direct ブリッジ・ノード) のホスト名または IP アドレス。 ほとんどの場合、Connect:Direct ブリッジ・ノードは、Connect:Direct ブリッジ・エージェントと同じシステム上にあります。その場合、このプロパティのデフォルト値 (ローカル・システムの IP アドレス) は適正です。システムに複数の IP アドレスがある場合、または Connect:Direct ブリッジ・ノードが Connect:Direct ブリッジ・エージェントと異なるシステム上にあり、両者のシステムが同一のファイル・システムを共有する場合は、このプロパティを使用して Connect:Direct ブリッジ・ノード用の適切なホスト名を指定します。 cdNode プロパティを設定しなかった場合、このプロパティは無視されます。	ローカル・システムのホスト名または IP アドレス
cdNodePort	クライアント・アプリケーションがノードとの通信に使用する Connect:Direct ブリッジ・ノードのポート番号。Connect:Direct の製品資料で、このポートは API ポートと呼ばれています。 cdNode プロパティを設定しなかった場合、このプロパティは無視されます。	1363
cdTmpDir	ファイルが宛先の Connect:Direct ノードに転送される前に、一時的に保管される、Connect:Direct ブリッジ・エージェントが実行されているシステム上の場所。 このプロパティは、ファイルが一時的に保管されるディレクトリーの絶対パスを指定します。例えば、cdTmpDir を /tmp に設定すると、ファイルは /tmp ディレクトリーに一時的に置かれます。 cdNode プロパティを設定しなかった場合、このプロパティは無視されます。 このプロパティの値には、環境変数を含めることができます。 詳細については、535 ページの『MFT プロパティでの環境変数の使用』を参照してください。	<code>value_of_java.io.tmpdir/cdbridge-agentName</code> Windows の場合: <code>value_of_java.io.tmpdir\cdbridge-agentName</code>
cdTrace	Connect:Direct ブリッジ・エージェントと Connect:Direct ノード間で送信されるデータをエージェントがトレースするかどうかを指定します。このプロパティの値は、true または false のいずれかになります。	false
cdMaxConnectionRetries	正常な接続がまだ確立されていない場合のファイル転送で、Connect:Direct 接続を試行できる最大回数。この回数を超えると転送は失敗します。	-1 (試行回数は無制限)
cdMaxPartialWorkConnectionRetries	前回の接続試行が正常に終了し、転送作業が完了した場合に、ファイル転送で Connect:Direct 接続を試行できる最大回数。この回数を超えると転送は失敗します。	-1 (試行回数は無制限)
cdMaxWaitForProcessEndStats	プロセス終了後、Connect:Direct ノードの統計情報内で Connect:Direct プロセス完了情報が使用できるようになるまでの最大待機時間 (ミリ秒)。この時間を超えるとファイル転送が失敗と判定されます。一般に、この情報は即時に使用可能になりますが、特定の失敗条件下においては情報がパブリッシュされません。そうした条件において、このプロパティで指定した待機時間後にファイル転送は失敗します。	60000
cdAppName	Connect:Direct ブリッジ・エージェントがブリッジの一部である Connect:Direct ノードに接続するときに使用するアプリケーション名。	Managed File Transfer <i>current version</i> (<i>current version</i> は、製品のバージョン番号です)。
cdNodeLocalPortRange	Connect:Direct ブリッジ・エージェントと、ブリッジの一部である Connect:Direct ノードとの間のソケット接続で使用するローカル・ポートの範囲。この値の形式は、値または範囲のコンマ区切りリストです。 デフォルトでは、オペレーティング・システムによってローカル・ポート番号が選択されます。	なし
cdNodeProtocol	Connect:Direct ブリッジ・エージェントがブリッジの一部である Connect:Direct ノードに接続するときに使用するプロトコル。有効な値は、以下のとおりです。 <ul style="list-style-type: none">TCP/IPSSLTLS	TCP/IP

表 45. 拡張エージェント・プロパティ (続き)		
プロパティ名	説明	デフォルト値
cdNodeKeystore	Connect:Direct ブリッジ・エージェントと、ブリッジの一部である Connect:Direct ノードとの間のセキュア通信で使用する鍵ストアのパス。 SSL または TLS に cdNodeProtocol プロパティを設定しなかった場合、このプロパティは無視されます。 このプロパティの値には、環境変数を含めることができます。 詳細については、535 ページの『MFT プロパティでの環境変数の使用』を参照してください。	なし
cdNodeKeystoreType	cdNodeKeystore プロパティで指定する鍵ストアのファイル・フォーマット。有効な値は、jks と pkcs12 です。 SSL または TLS に cdNodeProtocol プロパティを設定しなかった場合、このプロパティは無視されます。	jks
cdNodeKeystoreCredentialsFile	cdNodeKeystore 資格情報が含まれているファイルのパス。 このプロパティの値には、環境変数を含めることができます。 詳細については、535 ページの『MFT プロパティでの環境変数の使用』を参照してください。	 認証資格情報ファイルの作成に関する詳細は、MFT 資格情報ファイルの作成を参照してください  このファイルの場合およびアクセス権については、MQMFTCredentials.xml の構成を参照してください。  認証資格情報ファイルの作成について詳しくは、MFT と IBM MQ の接続認証を参照してください。
cdNodeTruststore	Connect:Direct ブリッジ・エージェントと、ブリッジの一部である Connect:Direct ノードとの間のセキュア通信で使用するトラストストアのパス。 SSL または TLS に cdNodeProtocol プロパティを設定しなかった場合、このプロパティは無視されます。 このプロパティの値には、環境変数を含めることができます。 詳細については、535 ページの『MFT プロパティでの環境変数の使用』を参照してください。	なし
cdNodeTruststoreType	cdNodeTruststore プロパティで指定するトラストストアのファイル・フォーマット。有効な値は、jks と pkcs12 です。 SSL または TLS に cdNodeProtocol プロパティを設定しなかった場合、このプロパティは無視されます。	jks
cdNodeTruststoreCredentialsFile	cdNodeTruststore 資格情報が含まれているファイルのパス。 このプロパティの値には、環境変数を含めることができます。 詳細については、535 ページの『MFT プロパティでの環境変数の使用』を参照してください。	MFT 資格情報ファイルの作成および MQMFTCredentials.xml の構成を参照してください  認証資格情報ファイルの作成の詳細については、MFT 認証資格情報の作成もご覧ください。  認証資格情報ファイルの作成について詳しくは、MFT と IBM MQ の接続認証を参照してください。
logCDProcess	output0.log ファイルでエージェントのイベント・ログに記録する Connect:Direct プロセス・ロギングのレベル。このプロパティがとることのできる値は None、Failures、または All です。	なし
ファイルからメッセージへの転送とメッセージからファイルへの転送に関するエージェント・プロパティ:		
deleteTmpFileAfterRenameFailure	このプロパティを値 false に設定すると、名前変更操作が失敗した場合に一時ファイルが宛先から削除されません。この場合、転送されたデータは、宛先の一時ファイル (.part) に残ります。このファイルは、後から手動で名前変更できます。デフォルトでは、このプロパティの値は true です。このプロパティは、メッセージからファイルへの転送とファイルからファイルへの転送の両方に適用されます。	true

表 45. 拡張エージェント・プロパティ (続き)		
プロパティ名	説明	デフォルト値
enableQueueInputOutput	デフォルトでは、エージェントが転送の一環としてソース・キューからデータを読み取ったり、宛先キューにデータを書き込んだりすることはできません。この値を true に設定すると、エージェントは、ファイルからメッセージへの転送とメッセージからファイルへの転送を実行できるようになります。このプロパティの値は、true または false のいずれかになります。	false
enableSystemQueueInputOutput	エージェントが IBM MQ のシステム・キューとの間で読み取り/書き込みを実行できるかどうかを指定します。システム・キューには、SYSTEM 修飾子が接頭部として付いています。 注: システム・キューは、IBM MQ、Managed File Transfer、さらには重要な情報を送信する他のアプリケーションが使用するキューです。このプロパティを変更すると、エージェントがそれらのキューにアクセスできるようになります。このプロパティを有効にする場合は、ユーザー・サンドボックス機能を使用して、エージェントがアクセスできるキューを制限してください。	false
enableClusterQueueInputOutput	エージェントが IBM MQ クラスター・キューとの間で読み取り/書き込みを実行できるかどうかを指定します。 注: enableQueueInputOutput プロパティへの追加として、enableClusterQueueInputOutput エージェント・プロパティの指定が必要です。	false
maxDelimiterMatchLength	ファイルからメッセージへの転送の一環として 1 つのテキスト・ファイルを複数のメッセージに分割するときに使用する Java 正規表現のマッチング項目と見なせる文字の最大数。	5
maxInputOutputMessageLength	エージェントによってソース・キューから読み取られるメッセージ、または宛先キューに書き込まれるメッセージの最大長 (バイト単位)。転送のソース・エージェントの maxInputOutputMessageLength プロパティは、ソース・キューのメッセージから読み取ることができるバイト数を決定します。転送の宛先エージェントの maxInputOutputMessageLength プロパティは、宛先キューのメッセージに書き込むバイト数を決定します。メッセージの長さがこのプロパティの値を超えた場合、転送はエラーを出して失敗します。このプロパティは、Managed File Transfer の内部キューに影響を及ぼしません。このプロパティの変更について詳しくは、322 ページの『メッセージ・サイズに関連する MQ 属性および MFT プロパティを設定する際のガイダンス』を参照してください。	1048576
monitorGroupRetryLimit	メッセージ・グループがまだキューに存在している場合に、モニターがメッセージからファイルへの転送を再度起動する最大回数。メッセージからファイルへの転送が起動された回数は、グループの最初のメッセージの MQMD バックアウト・カウントで判別されます。 エージェントが再始動した場合、転送の起動回数が monitorGroupRetryLimit の値を超えている場合でも、モニターは再度転送を起動します。この動作により、転送の起動回数が monitorGroupRetryLimit の値を超えてしまった場合、エージェントはイベント・ログにエラーを書き込みます。 このプロパティに値 -1 が指定された場合、モニターは、起動条件が満たされなくなるまで何度でも転送を再度起動します。	10
汎用エージェント・プロパティ		
agentStatusPublishRateLimit	ファイル転送状況に変更があった場合に、エージェントがその状況をリパブリッシュする最大速度 (秒)。 このプロパティに小さすぎる値を設定すると、IBM MQ ネットワークのパフォーマンスにマイナスの影響を与える可能性があります。	30
agentStatusPublishRateMin	エージェントが自身の状況をパブリッシュする最小速度 (秒単位)。この値は、agentStatusPublishRateLimit プロパティの値以上でなければなりません。	300
enableMemoryAllocationChecking	転送を受け入れる前に転送の実行に使用可能なメモリーが十分あることを Managed File Transfer Agent が検査するかどうかを決定します。この検査は、ソース・エージェントと宛先エージェントの両方で行われます。使用可能なメモリーが不足している場合は、転送はリジェクトされます。 転送に必要なメモリーの計算時には、転送に必要な最大メモリーが使用されます。そのため、値は、転送で使用される実際のメモリーより大きくなる可能性があります。この理由により、enableMemoryAllocationChecking プロパティを true に設定すると、実行できる同時転送数が減少する可能性があります。メモリー不足エラーのために Managed File Transfer で障害が起こる問題が発生している場合のみ、このプロパティを true に設定することをお勧めします。大量のメモリーを消費する可能性のある転送には、メッセージのサイズが大きい場合の、ファイルからメッセージへの転送とメッセージからファイルへの転送があります。	false

表 45. 拡張エージェント・プロパティ (続き)

プロパティ名	説明	デフォルト値
enableDetailedReplyMessages	<p>このプロパティを <code>true</code> に設定すると、管理対象転送要求応答に、転送されるファイルに関する詳細な情報を含めることができます。詳細な情報とその形式は、転送ログの進行メッセージ (つまり <code><transferSet></code> エレメント) に公開されるものと同じです。詳しくは、748 ページの『ファイル転送ログ・メッセージ・フォーマット』を参照してください。</p> <p>詳細な応答情報は、管理対象転送要求で詳細な応答情報が必要であると指定した場合にのみ含まれます。この要件を指定するには、ソース・エージェントに送信される <code>managedTransfer XML</code> 要求メッセージの <code><reply></code> エレメントの <code>detailed</code> 属性を設定します。詳しくは、780 ページの『ファイル転送要求メッセージ・フォーマット』を参照してください。</p> <p>転送要求ごとに、複数の応答メッセージを生成できます。その数は、転送の転送ログ進行メッセージの数値に 1 を加えたものと同じになります (最初の応答メッセージは単純な ACK 応答です)。詳細情報は ACK 応答メッセージを除くすべてのメッセージに含まれますが、全体の転送結果は最後の詳細な応答メッセージのみに含まれます。</p>	true
enableUserMetadataOptions	<p>新しい転送要求でユーザー定義メタデータの既知のキーを使用して追加の転送オプションを指定できるかどうかを決定します。これらの既知のキーは、常に接頭部 <code>com.ibm.wmqfte.</code> で始まります。enableUserMetadataOptions プロパティが <code>true</code> に設定されると、この接頭部を使用するキーは、ユーザー定義の用途ではサポートされなくなります。enableUserMetadataOptions プロパティが <code>true</code> に設定されている場合に現在サポートされるキーは、以下のとおりです。</p> <ul style="list-style-type: none"> • <code>com.ibm.wmqfte.insertRecordLineSeparator</code> • <code>com.ibm.wmqfte.newRecordOnLineSeparator</code> • <code>com.ibm.wmqfte.convertLineSeparators</code> <p>これらのキーの意味については、437 ページの『fteCreateTransfer: 新規ファイル転送の開始』を参照してください。</p> <p>このプロパティの値は、<code>true</code> または <code>false</code> のいずれかになります。</p>	false
failTransferOnFirstFailure	<p>管理対象転送内にある転送項目が失敗するとすぐにその管理対象転送を失敗させるようにエージェントを構成できるようになります。</p> <p>この機能を有効にするには、ソース・エージェントと宛先エージェントの両方に APAR IT03450 を適用し、ソース・エージェントの <code>agent.properties</code> ファイルで <code>failTransferOnFirstFailure</code> プロパティを <code>true</code> に設定する必要があります。宛先エージェントでプロパティを <code>true</code> に設定するのは任意です。</p> <p><code>failTransferOnFirstFailure</code> プロパティが <code>true</code> に設定されると、エージェントは管理対象転送要求の処理を通常どおりに開始します。ただし、転送項目が失敗するとすぐに、管理対象転送は失敗としてマークされ、それ以降の転送項目は処理されません。管理対象転送が失敗する前に正常に処理された転送項目は、次のように処理されます。</p> <ul style="list-style-type: none"> • これらの転送項目に対してはソースの後処理が優先されます。例えば、転送項目のソース後処理が <code>delete</code> に設定されていた場合、ソース・ファイルは削除されます。 • 書き込まれた宛先ファイルは宛先ファイル・システムに残り、削除されません。 <p><code>failTransferOnFirstFailure</code> プロパティが <code>true</code> に設定されておらず、管理対象ファイル転送に複数のファイルが含まれていて、これらのファイルの 1 つが転送に失敗した場合 (例えば宛先ファイルがすでに存在していて、<code>overwrite</code> プロパティが <code>error</code> に設定されている場合など)、ソース・エージェントは続行し、要求内の残りのファイルを転送しようとします。</p>	false
itemsPerProgressMessage	<p>エージェントが次の進行状況ログ・メッセージをパブリッシュするまでに転送されるファイルの数。このプロパティを使用して、転送中に進行状況ログ・メッセージが調整キュー・マネージャーにパブリッシュされる速度を制御します。</p> <p>このプロパティに設定できる最大値は 1000 です。</p> <p>注: 進行メッセージには、前回の進行メッセージのパブリッシュ以降に転送されたすべてのファイルに関する情報が含まれます。この値を増やすと進行メッセージのサイズが大きくなるため、パフォーマンスに影響を与える可能性があります。</p>	50

表 45. 拡張エージェント・プロパティ (続き)		
プロパティ名	説明	デフォルト値
maxInlineFileSize	<p>単一のファイル間転送またはファイルからメッセージへの転送で、初期転送要求メッセージに自動的に組み込むことのできる最大ファイル・サイズ (バイト単位)。</p> <p>このプロパティを使用すると、転送の速度を改善できますが、ファイル・サイズの設定値が大きすぎると、このプロパティによってパフォーマンスが低下する可能性があります。このプロパティに推奨される初期サイズは 100 KB ですが、ご使用のシステムに最適なファイル・サイズを見つけるまでさまざまな値を十分にテストすることをお勧めします。</p>	0
入出力プロパティ:		
doNotUseTempOutputFile	<p>デフォルトでは、エージェントは宛先の一時ファイルに書き込み、ファイル転送が完了してから、この一時ファイルを必要なファイル名に変更します。この値を TRUE に設定すると、最終宛先ファイルに、エージェントが直接書き込むことになります。</p> <p> z/OS システムでは、この動作は順次 データ・セットではなく PDS データ・セット・メンバーに適用されます。</p> <p>転送の場合、このプロパティの値は、宛先エージェントによって定義されません。</p>	false
enableMandatoryLocking	<p>通常のファイルにアクセスするときに、Managed File Transfer は読み取りには共有ロックを、書き込みには排他ロックを使用します。</p> <p> Windows では、ファイル・ロックは警告のみです。このプロパティが TRUE に設定されると、Managed File Transfer はファイル・ロックを施行します。Windows の場合、これは、別のアプリケーションによりファイルが開かれていると、そのファイルのモニターはファイルが閉じられるまで起動しないことを意味します。そのファイルを含んでいる Managed File Transfer 転送は失敗します。</p> <p> UNIX タイプのプラットフォームでは、ファイル・ロックはプロセス全体を通して施行されます。UNIX タイプのプラットフォームの場合、このプロパティの設定は影響を与えません。</p> <p>このプロパティは、通常の Managed File Transfer エージェントだけに適用されます。Managed File Transfer は、ファイル・サーバーのファイル・ロック・メカニズムをサポートしていません。したがって、このプロパティはプロトコル・ブリッジ・エージェントでは無効です。プロトコル・ブリッジ・エージェントは、ファイル転送時にファイル・サーバーのファイルをロックしないからです。</p> <p>このプロパティの値は、true または false のいずれかになります。</p>	false
ioIdleThreadTimeout	スレッドがシャットダウンするまで、ファイル・システムの入出力スレッドがアイドルとなる 時間 (ミリ秒単位)。	10000
ioQueueDepth	キューに入れられる入出力要求の最大数。	10
ioThreadPoolSize	<p>有効なファイル・システムの入出力スレッドの最大数。通常は、個々の転送は各自のファイル・システム入出力スレッドを使用しますが、同時転送の数がこの制限を超過すると、ファイル・システムの入出力スレッドは、この転送間で共有されます。</p> <p>進行中の同時転送の数が、ioThreadPoolSize 値より多くなることがたびたび起きるようであれば、個々の転送が各自のファイル・システム入出力スレッドを持てるようにこの値を大きくすると改善が見られます。</p>	10
textReplacementCharacterSequence	<p>テキスト・モード転送で、データ・バイトのいずれかが、ソース・コード・ページから 宛先コード・ページに変換されない場合、ファイル転送の失敗がデフォルトの動作となります。</p> <p>転送を正常に完了できるようにこのプロパティを設定するには、特定の文字値を挿入します。このプロパティ値は、単一文字です。通常、マップ不可能な文字には疑問符 (?) が使用されます。例えば、textReplacementCharacterSequence=? という形式を使用します。ここで、疑問符 (?) は置換文字です。置換文字に空白文字を使用することはできません。</p>	なし
マルチチャネル・サポート:		

表 45. 拡張エージェント・プロパティ (続き)

プロパティ名	説明	デフォルト値
agentMultipleChannelsEnabled	<p>このプロパティを true に設定すると、Managed File Transfer Agent が複数の IBM MQ チャネルを介して転送データ・メッセージを送信できるようになります。シナリオによっては、このプロパティを設定するとパフォーマンスが改善されることがあります。ただし、マルチチャネル・サポートを使用可能にするのは、パフォーマンス上の明らかな利点がある場合のみにしてください。複数のチャネルを介して送信されるのは、SYSTEM.FTE.DATA.destinationAgentName キューに入れられるメッセージのみです。他のすべてのメッセージの動作は変更されません。</p> <p>このプロパティを true に設定した場合は、以下のいずれかのトピック中の IBM MQ 構成ステップをすべて行って、マルチチャネル・サポートを使用可能にする必要もあります。</p> <ul style="list-style-type: none"> 570 ページの『複数のチャネルに対応した MFT エージェントの構成 (クラスター)』 571 ページの『複数のチャネルに対応した MFT エージェントの構成 (非クラスター)』 <p>さらに、Managed File Transfer エージェントに必要な標準の IBM MQ 構成ステップも実行する必要があります。これについては、95 ページの『初めて使用する際の MFT の構成』で詳しく説明しています。</p> <p>このプロパティの値は、true または false のいずれかになります。</p>	false
agentMessageBatchSize	<p>複数のチャネルを指定して構成すると、ソース・エージェントはラウンドロビン・ベースで各チャネルを介して転送用のデータ・メッセージを送信します。このプロパティによって、各チャネルで一度に送信できるメッセージの数を制御します。</p>	5
マルチ・インスタンス・キュー・マネージャーのプロパティ:		
agentQMgrStandby	<p>agentQMgr によって定義されている、複数インスタンスのエージェント・キュー・マネージャーのスタンバイ・インスタンスに、クライアント接続するために使用するホスト名およびポート番号 (IBM MQ の CONNAME 形式で指定)。例えば、host_name(port_number) のように指定します。</p> <p>エージェントは、接続切断エラー (MQRC 2009 など) を検出すると、スタンバイ・キュー・マネージャーへの接続を試行します。エージェントがスタンバイ・キュー・マネージャーに接続されると、スタンバイ・キュー・マネージャーが使用不可になるまでエージェントは接続されたままになります。</p>	デフォルトなし
プロセス・コントローラーのプロパティ:		
agentQMgrRetryInterval	<p>エージェントのプロセス・コントローラーがキュー・マネージャーの使用可否を検査する間隔 (秒単位)。</p>	30
maxRestartCount	<p>maxRestartInterval プロパティの値によって指定された時間間隔内に行える再開の最大回数。この値を超えると、エージェントのプロセス・コントローラーはエージェントの再開を中止し、その代わりに maxRestartDelay プロパティの値に基づいたアクションを実行します。</p>	4
maxRestartInterval	<p>エージェントのプロセス・コントローラーがエージェントの再開数を測定する間隔 (秒単位)。この間隔内に行われた再開の回数が maxRestartCount プロパティの値を超過すると、エージェントのプロセス・コントローラーはそのエージェントの再開を中止します。エージェントのプロセス・コントローラーは、その代わりに maxRestartDelay プロパティの値に基づいたアクションを実行します。</p>	120
maxRestartDelay	<p>エージェントの再開率が maxRestartCount プロパティおよび maxRestartInterval プロパティの値を超過した場合にエージェントのプロセス・コントローラーがとる動作を決定します。ゼロ以下の値を指定すると、エージェントのプロセス・コントローラーは停止します。ゼロより大きな値を指定した場合、その秒数だけ待機した後、エージェントのプロセス・コントローラーが保持する再開履歴情報がリセットされ、エージェントが再開されます。</p>	-1
プロトコル・ブリッジ・プロパティ		
protocolBridgeCredentialConfiguration	<p>このプロパティの値は、protocolBridgeCredentialExitClasses によって指定された出口クラスの initialize() メソッドに文字列として渡されます。</p>	NULL
protocolBridgeCredentialExitClasses	<p>プロトコル・ブリッジ資格情報ユーザー出口ルーチンを実装するクラスのコンマ区切りリストを指定します。詳しくは、225 ページの『出口クラスを使用したファイル・サーバーの資格情報のマップ』を参照してください。</p>	デフォルトなし

表 45. 拡張エージェント・プロパティ (続き)		
プロパティ名	説明	デフォルト値
protocolBridgeDataTimeout	プロトコル・ブリッジ・エージェントが、FTP サーバーへのデータ接続が確立されるのを待機する場合のタイムアウト、または既に確立されている接続上で FTP サーバーからデータを受信するのを待機する場合のタイムアウト (ミリ秒)。このプロパティを値 0 に設定すると、プロトコル・ブリッジ・エージェントは無期限に待機します。タイムアウト時間に達すると、プロトコル・ブリッジ・エージェントは FTP サーバーへの既存のすべてのデータ接続を閉じて、現在の転送が再開される前に新しいデータ接続の確立を試行します。新しいデータ接続を確立しようとしたときに失敗すると、現在の転送も失敗します。	0
protocolBridgeLogoutBeforeDisconnect	FTP セッションをクローズし、切断する前に、プロトコル・ブリッジ・エージェントがファイル・サーバーからユーザーをログアウトするかどうかを指定します。このプロパティを <code>true</code> に設定すると、プロトコル・ブリッジ・エージェントはファイル・サーバーに <code>FTP QUIT</code> コマンドを発行します。	false
protocolBridgePropertiesConfiguration	<code>protocolBridgeServerPropertiesExitClasses</code> プロパティによって指定された出口クラスの <code>initialize()</code> メソッドに、ブリッジ・プロパティの 1 つとして渡されます。	デフォルトなし
protocolBridgePropertiesExitClasses	プロトコル・ブリッジ・サーバー・プロパティ・ユーザー・出口ルーチンを実装するクラスのコンマ区切りリストを指定します。 詳しくは、219 ページの『ProtocolBridgePropertiesExit2: プロトコル・ファイル・サーバー・プロパティの検索』を参照してください。	デフォルトなし
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> V 9.0.0 V 9.0.0 </div> <div> プロトコル・ブリッジ・エージェントのロギング・プロパティ </div> </div>		
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> V 9.0.0 V 9.0.0 </div> <div> agentLog </div> </div>	プロトコル・ブリッジ・エージェントと FTP/SFTP/FTPS ファイル・サーバー間の FTP コマンドと応答のロギングを有効または無効にするキー値ペア・コンポーネントと操作。 以下に例を示します。 agentLog=on すべてのコンポーネントのロギングをオンにします agentLog=off すべてのコンポーネントのロギングをオフにします agentLog=ftp=on, sftp=on, ftps=off ロギングを FTP と SFTP ではオンにして、FTPS ではオフにします	デフォルトなし
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> V 9.0.0 V 9.0.0 </div> <div> agentLogFileSize </div> </div>	キャプチャー・ファイルの最大サイズを定義します (メガバイト)。正規のトレース・デフォルト・ファイル・サイズのデフォルトと同じです。	20
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> V 9.0.0 V 9.0.0 </div> <div> agentLogFiles </div> </div>	保持されるキャプチャー・ファイルの最大数を定義します。この数を超えると、最も古いファイルが破棄されます。	10
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> V 9.0.0 V 9.0.0 </div> <div> agentLogFilter </div> </div>	デフォルトで、エージェントが接続しているすべての FTP サーバーとの通信をキャプチャーします。 以下に例を示します。 <ul style="list-style-type: none"> • ホスト/IP アドレスでのフィルター <div style="background-color: #f0f0f0; padding: 5px; margin: 5px 0;"> <pre>host=ftpprod.ibm.com, ftp2.ibm.com host=9.182.*</pre> </div> • メタデータに基づくフィルター <div style="background-color: #f0f0f0; padding: 5px; margin: 5px 0;"> <pre>metadata="outbound files to xyz corp"</pre> </div> 	*
キュー・プロパティ:		
publicationMDUser	調整キュー・マネージャーによってパブリッシュされるために送信されたメッセージと関連付ける MQMD ユーザー ID。このプロパティを設定しない場合は、MQMD ユーザー ID の設定に関する IBM MQ 規則に基づいて MQMD ユーザー ID が設定されます。	デフォルトなし
リソース・モニター・プロパティ:		
monitorFilepathPlatformSeparator	<code>\$FILEPATH</code> 変数内で、プラットフォーム固有のパス区切り文字を使用するかどうかを指定します。値が <code>true</code> の場合、プラットフォーム固有のパス区切り文字が使用されます。値が <code>false</code> の場合、UNIX スタイルのスラッシュ (<code>/</code>) パス区切り文字がすべてのプラットフォームで使用されます。	true

表 45. 拡張エージェント・プロパティ (続き)

プロパティ名	説明	デフォルト値
monitorMaxResourcesInPoll	<p>各ポーリング間隔で起動されるモニター対象リソースの最大数を指定します。例えば、モニター・パターンに*.txt、ポーリング間隔に10秒を指定し、monitorMaxResourcesInPoll プロパティを10に設定すると、monitorMaxResourcesInPoll プロパティによりエージェントはポーリング間隔ごとに最大値の10個のマッチングで起動するように制限されます。マッチング・リソースが10個の限度を超えると、その後のポーリング間隔で起動されます。</p> <p>また、monitorMaxResourcesInPoll プロパティと fteCreateMonitor コマンド上の -bs マッチング・パラメーターを併用すると、例えば各ポーリング間隔で1回のみ転送が起動されるよう制限することができます。</p> <p>ゼロ以下の値を指定すると、ポーリング間隔で起動されるモニター・リソースの数は無制限になります。</p>	-1
monitorReportTriggerFail	モニターで検出される環境および構成で発生した障害状態をログ・メッセージとして SYSTEM.FTE トピックに報告するかどうかを指定します。値を true にするとメッセージをログに記録します。値を false にするとメッセージをログに記録しません。	true
monitorReportTriggerNotSatisfied	トリガーが条件を満たさなかった場合、詳細情報を組み込んだログ・メッセージを SYSTEM.FTE トピックに送信するかどうかを指定します。値を true にするとメッセージをログに記録します。値を false にするとメッセージをログに記録しません。	false
monitorReportTriggerSatisfied	トリガーが条件を満たした場合、詳細情報を組み込んだログ・メッセージを SYSTEM.FTE トピックに送信するかどうかを指定します。値を true にするとメッセージをログに記録します。値を false にするとメッセージをログに記録しません。	false
monitorSilenceOnTriggerFailure	リソース・モニター・トリガーが連続してこの回数だけ失敗すると、障害がレポートされなくなる許容回数。	5
monitorStopOnInternalFailure	リソース・モニターの内部 FFDC 条件が連続してこの回数だけ発生すると、モニターの状態が停止に変更される許容回数。	10
ルート・ディレクトリー・プロパティ:		

表 45. 拡張エージェント・プロパティ (続き)

プロパティ名	説明	デフォルト値
commandPath	<p>以下のいずれかのメソッドを使用してコマンドを呼び出すことができるパスのセットを指定します。</p> <ul style="list-style-type: none"> エージェントの <code>Ant call</code>、<code>filecopy</code>、または <code>filemove</code> タスク エージェントに渡された XML メッセージでは、サポートされるいずれかの Managed File Transfer Agent コマンド XML スキーマ (例えば、<code>managedCall</code>、<code>managedTransfer</code>)。 <p><code>commandPath</code> プロパティの値の有効な構文については、365 ページの『commandPath MFT プロパティ』を参照してください。</p> <p>重要: 指定された <code>commandPath</code> にある任意のコマンドは、エージェントにコマンドを送信できるリモート・クライアント・システムから事実上呼び出すことができるため、このプロパティを設定するときは十分な注意を払ってください。このためデフォルトでは、<code>commandPath</code> を指定すると以下のようになります。</p> <ul style="list-style-type: none"> エージェントが開始すると、既存のエージェント・サンドボックスが構成されて、転送のためのアクセスが拒否されたディレクトリーのリストにすべての <code>commandPath</code> ディレクトリーが自動的に追加されます。 エージェントの開始時に既存のユーザー・サンドボックスが更新され、すべての <code>commandPath</code> ディレクトリー (およびそのサブディレクトリー) が <code><exclude></code> エレメントとして <code><read></code> エレメントと <code><write></code> エレメントに追加されます。 エージェントがエージェント・サンドボックスとユーザー・サンドボックスのいずれかを使用するには構成されていない場合、エージェントの開始時に、新しいエージェント・サンドボックスが作成され、その中で <code>commandPath</code> ディレクトリーが拒否ディレクトリーとして指定されます。 <p>このプロパティの値には、環境変数を含めることができます。</p> <p>詳細については、535 ページの『MFT プロパティでの環境変数の使用』を参照してください。</p> <p>V9.0.0.1 以下のリリースとの互換性が必要な場合は、<code>addCommandPathToSandbox</code> プロパティを <code>false</code> に設定することにより、このデフォルト動作をオーバーライドできます。</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 コンポーネント (またはそれ以前)。 インストール・プロパティ <code>enableFunctionalFixPack=7502</code> が設定されていないインストール済み環境での IBM WebSphere MQ 7.5.0 Fix Pack 2 Managed File Transfer コンポーネント (またはそれ以降)。 <p>重要: このオーバーライドによって事実上、クライアントはどのようなコマンドでもエージェント・システムに転送して、そのコマンドを呼び出すことができるようになります。したがって、このオーバーライドは、細心の注意を払って使用する必要があります。</p>	なし - どのコマンドも呼び出すことはできません。
<p>V9.0.0.1</p> <p>V9.0.0.1</p> <p><code>addCommandPathToSandbox</code></p>	<p><code>commandPath</code> プロパティで指定されたディレクトリー (およびそれらのすべてのサブディレクトリー) を以下に追加するかどうかを指定します。</p> <ul style="list-style-type: none"> 既存のエージェント・サンドボックスの拒否ディレクトリー。 定義されたユーザー・サンドボックスの <code><read></code> エレメントと <code><write></code> エレメントの <code><exclude></code> エレメント。 新しいエージェント・サンドボックス (エージェント・サンドボックスまたは 1 つ以上のユーザー・サンドボックスを使用するようにエージェントが構成されていない場合)。 <p>これにより、以下のリリースとの互換性が得られます。</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 コンポーネント (またはそれ以前)。 インストール・プロパティ <code>enableFunctionalFixPack=7502</code> が設定されていないインストール済み環境での IBM WebSphere MQ 7.5.0 Fix Pack 2 Managed File Transfer コンポーネント (またはそれ以降)。 <p>詳しくは、365 ページの『commandPath MFT プロパティ』を参照してください。</p>	True

表 45. 拡張エージェント・プロパティ (続き)

プロパティ名	説明	デフォルト値
<p>V9.0.1 V9.0.0.1 V9.0.1 V9.0.0.1</p> <p>additionalWildcardSandboxChecking</p>	<p>エージェントがファイルの転送先/転送元として使用できる場所を制限するために、ユーザー・サンドボックスまたはエージェント・サンドボックスが構成されているエージェントで、ワイルドカード転送に対する追加検査を行うかどうかを指定します。</p> <p>このプロパティが true に設定されている場合、追加の検査が有効になります。転送要求でワイルドカードのファイル・マッチング用に定義されたサンドボックスの外側にある場所の読み取りが試行されると、転送は失敗します。1つの転送要求内に複数の転送があり、サンドボックスの外側にある場所を読み取ろうとしたためにこれらの要求のいずれかが失敗した場合、転送全体が失敗します。チェックが失敗すると、失敗の理由がエラー・メッセージに示されず (50 ページの『ワイルドカード転送の追加検査』を参照)。</p> <p>このプロパティが省略されるか、false に設定されると、ワイルドカード転送で追加の検査は行われません。</p>	なし
sandboxRoot	<p>サンドボックスを使用する際に組み込むルート・パス・セットおよび除外するルート・パス・セットを指定します。この機能については、45 ページの『MFT エージェント・サンドボックスの処理』を参照してください。</p> <p>パスは、プラットフォーム固有の分離文字で区切ります。パスの接頭部に感嘆符文字 (!) 文字を付けて、パスをサンドボックスから除外することを示してください。この機能は、組み込んだルート・パスの下の1つのサブディレクトリーを除外する場合に便利です。</p> <p>sandboxRoot プロパティは、プロトコル・ブリッジ・エージェントではサポートされません。</p> <p>sandboxRoot プロパティと userSandboxes プロパティを一緒に指定することはできません。</p> <p>このプロパティの値には、環境変数を含めることができます。</p> <p>詳細については、535 ページの『MFT プロパティでの環境変数の使用』を参照してください。</p>	なし - サンドボックスなし
transferRoot	<p>エージェントに指定された相対パスのデフォルトのルート・ディレクトリー。このプロパティの値には、環境変数を含めることができます。</p> <p>詳細については、535 ページの『MFT プロパティでの環境変数の使用』を参照してください。</p>	エージェント・プロセスを開始したユーザーのホーム・ディレクトリー。
transferRootHLQ	<p>エージェントに指定された非完全修飾データ・セットのデフォルトの HLQ (ユーザー ID)</p>	エージェント・プロセスを開始したユーザーのユーザー名。
userSandboxes	<p>転送を要求するユーザーの MQMD ユーザー名に基づいて、ファイルの転送元や転送先として使用できるファイル・システムの領域を制限します。詳しくは、47 ページの『MFT ユーザー・サンドボックスの処理』を参照してください。</p> <p>userSandboxes プロパティは、プロトコル・ブリッジ・エージェントではサポートされません。</p> <p>sandboxRoot プロパティと userSandboxes プロパティを一緒に指定することはできません。</p>	false
スケジューラー・プロパティ:		
maxSchedulerRunDelay	<p>エージェントがスケジュール済み転送の検査を待機する最大間隔 (分単位)。このプロパティを使用可能にするには、正整数を指定します。このプロパティの使用理由について詳しくは、314 ページの『スケジュール済みファイル転送が実行されなかったり遅延したりする場合の対策』を参照してください。</p> <p>スケジュール済み転送の実行予定時刻にエージェントがコマンド・キューからコマンドを読み取っている場合があるので、スケジュール済み転送の開始前に追加の遅延が発生する可能性があります。この場合、スケジューラーはそのコマンドが完了した直後に実行します。</p>	-1
セキュリティ・プロパティ:		

表 45. 拡張エージェント・プロパティ (続き)

プロパティ名	説明	デフォルト値
agentQMGrAuthenticationCredentialsFile	MQ 接続の資格情報が含まれているファイルへのパス。	 認証資格情報ファイルの作成に関する詳細は、MFT 資格情報ファイルの作成を参照してください。  このファイルの場所およびアクセス権については、MQMFTCredentials.xml の構成を参照してください。  認証資格情報ファイルの作成について詳しくは、MFT と IBM MQ の接続認証を参照してください。
authorityChecking	『359 ページの『MFT エージェント・アクションのユーザー権限の制限』』に記載されているセキュリティ機能を有効にするかどうかを指定します。  IBM MQ 9.0.0 Fix Pack 3 および IBM MQ 9.0.4 より、すべてのエージェント権限キューで inquire が必須の権限になりました。	false
logAuthorityChecks	output0.log ファイルでエージェントのイベント・ログに記録する権限検査ロギングのレベル。このプロパティがとることのできる値は None、Failures、または All です。	なし
userIdForClientConnect	IBM MQ へのクライアント接続を介してフローされるユーザー ID。java が指定されている場合、JVM によって報告されるユーザー名が、IBM MQ 接続要求の一部として送られます。このプロパティがとることのできる値は None または java です。	なし
タイムアウト・プロパティ:		
maxTransferNegotiationTime	宛先エージェントがネゴシエーションを完了するまで転送が待機する最大時間 (ミリ秒)。この時間内にネゴシエーションが完了しない場合、転送は再同期状態に置かれ、別の転送が使用可能な場合はその転送を実行できます。 ソース・エージェントまたは宛先エージェントに大きな負荷がかかるシナリオの場合、エージェントが時間内にネゴシエーション要求に対応するには、デフォルト値では小さすぎる可能性があります。その可能性が特に高いのは、ソース・エージェントに多数のリソース・モニターが定義されている場合、あるいはソース・エージェントのリソース・モニターが大量のファイルを含むディレクトリをモニターしている場合です。その一方、数多くの転送要求がエージェントに送信される場合にも、この事態が起こる可能性があります。そのようなシナリオでは、このプロパティの値を 200,000 以上に設定しなければなりません。	30 000
recoverableTransferRetryInterval	リカバリー可能な転送エラーを検出してから転送の再開を試みるまでの待機時間 (ミリ秒)。	60 000
senderTransferRetryInterval	宛先で既に最大転送数が実行されているために、拒否された転送が再試行されるまで待機する時間 (ミリ秒)。最小値は 1000 です。	30 000
transferAckTimeout	相手側からの確認応答またはデータを転送が待機して再試行を実行するまでのタイムアウト (ミリ秒単位)。これは拡張プロパティであり、ほとんどの Managed File Transfer 構成では、このプロパティの値を変更する必要はありません。 確認応答は、完全なデータ・ウィンドウが受信されると、常に受信側エージェントから送信側エージェントへ送信されます。処理能力が制限されている、または信頼性のないネットワークの場合、および agentWindowSize と agentChunkSize の設定値が大きい場合、デフォルト値では十分な長さでない可能性があります。この場合、エージェント間でデータが不要に再転送されることがあります。したがって、この値を大きくすることが有効で、ネットワーク速度が遅いため転送がリカバリー・モードになる可能性を減らすことができます。	60 000
transferAckTimeoutRetries	エージェントが転送を中止してリカバリー状態に移行するまで行う、応答のない転送に対する確認応答の最大試行回数。	5

表 45. 拡張エージェント・プロパティ (続き)		
プロパティ名	説明	デフォルト値
xmlConfigReloadInterval	<p>ランタイム中にエージェントがXML 構成ファイルを再ロードする間隔(秒単位)。ランタイム中にエージェントがXML 構成ファイルを再ロードしないようにするには、このプロパティを-1に設定してください。このプロパティによって、以下のXML 構成ファイルが影響を受けます。</p> <ul style="list-style-type: none"> ConnectDirectCredentials.xml ConnectDirectNodeProperties.xml ConnectDirectProcessDefinitions.xml ProtocolBridgeCredentials.xml ProtocolBridgeProperties.xml UserSandboxes.xml 	30
トレースおよびロギング・プロパティ:		
javaCoreTriggerFile	<p>エージェントがモニターするファイルの場所への絶対パス。指定された場所にファイルが存在する場合、エージェントの開始によって Javacore が起動されます。エージェントが開始した後、この場所でファイルを更新すると、エージェントは再度 Javacore ファイルを起動します。</p> <p>別スレッドがこのファイルを 30 秒ごとにポーリングし、ファイルの作成や更新が行われたかどうかを検査します。前回のポーリング以降にファイルが作成されたか更新された場合、エージェントは以下のディレクトリーのいずれかに Javacore ファイルを生成します。</p> <ul style="list-style-type: none"> UNIX: <code>MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name</code> Linux: <code>MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name</code> Windows: <code>MQ_DATA_PATH\mqft\logs\coordination_qmgr_name\agents\agent_name</code> <p>このプロパティを指定すると、エージェントは始動時に以下のメッセージを出力します。</p> <pre>BFGAG0092I The <insert_0> file will be used to request JVM diagnostic information.</pre> <p>このプロパティの値には、環境変数を含めることができます。</p> <p>詳細については、535 ページの『MFT プロパティでの環境変数の使用』を参照してください。</p>	なし
トレース	<p>エージェントの開始時にトレースを有効にしてエージェントを実行するトレース仕様。トレース仕様は、クラス、等号、およびトレース・レベルのコンマ区切りのリストです。例えば <code>com.ibm.wmqfte.agent.Agent,com.ibm.wmqfte.commandhandler=all</code> です。コロン区切りのリストで、複数のトレース仕様を指定できます。例えば <code>com.ibm.wmqfte.agent.Agent=all:com.ibm.wmqfte.commandhandler=moderate</code> です。</p>	なし
outputLogFiles	保持する output.log ファイルの合計数。この値は、エージェント自体に加えて、エージェントのプロセス・コントローラーにも適用されます。	5
outputLogSize	出力が次のファイルで折り返す前の各 output.log ファイルの最大サイズ (MB 単位)。この値は、エージェント自体に加えて、エージェントのプロセス・コントローラーにも適用されます。	1
outputLogEncoding	エージェントが output.log ファイルを書き込む際に使用する文字エンコード。	エージェントが実行されているプラットフォームのデフォルトの文字エンコード。
traceFiles	保持するトレース・ファイルの合計数。この値は、エージェント自体に加えて、エージェントのプロセス・コントローラーにも適用されます。	5
traceSize	トレースが次のファイルで折り返す前の各トレース・ファイルの最大サイズ (MB 単位)。この値は、エージェント自体に加えて、エージェントのプロセス・コントローラーにも適用されます。	20
traceMaxBytes	トレース・ファイルに出力されるメッセージ・データ量の制限。	4096 バイト
logTransferRecovery	このプロパティを true の値に設定すると、転送がリカバリー状態になるたびに診断イベントがエージェントのイベント・ログである output0.log ファイルに報告されます。	IBM MQ 9.0.0.0 の場合、デフォルト値は true です。

表 45. 拡張エージェント・プロパティ (続き)

プロパティ名	説明	デフォルト値
logCapture	このエージェントに送信された転送要求メッセージと、エージェントが調整キュー・マネージャーにパブリッシュしたログ・メッセージをキャプチャーします。キャプチャーされるこれらのメッセージは、転送の問題をデバッグする際に役立ちます。キャプチャーされたメッセージは、capture?.log という名前のエージェント・ログ・ディレクトリーにあるファイルに保管されます。? は数値です。数字 0 を含むファイルには、キャプチャーされた最新のメッセージが保持されています。	false
logCaptureFileSize	キャプチャー・ファイルの最大サイズを定義します (メガバイト)。	10
logCaptureFiles	保持されるキャプチャー・ファイルの最大数を定義します。この数を超えると、最も古いファイルが破棄されます。	10
logCaptureFilter	エージェントがメッセージのトピック名の突き合わせに使用する Java 正規表現。この正規表現に一致したメッセージのみキャプチャーされます。	*(match all)
  resourceMonitorLog	リソース・モニターと、ロギングをオンまたはオフにする操作のキーと値のペア。 指定できる値は以下のとおりです。 <ul style="list-style-type: none"> • info • moderate • 詳細 • off 以下に例を示します。 <ul style="list-style-type: none"> • resourceMonitorLog=MON1,MON2=info:MON3=off MON1 と MON2 のロギングをオンにし、MON3 のロギングをオフにします。 • resourceMonitorLog=info すべてのリソース・モニターで info レベルのロギングをオンにします。 リソース・モニター・ログは、resmoneventN.log という名前のファイルに書き込まれます。ここで、N は数値を表します。例えば、resmonevent0.log です。 <p> 重要: 1つのエージェントのリソース・モニターはすべて同じログ・ファイルに書き込まれます。</p> 詳しくは、MFT リソース・モニターのロギングを参照してください。	info
  resourceMonitorLogFileSize	キャプチャー・ファイルの最大サイズを定義します (メガバイト)。	20
  resourceMonitorLogFiles	保持されるキャプチャー・ファイルの最大数を定義します。この数を超えると、最も古いファイルが破棄されます。	10
転送制限プロパティ:		
maxDestinationTransfers	宛先エージェントが時点を問わず常に一度に処理できる並行転送の最大数。ここで指定する合計値でカウントされるのは、エージェントに実行依頼されるそれぞれの転送要求です。要求を満たすために転送されるファイルの数は無関係です。つまり、1 個のファイルを転送する転送要求も、10 個のファイルを転送する転送要求も、同じようにカウントされる、ということです。 宛先エージェントが maxDestinationTransfers プロパティで指定された制限に達すると、エージェント・キューが転送します。 エージェント・プロパティ値の合計 (maxSourceTransfers + maxDestinationTransfers + maxQueuedTransfers) が状態格納キュー (SYSTEM.FTE.STATE.agent name) の MAXDEPTH 設定の値を超えると、エージェントは開始しません。	25 (Connect:Direct 以外のすべてのエージェントの場合) 5 (Connect:Direct ブリッジ・エージェントの場合)
maxFilesForTransfer	1 つの管理対象転送に許可される転送項目の最大数。管理対象転送に含まれる項目数が maxFilesForTransfer の値を上回る場合、管理対象転送は失敗し、それ以降の転送項目は処理されません。 このプロパティを設定すると、ユーザーが UNIX システムのルート・ディレクトリー「/」の転送を誤って指定した場合など、誤った転送要求により大量のファイルが転送されてしまうのを防ぐことができます。	5000

表 45. 拡張エージェント・プロパティ (続き)		
プロパティ名	説明	デフォルト値
maxSourceTransfers	<p>ソース・エージェントが時点を問わず常に一度に処理できる並行転送の最大数。ここで指定する合計値でカウントされるのは、エージェントに実行依頼されるそれぞれの転送要求です。要求を満たすために転送されるファイルの数は無関係です。つまり、1個のファイルを転送する転送要求も、10個のファイルを転送する転送要求も、同じようにカウントされる、ということです。</p> <p>宛先エージェントが maxSourceTransfers プロパティで指定された制限に達すると、ソース・エージェント・キューが転送します。</p> <p>エージェント・プロパティ値の合計 (maxSourceTransfers + maxDestinationTransfers + maxQueuedTransfers) が状態格納キュー (SYSTEM.FTE.STATE.agent name) の MAXDEPTH 設定の値を超えると、エージェントは開始しません。</p>	<p>25 (Connect:Direct ブリッジ・エージェント以外のすべてのエージェントの場合)</p> <p>5 (Connect:Direct ブリッジ・エージェントの場合)</p>
maxQueuedTransfers	<p>エージェントが新しい転送要求を拒否するまでに、ソース・エージェントによってキューに入れることができる保留中の転送の最大数。このプロパティを設定して、maxDestinationTransfers や maxSourceTransfers が制限に達した場合、あるいは制限を超えた場合でも、その時点で作成した新規の転送要求が受け入れられ、キューに入れられ、後で実行されるようにすることができます。</p> <p>キューに入れられた転送要求が処理される順序は、その優先順位とキューに入れられている期間の係数です。古くて、優先順位が高い保留中の転送が最初に選択されます。長期間キューに入れられている、優先順位が低い転送は、より新しく優先順位が高い転送に優先して選択されます。</p> <p>エージェント・プロパティ値の合計 (maxSourceTransfers + maxDestinationTransfers + maxQueuedTransfers) が状態格納キュー (SYSTEM.FTE.STATE.agent name) の MAXDEPTH 設定の値を超えると、エージェントは開始しません。</p>	1000
<p>V 9.0.1 V 9.0.1 転送リカバリー・タイムアウト・プロパティ:</p>		
<p>V 9.0.1</p> <p>V 9.0.1</p> <p>transferRecoveryTimeout</p>	<p>停止したファイル転送のリカバリーをソース・エージェントが試行し続ける時間 (秒単位) を設定します。</p> <p>このプロパティを設定しない場合のエージェントのデフォルトの動作は、転送が正常にリカバリーされるまで試行し続ける動作です。転送リカバリー・タイムアウト・プロパティには、以下の値を設定できます。</p> <p>-1</p> <p>エージェントは、停止した転送のリカバリーを、転送が完了するまで試行し続けます。このオプションを使用すると、このプロパティを設定しない場合のエージェントのデフォルトの動作と同じになります。</p> <p>0</p> <p>エージェントは、リカバリーに入るとすぐにファイル転送を停止します。</p> <p>>0</p> <p>エージェントは、指定された正整数値で設定された時間 (秒単位) だけ、停止した転送のリカバリーを試行し続けます。例えば、transferRecoveryTimeout=21600 は、エージェントがリカバリーに入ってから6時間にわたって転送のリカバリーを試行し続けることを示しています。このパラメーターの最大値は 99999999. です。</p>	-1
<p>ユーザー出力ルーチン・プロパティ:</p>		
agentForceConsistentPathDelimiters	<p>転送出口に対して指定するソース・ファイル情報と宛先ファイル情報のパス区切り文字を強制的に UNIX スタイルのスラッシュ (/) にします。有効なオプションは、true と false です。</p>	false
destinationTransferEndExitClasses	<p>宛先転送ユーザー出力ルーチンを実装するクラスのコンマ区切りリストを指定します。</p>	デフォルトなし
destinationTransferStartExitClasses	<p>宛先転送開始ユーザー出力ルーチンを実装するクラスのコンマ区切りリストを指定します。</p>	デフォルトなし
exitClassPath	<p>ユーザー出力ルーチンのクラスパスの役割を果たす、プラットフォーム固有のディレクトリーの文字区切りリストを指定します。</p> <p>エージェントの出口ディレクトリーは、このクラスパスにある項目の前に検索されます。</p>	エージェントの exits ディレクトリー
exitNativeLibraryPath	<p>ユーザー出力ルーチンのネイティブ・ライブラリー・パスの役割を果たす、プラットフォーム固有のディレクトリーの文字区切りリストを指定します。</p>	エージェントの exits ディレクトリー

表 45. 拡張エージェント・プロパティ (続き)		
プロパティ名	説明	デフォルト値
ioMaxRecordLength	レコード単位ファイルのサポート可能な最大レコード長 (バイト単位)。Managed File Transfer は、レコード単位ファイルへの書き込みをいかなるレコード長でもサポートできます。しかし、レコード長が大きいとメモリー不足エラーになる可能性があるため、このエラーが発生しないように、最大レコード長はデフォルトで 64 K に制限されます。レコード単位ファイルから読み取る場合は、レコード全体が単一の転送チャンクに収まらなければならないので、レコード長は転送チャンク・サイズによりさらに制限されます。このプロパティは入出力ユーザー出口のレコード単位ファイル専用です。	64 KB
monitorExitClasses	モニター出口ルーチンを実装するクラスのコンマ区切りリストを指定します。詳しくは、875 ページの『MFT リソース・モニター・ユーザー出口』を参照してください。	デフォルトなし
protocolBridgeCredentialExitClasses	プロトコル・ブリッジ資格情報ユーザー出口ルーチンを実装するクラスのコンマ区切りリストを指定します。詳しくは、225 ページの『出口クラスを使用したファイル・サーバーの資格情報のマップ』を参照してください。	デフォルトなし
sourceTransferEndExitClasses	ソース転送終了出口ルーチンを実装するクラスのコンマ区切りリストを指定します。	デフォルトなし
sourceTransferStartExitClasses	ソース転送開始出口ルーチンを実装するクラスのコンマ区切りリストを指定します。	デフォルトなし
IOExitClasses	入出力ユーザー出口ルーチンを実装するクラスのコンマ区切りリストを指定します。IOExit インターフェースを実装するクラスのみリストします。つまり、IOExitResourcePath や IOExitChannel などの他の入出力ユーザー出口インターフェースを実装するクラスはリストしないでください。詳しくは、260 ページの『MFT 転送入出力ユーザー出口の使用』を参照してください。	デフォルトなし
IBM MQ クライアント圧縮:		
agentDataCompression	このプロパティは、クライアント接続でのみサポートされます。 リモート IBM MQ サーバーとネゴシエーションするための、ファイル・データ転送用の圧縮タイプのコンマ区切りリスト。これらの圧縮タイプについては、メッセージ・データ圧縮リストのトピックを参照してください。 これらの値は、妥当性検査後に、エージェントのクライアント・チャンネルに (指定されている順序で) プロパティとしてパススルーされます。次いで IBM MQ クライアントは、このクライアント・チャンネルとリモート・サーバー・チャンネルの間のネゴシエーションを処理して、その 2 つのチャンネルの圧縮プロパティ間でマッチングに関する最低限の共通基準を検出します。一致項目が見つからない場合は、MQCOMPRESS_NONE が常に選択されます。	MQCOMPRESS_NONE
agentHeaderCompression	このプロパティは、クライアント接続でのみサポートされます。 リモート IBM MQ サーバーとネゴシエーションするための、ヘッダー・データ転送用の圧縮タイプのコンマ区切りリスト。受け入れられる値は MQCOMPRESS_NONE または MQCOMPRESS_SYSTEM です。これらの圧縮タイプについては、メッセージ・ヘッダー圧縮リストのトピックを参照してください。 これらの値は、妥当性検査後に、エージェントのクライアント・チャンネルに (指定されている順序で) プロパティとしてパススルーされます。次いで IBM MQ クライアントは、このクライアント・チャンネルとリモート・サーバー・チャンネルの間のネゴシエーションを処理して、その 2 つのチャンネルの圧縮プロパティ間でマッチングに関する最低限の共通基準を検出します。一致項目が見つからない場合は、MQCOMPRESS_NONE が常に選択されます。	MQCOMPRESS_NONE
  z/OS 固有:		
      adminGroup	セキュリティ・マネージャー・グループ。このグループのメンバーは以下を実行できます。 <ul style="list-style-type: none"> • fteStartAgent コマンドを使用してエージェントを開始します。 • fteStopAgent コマンドを使用してエージェントを停止します。 • fteSetAgentTraceLevel コマンドを使用してエージェントのトレースを有効または無効にする。 • -d パラメーターを指定して fteShowAgentDetails コマンドを実行し、ローカル・エージェントの詳細を表示します。 セキュリティ・マネージャー・グループ (例えば MFTADMIN) を定義し、開始済みタスクのユーザー ID と管理者 TSO ID をこのグループに追加します。エージェント・プロパティ・ファイルを編集し、 adminGroup プロパティの値をこのセキュリティ・マネージャー・グループの名前に設定します。 <pre>adminGroup=MFTADMIN</pre>	なし

表 45. 拡張エージェント・プロパティ (続き)

プロパティ名	説明	デフォルト値
  bpxwdynAllocAdditionalOptions	Managed File Transfer は、BPXWDYN テキスト・インターフェースを使用して、z/OS データ・セットの作成とオープンを行います。データ・セットの割り振りに BPXWDYN を使用すると、Managed File Transfer はデフォルトで、可能であれば、データ・デバイスがマウントされるようにします (ディスク・ベースのデータ・セットでは不要ですが、テープ・データ・セットの場合は必要です)。このオプションは特定の環境でサポートされない場合があるので、このプロパティを使用してこの動作を変更します。また、データ・セットへの転送時に、コマンド行で BPXWDYN のオプションを指定することもできます。これらのオプションは、このプロパティで指定したオプションに追加されず。 いくつかの BPXWDYN オプションは、 bpxwdynAllocAdditionalOptions プロパティを agent.properties ファイルで使用するときには指定してはなりません。これらのプロパティのリストについては、 642 ページの『MFT で使用できない BPXWDYN のプロパティ』 を参照してください。	デフォルトは以下のとおりです。 • MOUNT (z/OS V1R8 以降の場合)
  armELEMTYPE	オプションのプロパティ。エージェントを自動リスタート・マネージャー (ARM) で再始動するように構成する場合は、関連付けられた ARM ポリシー内で指定された ARM ELEMTYPE パラメータ値をこのプロパティに設定します。エージェントの場合、ELEMTYPE は SYSBFGAG に設定します。	設定なし
  armELEMENT	オプションのプロパティ。エージェントを自動リスタート・マネージャー (ARM) で再始動するように構成する場合は、関連付けられた ARM ポリシー内で指定された ARM ELEMENT パラメータ値をこのプロパティに設定します。ELEMENT 値には、エージェント名に対応する値を設定できます。	設定なし
他のプロパティ:		
  legacyXMLMessageMQMDFormat	エージェントにより生成される Managed File Transfer XML メッセージ (例えば、ログおよび転送の進行状況メッセージ) は、ブランクの MQMD 形式フィールドと共に、キューに送信されるようになりました。以前のバージョンの製品では、MQMD 形式フィールドを MQSTR (テキスト・メッセージ・ストリング) に設定していました。このプロパティを true に設定すると、エージェントにより生成される Managed File Transfer XML メッセージを、MQSTR の MQMD 形式フィールドと共にキューに送信することが可能になります。 注: コマンドへのエージェント応答メッセージは、対応するコマンド要求に一致するメッセージ形式で送信されます。 MQMD 形式フィールドが MQSTR に設定されている場合、データ変換が使用可能になっている MQ ネットワーク内のチャンネルが存在すると、Managed File Transfer コマンドの XML メッセージが壊れる可能性があります。	false
 adjustScheduleTimeForDaylightSav g	企業が以下のようにスケジュール転送を作成して、毎日スケジュールされた転送を実行しているとしたします。 • -oi パラメータを日にち • -tb パラメータをソース に設定した fteCreateTransfer コマンドで毎日実行している場合、このプロパティを true に設定すると、時計が 1 時間進んだらスケジュールされた転送の時間を 1 時間進め、時計が 1 時間戻ったら 1 時間戻します。 例えば、スケジュールされた転送が 1:00 AM に設定されている場合、時計が進むと転送は 2:00 AM に実行され、時計が戻ると転送は 1:00 AM に戻ります。	true

関連概念

57 ページの『MFT 構成オプション (Multiplatforms)』

Managed File Transfer は、セットアップに関する重要な情報が含まれた、操作に必要ないくつかのプロパティ・ファイルを提供します。これらのプロパティ・ファイルは、製品のインストール時に定義される構成ディレクトリにあります。

246 ページの『ファイル転送のリカバリーのタイムアウト・オプション』

停止したファイル転送のリカバリーをソース・エージェントが試行し続ける時間を秒単位で設定できます。転送が成功しないままエージェントが再試行間隔のタイムアウトに達した場合、その転送は失敗します。

45 ページの『MFT のサンドボックス』

ファイル・システムの中で、エージェントが転送処理時にアクセスできる領域を制限できます。エージェントがアクセスできる制限領域のことをサンドボックスといいます。制限の適用対象は、エージェントにすることも、転送を要求するユーザーにすることも可能です。

関連資料

605 ページの『MFT Java システム・プロパティ』

いくつかの Managed File Transfer コマンド・プロパティとエージェント・プロパティは Java システム・プロパティとして定義する必要があります。これは、コマンド・プロパティおよびエージェント・プロパティのメカニズムを使用できない初期の関数用の構成を定義するためです。

605 ページの『MFT の SSL プロパティ』

SSL または TLS を IBM MQ および Managed File Transfer とともに使用して、エージェントとキュー・マネージャーとの間の許可されない接続を防止し、エージェントとキュー・マネージャーとの間のメッセージ・トラフィックを暗号化します。

388 ページの『fteCreateAgent (MFT エージェントの作成)』

fteCreateAgent コマンドは、Managed File Transfer Agent とその関連構成を作成します。

393 ページの『fteCreateBridgeAgent (MFT プロトコル・ブリッジ・エージェントの作成および構成)』

fteCreateBridgeAgent コマンドは、Managed File Transfer プロトコル・ブリッジ・エージェントとその関連構成を作成します。プロトコル・ブリッジ・エージェントは、ファイルを送受信するファイル・サーバーごとに作成します。

402 ページの『fteCreateCDAgent (Connect:Direct ブリッジ・エージェントの作成)』

fteCreateCDAgent コマンドは、Managed File Transfer Agent、および Connect:Direct ブリッジのための関連する構成を作成します。

535 ページの『MFT プロパティでの環境変数の使用』

IBM WebSphere MQ 7.5 以降、ファイルまたはディレクトリ of the場所を表す Managed File Transfer プロパティで環境変数を使用できます。これにより、製品の一部の実行時に使用されるファイルまたはディレクトリの場所を、環境の変更に合わせて変えることができます。例えば、プロセスを実行するユーザーの変更などです。

複数のチャネルに対応した MFT エージェントの構成 (クラスター)

クラスター構成で IBM MQ マルチチャネル・サポートを使用する場合、最初に `agentMultipleChannelsEnabled` プロパティを `true` に設定してから、このトピックのステップを実行してください。

このタスクについて

クラスターの場合、マルチチャネル・サポートは、宛先エージェントのみのキュー・マネージャーの IBM MQ 定義で使用可能にされます。

95 ページの『初めて使用する際の MFT の構成』にリストされている Managed File Transfer エージェントに必要な標準の IBM MQ 構成ステップに加えて、このトピックのステップを実行する必要があります。

以下の構成例では、**runmqsc** コマンドを使用します。

手順

1. 使用するチャネルごとに、クラスター受信側チャネルを定義します。例えば、2つのチャネルを使用する場合、以下のようになります。

```
DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(CLUSRCVR) CLUSTER(MFTCLUSTER)
DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(CLUSRCVR) CLUSTER(MFTCLUSTER)
```

ここで、

- `DESTQMGRNAME` は、宛先エージェントのキュー・マネージャーの名前です。
- `MFTCLUSTER` は IBM MQ クラスターの名前です。

チャネルには `MFTCLUSTER.DESTMGRNAME_n` という命名規則を使用することが推奨されていますが、この規則は必須ではありません。

2. 各チャネルに対応するキュー・マネージャー別名を定義します。以下に例を示します。

```
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_1) RQMNAME(DESTQMGRNAME) CLUSTER(MFTCLUSTER)
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_2) RQMNAME(DESTQMGRNAME) CLUSTER(MFTCLUSTER)
```

キュー・マネージャー別名には `SYSTEM.FTE.DESTQMGRNAME_n` という命名規則を使用する必要があります。送信側エージェントがこの形式のキュー・マネージャー別名を検索するためです。n に使用する番号は、1 から開始して、連続している必要があります。ソース・エージェントのキュー・マネージャーでも使用可能にするために、クラスター全体で定義する必要があります。

ソース・エージェントと宛先エージェントの両方にキュー・マネージャー別名の番号を正しく判別させるには、キュー・マネージャーのデフォルト `XMITQ` を定義しないでください。

関連概念

95 ページの『初めて使用する際の MFT の構成』

Managed File Transfer のエージェントおよびキュー・マネージャーの構成タスクの中には、初回の使用時に一度実行すれば済むものもあります。

関連資料

550 ページの『MFT agent.properties ファイル』

各 Managed File Transfer Agent には、独自のプロパティ・ファイル `agent.properties` があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。agent.properties ファイルには、エージェントの動作を変更するプロパティを含めることもできます。

複数のチャネルに対応した MFT エージェントの構成 (非クラスター)

非クラスター構成で IBM MQ マルチチャネル・サポートを使用する場合、最初に `agentMultipleChannelsEnabled` プロパティを `true` に設定してから、このトピックのステップを実行してください。

このタスクについて

非クラスター構成の場合、マルチチャネル・サポートは、ソース・エージェントと宛先エージェントの両方のキュー・マネージャーの IBM MQ 定義で使用可能にされます。

95 ページの『初めて使用する際の MFT の構成』にリストされている Managed File Transfer エージェントに必要な標準の IBM MQ 構成ステップに加えて、このトピックのステップを実行する必要があります。

以下のステップでは、送信側と受信側のチャネルがソース・キュー・マネージャーと宛先キュー・マネージャー間の通信で使用されていると想定しています。

以下の構成例では、`runmqsc` コマンドを使用します。

手順

1. 宛先エージェントのキュー・マネージャーで、使用するチャネルごとに、受信側チャネルを定義します。例えば、2 つのチャネルを使用する場合、以下のようになります。

```
DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(RCVR) TRPTYPE(TCP)
DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(RCVR) TRPTYPE(TCP)
```

ここで、`DESTQMGRNAME` は、宛先エージェントのキュー・マネージャーの名前です。

チャネルには `TO.DESTMGRNAME_n` という命名規則を使用することが推奨されていますが、この規則は必須ではありません。受信側チャネルの名前は、ソース・エージェントのキュー・マネージャーの対応する送信側チャネルと一致する必要があります。

2. ソース・エージェントのキュー・マネージャーで、使用するチャネルごとに、伝送キューを定義します。例えば、2 つのチャネルを使用する場合、以下のようになります。

```
DEFINE QLOCAL(DESTQMGRNAME_1) USAGE(XMITQ)
DEFINE QLOCAL(DESTQMGRNAME_2) USAGE(XMITQ)
```

伝送キューには `DESTMGRNAME_n` という命名規則を使用することが推奨されていますが、この規則は必須ではありません。定義する伝送キューは、以下のステップの送信側チャネル定義およびキュー・マネージャー別名定義で参照されます。

- ソース・エージェントのキュー・マネージャーで、使用するチャンネルごとに、送信側チャンネルを定義します。例えば、2つのチャンネルを使用する場合、以下のようになります。

```
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)
```

チャンネルには TO.DESTMGRNAME_n という命名規則を使用することが推奨されていますが、この規則は必須ではありません。送信側チャンネルの名前は、宛先エージェントのキュー・マネージャーの対応する受信側チャンネルと一致する必要があります。

- ソース・エージェントのキュー・マネージャーで、各チャンネルに対応するキュー・マネージャー別名を定義します。以下に例を示します。

```
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_1) RQMNAME(DESTQMGRNAME) XMITQ(DESTQMGRNAME_1)
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_2) RQMNAME(DESTQMGRNAME) XMITQ(DESTQMGRNAME_2)
```

キュー・マネージャー別名には SYSTEM.FTE.DESTQMGRNAME_n という命名規則を使用する必要があります。送信側エージェントがこの形式のキュー・マネージャー別名を検索するためです。n に使用する番号は、1 から開始して、連続している必要があります。

エージェントにキュー・マネージャー別名の番号を正しく判別させるには、キュー・マネージャーのデフォルト XMITQ を定義しないでください。

関連概念

95 ページの『初めて使用する際の MFT の構成』

Managed File Transfer のエージェントおよびキュー・マネージャーの構成タスクの中には、初回の使用時に一度実行すれば済むものもあります。

関連資料

550 ページの『MFT agent.properties ファイル』

各 Managed File Transfer Agent には、独自のプロパティ・ファイル agent.properties があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。agent.properties ファイルには、エージェントの動作を変更するプロパティを含めることもできます。

enableUserMetadataOptions: サポートされている MFT ユーザー定義メタデータ・キー

エージェント・プロパティ **enableUserMetadataOptions** が true の値に設定されている場合、以下のユーザー定義のメタデータ・キーを新規の転送要求に対して指定すると、これらのメタデータ・キーがサポートされます。

表 46. メタデータ・キー		
キーの名前	説明	デフォルト値
com.ibm.wmqfte.insertRecordLineSeparator	テキスト転送用。このキーを true に設定すると、z/OS データ・セットなどのレコード単位のファイルを読み取る際に、レコードと次のレコードとの間に行分離文字が挿入されます。 このキーを false に設定すると、レコード単位のファイルを読み取る際に、レコード間に行分離文字は挿入されません。	true
com.ibm.wmqfte.newRecordOnLineSeparator	テキスト転送用。このキーを true に設定すると、z/OS データ・セットなどのレコード単位のファイルに書き込むときに、行分離文字は新規レコードを示し、データの一部として書き込まれません。 このキーを false に設定すると、レコード単位のファイルに書き込むときに、行分離文字は他のあらゆる文字と同じように扱われます（つまり、レコードは分割されません）。	true

表 46. メタデータ・キー (続き)		
キーの名前	説明	デフォルト値
com.ibm.wmqfte.convertLineSeparators	<p>テキスト転送用。行分離文字シーケンスの CRLF および LF を宛先に必要な行分離文字シーケンスに変換するかどうかを指定します。現在のところ、この変換は以下の場合にのみ有効です。</p> <ol style="list-style-type: none"> 1. ユーザー定義のメタデータ・キー com.ibm.wmqfte.newRecordOnLineSeparator が false に設定されていて、転送先がレコード単位のファイルである場合。 2. ユーザー定義のメタデータ・キー com.ibm.wmqfte.com.ibm.wmqfte.insertRecordLineSeparator が false に設定されていて、転送元がレコード単位のファイルである場合。 	true

関連情報

552 ページの『表 45』

[fteCreateTransfer -md パラメーター](#)

追加の MFT エージェント構成ファイル

Managed File Transfer エージェントでは、agent.properties ファイルのほかに、いくつかの XML 構成ファイルを構成ディレクトリーで保持できます。

構成ファイル

エージェントで使用する追加情報を指定するために、以下の XML 構成ファイルを使用できます。

ProtocolBridgeCredentials.xml

プロトコル・ブリッジ・エージェントの場合は、このファイルを使用して、エージェントの接続先の FTP サーバーまたは SFTP サーバーにログインするときに使用する資格情報を指定できます。

ProtocolBridgeProperties.xml

プロトコル・ブリッジ・エージェントの場合は、このファイルを使用して、エージェントの接続先のデフォルト以外のプロトコル・ファイル・サーバーのプロパティを定義できます。

fteCreateBridgeAgent コマンドがこのファイルのデフォルト・プロトコル・ファイル・サーバーを作成してくれます。

ConnectDirectCredentials.xml

Connect:Direct ブリッジ・エージェントの場合は、このファイルを使用して、転送に関与する Connect:Direct ノードに接続するときに使用する資格情報を指定できます。

ConnectDirectNodeProperties.xml

Connect:Direct ブリッジ・エージェントの場合は、このファイルを使用して、転送に関与する Connect:Direct ノードに関するオペレーティング・システム情報を指定できます。

ConnectDirectProcessDefinition.xml

Connect:Direct ブリッジ・エージェントの場合は、このファイルを使用して、ファイル転送の一部として呼び出すユーザー定義 Connect:Direct プロセスを指定できます。

UserSandboxes.xml

このファイルを使用して、エージェントが読み取りまたは書き込みを行えるファイル・システムの領域を指定できます。

構成ファイルの更新

agent.properties ファイルの場合とは異なり、これらの XML 構成ファイルについては、更新したときにエージェントを再始動しなくても、エージェントはその変更内容を認識できます。

転送を実行するときに、エージェントが最後に XML 構成ファイルを確認した時点からの経過時間が 10 秒を超えている場合は、エージェントがその XML 構成ファイルの最終変更日時を確認します。エージェントが XML 構成ファイルを最後に読み取った時点以降にそのファイルが変更されている場合は、エージェントがそのファイルを再び読み取ります。XML スキーマに照らしてそのファイルの内容が有効であれば、エー

ジェントはその情報を更新します。ファイルの内容が無効であれば、エージェントは、旧バージョンのファイルの情報を使用し、output0.log ファイルにメッセージを書き込みます。

プロトコル・ブリッジの資格情報ファイルのフォーマット

Managed File Transfer Agent の構成ディレクトリーにある ProtocolBridgeCredentials.xml ファイルは、プロトコル・ブリッジ・エージェントがプロトコル・サーバーでの許可に使用するユーザー名および資格情報を定義します。

ProtocolBridgeCredentials.xml ファイルは ProtocolBridgeCredentials.xsd スキーマに準拠する必要があります。ProtocolBridgeCredentials.xsd スキーマ文書は、MQMFT インストール済み環境の MQ_INSTALLATION_PATH/mqft/samples/schema ディレクトリーにあります。

ProtocolBridgeCredentials.xml ファイルは、ユーザーが手動で作成する必要があります。このファイルは **fteCreateBridgeAgent** コマンドでは作成されなくなりました。MQMFT インストール済み環境の MQ_INSTALLATION_PATH/mqft/samples ディレクトリーにサンプル・ファイルがあります。

IBM WebSphere MQ 7.5 では、指定されたエージェントの <server> または <serverHost> エレメントを含む新しい <agent> エレメントが導入されました。

ProtocolBridgeCredentials.xml ファイルはエージェントが定期的に再ロードするので、このファイルに有効な変更を加えるとエージェントの動作に反映されます。デフォルトの再ロード間隔は 30 秒です。この間隔は、agent.properties ファイルの xmlConfigReloadInterval エージェント・プロパティを指定することにより変更できます。

スキーマ-7.5 以降

以下のスキーマは、IBM MQ 8 の ProtocolBridgeCredentials.xml ファイルで有効なエレメントを記述します。

```
<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>
```

ProtocolBridgeCredentials.xml ファイルについて

ProtocolBridgeCredentials.xml ファイルで使用するエレメントと属性を以下のリストにまとめます。

<credentials>

プロトコル・サーバーに接続するためにプロトコル・ブリッジ・エージェントにより使用される資格情報を記述するエレメントを含む、グループ・エレメント。

<agent>

指定されたエージェントの <server> または <serverHost> 定義が含まれているエレメント。

<server>

プロトコル・ブリッジの接続先のプロトコル・サーバー。

<server> エレメントは、7.0.4 以前ではサポートされません。

属性	説明
名前	プロトコル・サーバーの名前。
pattern	ワイルドカードまたは正規表現を使用してプロトコル・サーバー名のパターンを指定している場合は、wildcard または regex を使用します。
trustStorePassword または trustStorePasswordCipher	<server> エレメントが FTPS サーバーを参照する場合は必須です。トラストストアへのアクセスに使用するパスワード。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。
keyStorePassword または keyStorePasswordCipher	オプション。鍵ストアへのアクセスに使用するパスワード。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。

<serverHost>

プロトコル・ブリッジの接続先のプロトコル・サーバーのホスト名。

ProtocolBridgeCredentials.xml ファイルには、<serverHost> エレメントまたは <server> エレメントを含めることができますが、この2つのタイプを組み合わせて使用することはできません。<serverHost> を使用すると、名前はプロトコル・サーバーのホスト名とマッチングされます。<server> を使用すると、名前はプロトコル・サーバーの名前 (ProtocolBridgeProperties.xml ファイル内で定義されている) とマッチングされます。

属性	説明
名前	プロトコル・サーバーのホスト名または IP アドレス。
trustStorePassword または trustStorePasswordCipher	<serverHost> エレメントが FTPS サーバーを参照する場合は必須です。トラストストアへのアクセスに使用するパスワード。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。
keyStorePassword または keyStorePasswordCipher	オプション。鍵ストアへのアクセスに使用するパスワード。このプロパティは、keyStore 属性を設定していない場合はオプションで、設定している場合は必須です。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。

<user>

Managed File Transfer のユーザー名からプロトコル・サーバーのユーザー名へのユーザー・マッピング。

属性	説明
名前	Managed File Transfer で使用されるユーザー名。
serverUserId または serverUserIdCipher	プロトコル・サーバーで使用されるユーザー名。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。
serverPassword または serverPasswordCipher	プロトコル・サーバーで使用されるユーザー名のパスワード。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。
hostKey	サーバーのホストの SSH 指紋。

<privateKey>

ユーザーの秘密鍵。

属性	説明
keyPassword または keyStorePasswordCipher	秘密鍵のパスワード。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。
associationName	トレースとロギングに使用される名前。

プロトコル・ブリッジ・プロパティ・ファイルのフォーマット

エージェント構成ディレクトリー内の ProtocolBridgeProperties.xml ファイルは、プロトコル・ファイル・サーバーのプロパティを定義します。

ProtocolBridgeProperties.xml ファイルは ProtocolBridgeProperties.xsd スキーマに準拠する必要があります。 ProtocolBridgeProperties.xsd スキーマ文書は、Managed File Transfer インストール済み環境の MQ_INSTALLATION_PATH/mqft/samples/schema ディレクトリーにあります。 テンプレート・ファイル ProtocolBridgeProperties.xml は、 **fteCreateBridgeAgent** コマンドにより、エージェントの構成ディレクトリーに作成されます。

ProtocolBridgeProperties.xml ファイルはエージェントが定期的に再ロードするので、このファイルに有効な変更を加えるとエージェントの動作に反映されます。 デフォルトの再ロード間隔は 30 秒です。 この間隔は、agent.properties ファイルの xmlConfigReloadInterval エージェント・プロパティを指定することにより変更できます。

スキーマ

以下のスキーマは ProtocolBridgeProperties.xml ファイルを記述しています。

注：maxReconnectRetry 属性および reconnectWaitPeriod 属性は、IBM WebSphere MQ 7.5 または IBM WebSphere MQ File Transfer Edition 7.0.2 以降ではサポートされません。

```
<schema targetNamespace="http://wmqfte.ibm.com/ProtocolBridgeProperties" elementFormDefault="qualified"
  xmlns="https://www.w3.org/2001/XMLSchema" xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties">
<!--
  Example: ProtocolBridgeProperties.xml

  <?xml version="1.0" encoding="UTF-8"?>
  <tns:serverProperties xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties"
    xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeProperties
      ProtocolBridgeProperties.xsd">
    <tns:credentialsFile path="$HOME/ProtocolBridgeCredentials.xml"/>
    <tns:defaultServer name="myserver"/>
    <tns:ftpServer name="myserver" host="myhost.hursley.ibm.com" port="1234" platform="windows"
      timeZone="Europe/London" locale="en-GB" fileEncoding="UTF-8">
```

```

        listFormat="unix" limitedWrite="false"/>
        <tns:sftpServer name="server1" host="myhost.hursley.ibm.com" platform="windows"
          fileEncoding="UTF-8" limitedWrite="false">
          <limits maxListFileNames="10"/>
        </tns:sftpServer>
    </tns:serverProperties>
-->

<!-- Root element for the document -->
<element name="serverProperties" type="tns:serverPropertiesType"></element>

<!--
    A container for all protocol bridge server properties
-->
<complexType name="serverPropertiesType">
    <sequence>
        <element name="credentialsFile" type="tns:credentialsFileName" minOccurs="0" maxOccurs="1"/>
        <element name="defaultServer" type="tns:serverName" minOccurs="0" maxOccurs="1"/>
        <choice minOccurs="0" maxOccurs="unbounded">
            <element name="ftpServer" type="tns:ftpServerType"/>
            <element name="sftpServer" type="tns:sftpServerType"/>
            <element name="ftpsServer" type="tns:ftpsServerType"/>
            <element name="ftpsfgServer" type="tns:ftpsfgServerType"/>
            <element name="ftpsfgServer" type="tns:ftpsfgServerType"/>
        </choice>
    </sequence>
</complexType>

<!--
    A container for a server name
-->
<complexType name="serverName">
    <attribute name="name" type="tns:serverNameType" use="required"/>
</complexType>

<!--
    A container for a credentials file name
-->
<complexType name="credentialsFileName">
    <attribute name="path" type="string" use="required"/>
</complexType>

<!--
    A container for all the information about an FTP server
-->
<complexType name="ftpServerType">
    <sequence>
        <element name="limits" type="tns:generalLimitsType" minOccurs="0" maxOccurs="1" />
    </sequence>
    <attributeGroup ref="tns:ftpServerAttributes"/>
    <attribute name="passiveMode" type="boolean" use="optional"/>
</complexType>

<!--
    A container for all the information about an SFG FTP server
-->
<complexType name="ftpsfgServerType">
    <sequence>
        <element name="limits" type="tns:generalLimitsType" minOccurs="0" maxOccurs="1" />
    </sequence>
    <attributeGroup ref="tns:ftpServerAttributes"/>
</complexType>

<!--
    A container for all the information about an SFTP server
-->
<complexType name="sftpServerType">
    <sequence>
        <element name="limits" type="tns:sftpLimitsType" minOccurs="0" maxOccurs="1" />
    </sequence>
    <attributeGroup ref="tns:sftpServerAttributes"/>
</complexType>

<!--
    A container for all the information about a FTPS server
-->
<complexType name="ftpsServerType">
    <sequence>
        <element name="limits" type="tns:generalLimitsType" minOccurs="0" maxOccurs="1" />
    </sequence>
    <attributeGroup ref="tns:ftpsServerAttributes"/>
</complexType>

```

```

<!--
  A container for all the information about a SFG FTPS server
-->
<complexType name="ftpsSfgServerType">
  <sequence>
    <element name="limits" type="tns:generalLimitsType" minOccurs="0" maxOccurs="1" />
  </sequence>
  <attributeGroup ref="tns:ftpsServerAttributes"/>
</complexType>

<!--
  Attributes common to all server types
-->
<attributeGroup name="generalServerAttributes">
  <attribute name="name" type="tns:serverNameType" use="required"/>
  <attribute name="host" type="string" use="required"/>
  <attribute name="port" type="nonNegativeInteger" use="optional"/>
  <attribute name="platform" type="tns:platformType" use="required"/>
  <attribute name="fileEncoding" type="string" use="required"/>
  <attribute name="limitedWrite" type="boolean" use="optional"/>
  <attribute name="controlEncoding" type="string" use="optional"/>
</attributeGroup>

<!--
  Attributes common to ftp and ftps server types
-->
<attributeGroup name="ftpServerAttributes">
  <attributeGroup ref="tns:generalServerAttributes"/>
  <attribute name="timeZone" type="string" use="required"/>
  <attribute name="locale" type="tns:localeType" use="required"/>
  <attribute name="listFormat" type="tns:listFormatType" use="optional"/>
  <attribute name="listFileRecentDateFormat" type="tns:dateFormatType" use="optional"/>
  <attribute name="listFileOldDateFormat" type="tns:dateFormatType" use="optional"/>
  <attribute name="monthShortNames" type="tns:monthShortNamesType" use="optional"/>
</attributeGroup>

<!--
  Attributes common to ftps server types
-->
<attributeGroup name="ftpsServerAttributes">
  <attributeGroup ref="tns:ftpServerAttributes"/>
  <attribute name="ftpsType" type="tns:ftpsTypeType" use="optional"/>
  <attribute name="trustStore" type="string" use="required"/>
  <attribute name="trustStoreType" type="string" use="optional"/>
  <attribute name="keyStore" type="string" use="optional"/>
  <attribute name="keyStoreType" type="string" use="optional"/>
  <attribute name="ccc" type="boolean" use="optional"/>
  <attribute name="protFirst" type="boolean" use="optional"/>
  <attribute name="auth" type="string" use="optional"/>
  <attribute name="connectTimeout" type="nonNegativeInteger" use="optional"/>
</attributeGroup>

<!--
  A container for limit-type attributes for a server. Limit parameters
  are optional, and if not specified a system default will be used.
-->
<complexType name="generalLimitsType">
  <attributeGroup ref="tns:generalLimitAttributes"/>
</complexType>

<complexType name="sftplimitsType">
  <attributeGroup ref="tns:generalLimitAttributes"/>
  <attribute name="connectionTimeout" type="nonNegativeInteger" use="optional"/>
</complexType>

<!--
  Attributes for limits common to all server types
-->
<attributeGroup name="generalLimitAttributes">
  <attribute name="maxListFileNames" type="positiveInteger" use="optional"/>
  <attribute name="maxListDirectoryLevels" type="nonNegativeInteger" use="optional"/>
  <attribute name="maxReconnectRetry" type="nonNegativeInteger" use="optional"/>
  <attribute name="reconnectWaitPeriod" type="nonNegativeInteger" use="optional"/>
  <attribute name="maxSessions" type="positiveInteger" use="optional"/>
  <attribute name="socketTimeout" type="nonNegativeInteger" use="optional"/>
</attributeGroup>

<!--
  The type for matching valid server names. Server names must be at least 2 characters in length

```

and

```
are limited to alphanumeric characters and the following characters: ".", "_", "/" and "%".
-->
<simpleType name="serverNameType">
  <restriction base="string">
    <pattern value="[0-9a-zA-Z\._/%]{2,}" />
  </restriction>
</simpleType>

<!--
  The types of platform supported.
-->
<simpleType name="platformType">
  <restriction base="string">
  </restriction>
</simpleType>

<!--
  The type for matching a locale specification.
-->
<simpleType name="localeType">
  <restriction base="string">
    <pattern value="(.)[-_](.)/" />
  </restriction>
</simpleType>

<!--
  The types of list format supported (for FTP servers).
-->
<simpleType name="listFormatType">
  <restriction base="string">
  </restriction>
</simpleType>

<!--
  Date format for FTP client directory listing on an FTP server. This is
  the format to be passed to methods setDefaultDateFormatStr and
  setRecentDateFormatStr for Java class:
  org.apache.commons.net.ftp.FTPClientConfig
-->
<simpleType name="dateFormatType">
  <restriction base="string">
  </restriction>
</simpleType>

<!--
  A list of language-defined short month names can be specified. These are
  used for translating the directory listing received from the FTP server.
  The format is a string of three character month names separated by "|"
-->
<simpleType name="monthShortNamesType">
  <restriction base="string">
    <pattern value="(...\|){11}(...)" />
  </restriction>
</simpleType>

<!--
  The enumerations of the allowed FTPS types: "implicit" & "explicit"
  If not specified the default is "explicit"
-->
<simpleType name="ftpsTypeType">
  <restriction base="string">
    <enumeration value="explicit" />
    <enumeration value="implicit" />
  </restriction>
</simpleType>

<!--
  Attribute Group for SFTP Servers
-->
<attributeGroup name="sftpServerAttributes">
  <attributeGroup ref="tns:generalServerAttributes" />
  <attribute name="cipherList" type="string" use="optional" />
</attributeGroup>
</schema>
```

ProtocolBridgeProperties.xml ファイルについて

ProtocolBridgeProperties.xml ファイルで使用するエレメントと属性を以下のリストにまとめます。

<serverProperties>

XML 文書のルート・エレメント

<credentialsFile>

資格情報が含まれているファイルのパス。IBM WebSphere MQ 7.5 以降の場合は、このプロパティの値に環境変数を含めることができます。詳しくは、[535 ページの『MFT プロパティでの環境変数の使用』](#)を参照してください。

<defaultServer>

ファイル転送用のデフォルト・サーバーの働きをするプロトコル・ファイル・サーバー

<ftpServer>

FTP ファイル・サーバー

<sftpServer>

SFTP ファイル・サーバー

<ftpsServer>

FTPS ファイル・サーバー

すべてのタイプのプロトコル・ファイル・サーバーに適用される汎用サーバー属性:

属性	説明
名前	必須。プロトコル・ファイル・サーバーの名前。プロトコル・サーバー名は、2文字以上の長さでなければならず、大/小文字の区別はなく、英数字と以下の文字に制限されます。 <ul style="list-style-type: none">• ピリオド (.)• 下線 (_)• スラッシュ (/)• パーセント記号 (%)
host	必須。ファイルの送信先または送信元となるプロトコル・ファイル・サーバーのホスト名または IP アドレス。
port	オプション。ファイルの送信先または送信元となるプロトコル・ファイル・サーバーのポート番号。
platform	必須。ファイルの送信先または送信元となるプロトコル・ファイル・サーバーのプラットフォーム。UNIX または WINDOWS のいずれかを指定します。このプロパティは、FTP、FTPS、または SFTP サーバー上でのパスの入力方法に従って設定します。例えば、Windows で実行している FTP サーバーにログインするときに UNIX スタイルのパス (スラッシュを使用するパス) を入力しなければならない場合は、この値を WINDOWS ではなく UNIX に設定します。Windows で実行されるサーバーの多くは、UNIX スタイルのファイル・システムに対応しています。
fileEncoding	必須。ファイル・サーバーで使用されている文字エンコード方式を定義します。このプロパティは、テキスト・モードでファイルを転送する場合に使用され、プラットフォーム間をファイルが移動する際に、エンコード手順が正しく変更されるようにします。例: UTF-8。
limitedWrite	オプション。ファイル・サーバーに書き込みを行う際のデフォルト・モードでは、一時ファイルを作成し、転送が完了した後にそのファイルをリネームします。書き込み専用として構成されたファイル・サーバーの場合、ファイルは、最終的な名前をそのまま使用して作成されます。このプロパティの値は、true または false のいずれかになります。デフォルトは false です。

属性	説明
controlEncoding	オプション。プロトコル・ファイル・サーバーに送信される制御メッセージの制御エンコード値。このプロパティは、使用されるファイル名のエンコードに影響し、プロトコル・ファイル・サーバーの制御エンコードとの互換性が必要です。デフォルトは UTF-8 です。

FTP サーバーと FTPS サーバーのみに適用される汎用属性:

属性	説明
timeZone	必須。ファイルの送信先または送信元となるプロトコル・ファイル・サーバーのタイム・ゾーン。例: America/New_York または Asia/Tokyo。
ロケール (locale)	必須。ファイルの送信先または送信元となるプロトコル・ファイル・サーバーで使用される言語。例: en_US または ja_JP。
listFormat	オプション。プロトコル・ファイル・サーバーから返される情報をリストするファイルの形式を定義するリスト形式。Windows または UNIX を使用します。デフォルトは UNIX です。
listFileRecentDateFormat	オプション。FTP サーバー上の FTP クライアント・ディレクトリー・リストの最新の日付形式 (1 年未満)。この属性と listFileOldDateFormat 属性を使用することで、プロトコル・ファイル・サーバーから返される、予期される日付形式を再定義することができます。デフォルトは、プロトコル・ファイル・サーバーで定義されている値です。
listFileOldDateFormat	オプション。FTP サーバー上の FTP クライアント・ディレクトリー・リストの古い日付形式 (1 年を超えている)。この属性と listFileRecentDateFormat 属性を使用することで、プロトコル・ファイル・サーバーから返される、予期される日付形式を再定義することができます。デフォルトは、プロトコル・ファイル・サーバーで定義されている値です。
monthShortNames	オプション。プロトコル・ファイル・サーバーから返される日付情報のデコードに使用される月名の置換リスト。このプロパティは、デフォルト・ロケールの月の値をオーバーライドする、12 個のコンマ区切りの名前のリストで構成されます。デフォルトは、プロトコル・ファイル・サーバーで定義されている値です。

FTP サーバーのみに適用される汎用属性:

属性	説明
passiveMode	オプション。FTP サーバーへの接続がパッシブかアクティブかを制御します。このプロパティの値を false に設定すると、接続はアクティブになります。この値を true に設定すると、接続はパッシブになります。デフォルトは false です。

FTPS サーバーのみに適用される汎用属性:

属性	説明
ftpsType	オプション。使用される FTPS プロトコルの形式が明示的か暗黙的かを指定します。デフォルトは explicit です。
trustStore	必須。FTPS サーバーによって提示される証明書が信頼できるかどうかの判別使用するトラストストアの場所。
trustStoreType	オプション。トラストストア・ファイルの形式。デフォルトは JKS です。

属性	説明
keyStore	オプション。FTPS サーバーからユーザー確認が行われた場合に、証明書の情報を提供するために使用する鍵ストアの場所。デフォルトでは、プロトコル・ブリッジは、クライアントの認証を必要とするように構成された FTPS サーバーに接続できません。
keyStoreType	オプション。鍵ストア・ファイルの形式。デフォルトは JKS です。
ccc	オプション。認証の完了後に保護されていない (暗号化されていない) コマンド・チャンネルを使用するかどうかを選択します。デフォルト値は <code>false</code> で、この場合は FTPS セッションの期間全体でコマンド・チャンネルが暗号化されたままになります。この属性は、 <code>ftpsType</code> が <code>explicit</code> に設定されている場合のみ適用できます。
protFirst	オプション。 USER/PASS コマンドを FTPS サーバーに発行するのを PBSZ/PROT コマンドの前にするか後にするかを指定します。デフォルト値は <code>false</code> です。この場合、 USER/PASS コマンドが最初に送信され、その後に PBSZ/PROT コマンドが送信されます。この属性は、 <code>ftpsType</code> が <code>explicit</code> に設定されている場合のみ適用できます。
auth	オプション。 AUTH コマンドの一部として指定されるプロトコルを指定します。まず指定されたプロトコルを試行してから、デフォルトでは、FTPS サーバーが 504 応答コードで拒否しなくなるまで、TLS、SSL、TLS-C、または TLS-P を試行します。この属性は、 <code>ftpsType</code> が <code>explicit</code> に設定されている場合のみ適用できます。

<limits>

すべてのタイプのサーバーに共通の属性や、サーバーのタイプに固有の属性のコンテナー・エレメント。以下の属性があります。

すべてのタイプのプロトコル・ファイル・サーバーに適用する汎用制限属性:

属性	説明
maxListFileNames	オプション。プロトコル・ファイル・サーバー上のディレクトリーでファイル名をスキャンする際に収集される名前の最大数。デフォルトは 999999999 です。
maxListDirectoryLevels	オプション。プロトコル・サーバー上でファイル名を繰り返しスキャンするディレクトリー・レベルの最大数。デフォルトは 1000 です。
maxReconnectRetry (この属性は非推奨になっています。)	非推奨。この属性は、IBM WebSphere MQ 7.5 または IBM WebSphere MQ File Transfer Edition 7.0.2 以降ではサポートされません。 オプション。プロトコル・ブリッジ・エージェントが試行を停止する前に、プロトコル・サーバーが再接続を試行する最大回数。デフォルトは 2 です。
reconnectWaitPeriod (この属性は非推奨になっています。)	非推奨。この属性は、IBM WebSphere MQ 7.5 または IBM WebSphere MQ File Transfer Edition 7.0.2 以降ではサポートされません。 オプション。再接続を試行する前に待機する時間 (秒単位)。デフォルトは 10 秒です。
maxSessions	オプション。プロトコル・サーバーに関するセッションの最大数。この数値は、プロトコル・ブリッジ・エージェントのソース転送と宛先転送の最大数の合計以上でなければなりません。デフォルトは、エージェント・プロパティー <code>maxSourceTransfers</code> 、 <code>maxDestinationTransfers</code> 、および <code>maxCommandHandlerThreads</code> の値の合計に 1 を加えた値です。これらの 3 つのプロパティーがデフォルト値の 25、25、および 5 を使用する場合、 <code>maxSessions</code> のデフォルトは 56 になります。

属性	説明
socketTimeout	オプション。ソケット・タイムアウト (秒単位)。この属性の値は、ファイルのストリーミング中に使用されます。デフォルトは 30 秒です。

SFTP サーバーのみに適用される制限属性:

属性	説明
connectionTimeout	オプション。接続要求に対するプロトコル・ファイル・サーバーからの応答を待機する時間 (秒単位)。タイムアウトは、プロトコル・ファイル・サーバーが使用可能でないことを示します。デフォルト値は 30 秒です。
cipherList	<p>オプション。プロトコル・ブリッジ・エージェントと SFTP サーバーの間で通信するために使用される暗号のコンマ区切りリストを指定します。暗号は、このリストで指定された順序で呼び出されます。暗号は、使用前にサーバーとクライアントで使用可能になっていなければなりません。</p> <p>プロトコル・ブリッジ・エージェントがサポートする暗号は、以下のとおりです。</p> <ul style="list-style-type: none"> • blowfish-cbc • 3des-cbc • aes128-cbc • aes192-cbc • aes256-cbc • aes128-ctr • aes192-ctr • aes256-ctr • 3des-ctr • arcfour • arcfour128 • arcfour256 <p>デフォルトでは、プロトコル・ブリッジ・エージェントによって使用される暗号のリストは、aes128-cbc, aes192-cbc, aes256-cbc です。</p>

Connect:Direct 資格情報ファイル・フォーマット

Managed File Transfer Agent 構成ディレクトリーの ConnectDirectCredentials.xml ファイルは、Connect:Direct エージェントが Connect:Direct ノードに対して自身を許可するのに使用するユーザー名および資格情報を定義します。

ConnectDirectCredentials.xml ファイルは ConnectDirectCredentials.xsd スキーマに準拠する必要があります。ConnectDirectCredentials.xsd スキーマ文書は、MQMFT インストール済み環境の MQ_INSTALLATION_PATH/mqft/samples/schema ディレクトリーにあります。サンプルの ConnectDirectCredentials.xml ファイルは、MQMFT インストール済み環境の MQ_INSTALLATION_PATH/mqft/samples/credentials ディレクトリーにあります。

ファイル ConnectDirectCredentials.xml はエージェントによって定期的に再ロードされ、ファイルに有効な変更が加えられていれば、それはエージェントの動作に反映されます。デフォルトの再ロード間隔は 30 秒です。この間隔は、agent.properties ファイルでエージェント・プロパティ **xmlConfigReloadInterval** を指定することによって変更できます。

スキーマ

次のスキーマは、ConnectDirectCredentials.xml ファイル内で有効なエレメントについて示します。

```
<?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"/>
  </--
  </schema>
```

```

    <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>

```

ConnectDirectCredentials.xml ファイルについて

ConnectDirectCredentials.xml ファイルで使用するエレメントと属性を以下のリストにまとめます。

<credentials>

Connect:Direct ブリッジ・エージェントが Connect:Direct ノードに接続するとき使用する資格情報を記述するエレメントを含むグループ・エレメント。

<agent>

名前付きエージェントの <pnode> 定義のエレメントが含まれているグループ・エレメント。

<pnode>

Connect:Direct 転送の 1 次ノード (PNODE)。このノードが 2 次ノード (SNODE) への接続を開始します。

属性	説明
名前	Connect:Direct ノードの名前。この属性の値は、多数のノード名と一致するパターンにすることができます。
pattern	name 属性の値として使用されるパターンのタイプを指定します。pattern 属性の有効な値は以下のとおりです。 <ul style="list-style-type: none"> wildcard - ワイルドカードを使用 regex - Java 正規表現を使用

<user>

転送要求を実行依頼する IBM MQ ユーザー。

属性	説明
名前	Managed File Transfer で使用されるユーザー名。この属性の値は、多数のユーザー名と一致するパターンにすることができます。
ignorecase	名前の大/小文字を無視するかどうかを指定します。ignorecase 属性の有効な値は以下のとおりです。 <ul style="list-style-type: none"> true - 名前に大/小文字の区別がない false - 名前に大/小文字の区別がある
pattern	name 属性の値として使用されるパターンのタイプを指定します。pattern 属性の有効な値は以下のとおりです。 <ul style="list-style-type: none"> wildcard - ワイルドカードを使用 regex - Java 正規表現を使用

属性	説明
cdUserId または cdUserIdCipher	Connect:Direct ブリッジが、関連付けられている Connect:Direct ノードに接続するために使用するユーザー名。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。
cdPassword または cdPasswordCipher	cdUserId 属性によって指定されたユーザー名に関連付けられているパスワード。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。
pnodeUserId または pnodeUserIdCipher	Connect:Direct 1 次ノードで使用するユーザー名。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。
pnodePassword または pnodePasswordCipher	pnodeUserId 属性によって指定されたユーザー名に関連付けられているパスワード。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。

<snode>

Connect:Direct ファイル転送の間、2 次ノード (SNODE) の役割を果たす Connect:Direct ノード。

属性	説明
名前	Connect:Direct ノードの名前。この属性の値は、多数のノード名と一致するパターンにすることができます。
pattern	name 属性の値として使用されるパターンのタイプを指定します。pattern 属性の有効な値は以下のとおりです。 <ul style="list-style-type: none"> • wildcard - ワイルドカードを使用 • regex - Java 正規表現を使用
userId または userIdCipher	ファイル転送の間、このノードに接続するために使用するユーザー名。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。
password または passwordCipher	userId 属性によって指定されたユーザー名に関連付けられているパスワード。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。

例

この例では、Connect:Direct ブリッジ・エージェントが Connect:Direct ノード pnode1 に接続します。接頭部 fteuser で始まり、その後単一文字が続く IBM MQ ユーザー (例えば、fteuser2) が Connect:Direct ブリッジを含む転送を要求すると、Connect:Direct ブリッジ・エージェントはユーザー名 cduser とパスワード passw0rd を使用して Connect:Direct ノード pnode1 に接続します。Connect:Direct ノード pnode1 が転送における自らの役割を実行するときには、ユーザー名 pnodeuser およびパスワード passw0rd1 を使用します。

Connect:Direct 転送における 2 次ノードの名前が接頭部 FISH で始まる場合、ノード pnode1 はユーザー名 fishuser およびパスワード passw0rd2 を使用して 2 次ノードに接続します。Connect:Direct 転送における 2 次ノードの名前が接頭部 CHIPS で始まる場合、ノード pnode1 はユーザー名 chipsuser およびパスワード passw0rd3 を使用して 2 次ノードに接続します。

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:credentials xmlns:tns="http://wmqfte.ibm.com/ConnectDirectCredentials"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ConnectDirectCredentials
  ConnectDirectCredentials.xsd">
  <tns:agent name="CDAGENT01">
  <tns:pnode name="pnode1" pattern="wildcard">
```

```

<tns:user name="fteuser?" pattern="wildcard" ignorecase="true"
  cdUserId="cduser" cdPassword="passw0rd"
  pnodeUserId="pnodeuser" pnodePassword="passw0rd1">
<tns:snode name="FISH*" pattern="wildcard"
  userId="fishuser" password="passw0rd2"/>
<tns:snode name="CHIPS*" pattern="wildcard"
  userId="chipsuser" password="passw0rd3"/>
</tns:user>
</tns:pnode>
</tns:agent>
</tns:credentials>

```

Connect:Direct ノードのプロパティ・ファイルのフォーマット

Connect:Direct ブリッジ・エージェントの構成ディレクトリーにある ConnectDirectNodeProperties.xml ファイルでは、ファイル転送にかかわっているリモート Connect:Direct ノードに関する情報を指定します。

ConnectDirectNodeProperties.xml ファイルは ConnectDirectNodeProperties.xsd スキーマに準拠する必要があります。ConnectDirectNodeProperties.xsd スキーマ文書は、MFT インストール済み環境の MQ_INSTALLATION_PATH/mqft/samples/schema ディレクトリーにあります。

ConnectDirectNodeProperties.xml テンプレート・ファイルは、**fteCreateCDAgent** コマンドにより、エージェントの構成ディレクトリーに作成されます。

ファイル ConnectDirectNodeProperties.xml はエージェントによって定期的に再ロードされ、ファイルに有効な変更が加えられていれば、それはエージェントの動作に反映されます。デフォルトの再ロード間隔は 30 秒です。この間隔は、agent.properties ファイルの xmlConfigReloadInterval エージェント・プロパティを指定することにより変更できます。

スキーマ

次のスキーマは、ConnectDirectNodeProperties.xml ファイル内で有効なエレメントについて示します。

```

<?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>

```

ConnectDirectNodeProperties.xml ファイルについて

ConnectDirectNodeProperties.xml ファイルで使用するエレメントと属性を以下のリストにまとめます。

nodeProperties

XML 文書のルート・エレメント。

credentialsFile

機密情報が保管されている資格情報ファイルのパス。IBM WebSphere MQ 7.5 以降の場合は、このプロパティの値に環境変数を含めることができます。詳しくは、[535 ページの『MFT プロパティでの環境変数の使用』](#)を参照してください。

ノード

Connect:Direct ノードを 1 つ以上指定します。

属性	説明
名前	node エレメントで指定する定義を使用する Connect:Direct ノードの名前を識別するパターン。パターン・マッチングには、大/小文字の区別がありません。
pattern	name 属性で指定する変数の値と突き合わせるパターンを指定します。pattern 属性の有効な値は以下のとおりです。 <ul style="list-style-type: none">• wildcard - ワイルドカードを使用• regex - Java 正規表現を使用 MFT で使用する正規表現のタイプについては、 659 ページの『MFT が使用する正規表現』 を参照してください。
タイプ	name 属性で指定するパターンに適合する Connect:Direct ノードのオペレーティング・システムのタイプを指定します。type 属性の有効な値は以下のとおりです。 <ul style="list-style-type: none">• Windows - ノードは Windows で実行されています• UNIX - ノードは UNIX または Linux で実行されています•  z/OS、zos、os/390、os390 - ノードは z/OS で実行されています この属性の値には、大/小文字の区別がありません。

例

この例では、Connect:Direct 資格情報ファイル名を ConnectDirectCredentials.xml として指定します。サンプル・コードには、以下のプラットフォーム接続を指定します。

- 「"cdnodew"」で始まる名前の Connect:Direct ノードがすべて Windows プラットフォームで稼働している。
- "cdnodeu" で始まる名前を持つすべての Connect:Direct ノードは、UNIX プラットフォーム上で実行されます。
- 「"cdnodez"」で始まる名前の Connect:Direct ノードがすべて z/OS プラットフォームで稼働している。
- 他のすべての Connect:Direct ノードは UNIX プラットフォームで稼働している。

Connect:Direct ブリッジ・エージェントは、このファイルの先頭から末尾に向かって一致項目を検索し、最初に見つかった一致項目を使用します。

```
<?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 プロセスの定義ファイルのフォーマット

Connect:Direct ブリッジ・エージェントの構成ディレクトリーにある ConnectDirectProcessDefinitions.xml ファイルでは、ファイル転送の一部として開始するユーザー定義 Connect:Direct プロセスを指定します。

ConnectDirectProcessDefinitions.xml ファイルは ConnectDirectProcessDefinitions.xsd スキーマに準拠する必要があります。ConnectDirectProcessDefinitions.xsd スキーマ文書は、MFT インストール済み環境の MQ_INSTALLATION_PATH/mqft/samples/schema ディレクトリーにあります。ConnectDirectProcessDefinitions.xml テンプレート・ファイルは、**fteCreateCDAgent** コマンドにより、エージェントの構成ディレクトリーに作成されます。

ファイル ConnectDirectProcessDefinitions.xml はエージェントによって定期的に再ロードされ、ファイルに有効な変更が加えられていれば、それはエージェントの動作に反映されます。デフォルトの再ロード間隔は 30 秒です。この間隔は、agent.properties ファイルの xmlConfigReloadInterval エージェント・プロパティを指定することにより変更できます。

スキーマ

次のスキーマは、ConnectDirectProcessDefinitions.xml ファイル内で有効なエレメントについて示します。

```
<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>
</schema>
```

```

    </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>

```

ConnectDirectProcessDefinitions.xml ファイルについて

ConnectDirectProcessDefinitions.xml ファイルで使用する要素と属性を以下のリストにまとめます。

cdProcess

XML 文書のルート・要素。

processSet

一連のユーザー定義プロセスに関するすべての情報を格納したグループ・要素。

condition

processSet 要素に含まれているプロセス・セットを使用するかどうかを決定するための転送の検査条件を格納したグループ・要素。

match

変数の値が特定の値に合致するかどうかを検査するための条件。

属性	説明
variable	変数を指定します。この変数の値が value 属性の値と比較されます。この変数は、組み込みシンボルです。詳しくは、 660 ページの『ユーザー定義 Connect:Direct プロセスで使用する置換変数』 を参照してください。
値	variable 属性で指定する変数の値と突き合わせるパターンを指定します。
pattern	value 属性で指定する変数の値と突き合わせるパターンを指定します。pattern 属性の有効な値は以下のとおりです。 <ul style="list-style-type: none"> wildcard - ワイルドカードを使用 regex - Java 正規表現を使用 この属性は任意指定であり、デフォルトは wildcard です。

defined

変数が定義されているかどうかを検査するための条件。

属性	説明
variable	変数を指定します。この変数が存在すると、この一致条件は満たされたこととなります。この変数は、組み込みシンボルです。詳しくは、 660 ページの『ユーザー定義 Connect:Direct プロセスで使用する置換変数』 を参照してください。

process

一致項目が見つかったときに呼び出す Connect:Direct プロセスの場所に関する情報を格納したグループ・要素。

transfer

転送要求の実行中に呼び出す Connect:Direct プロセス。

属性	説明
process	オプション。転送要求の実行中に呼び出す Connect:Direct プロセスを格納したファイルの名前を指定します。ファイル・パスは、Connect:Direct ブリッジ・エージェントの構成ディレクトリーを基準にした相対パスです。この属性は任意指定であり、デフォルトは、MFT によって生成されたプロセスを使用する、という動作です。IBM WebSphere MQ 7.5 以降の場合は、このプロパティーの値に環境変数を含めることができます。詳しくは、535 ページの『MFT プロパティーでの環境変数の使用』を参照してください。

例

この例では、3 つの processSet エlementがあります。

最初の processSet Elementは、転送要求の **%FTESNODE** 変数の値がパターン Client* と一致し、**%FTESUSER** 変数の値が Admin である場合に、Connect:Direct ブリッジ・エージェントが *agent_configuration_directory/AdminClient.cdp* にある Connect:Direct プロセスを転送の一部として実行依頼することを指定します。

2 番目の processSet Elementは、転送要求にパターン Client* と一致する値を持つ **%FTESNODE** 変数がある場合に、Connect:Direct ブリッジ・エージェントが *agent_configuration_directory/Client.cdp* にある Connect:Direct プロセスを転送の一部として実行依頼することを指定します。Connect:Direct ブリッジ・エージェントは、各 processSet Elementを定義順に読み取り、一致項目が見つかったら、最初の一致項目を使用し、その他の一致項目は検索しません。1 番目と 2 番目の両方の processSet の条件に合致する転送要求の場合、Connect:Direct ブリッジ・エージェントは、1 番目の processSet で指定されているプロセスだけ呼び出します。

3 番目の processSet Elementは、条件がないので、すべての転送に合致します。1 番目と 2 番目の processSet の条件に合致しない転送要求の場合、Connect:Direct ブリッジ・エージェントは、3 番目の条件で指定されている Connect:Direct プロセスを実行します。転送の一部として呼び出されるこのプロセスは、*agent_configuration_directory/Default.cdp* にあります。

```
<?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>
```

MFT ユーザー・サンドボックスの処理

ファイルの転送先および転送元とすることが可能なファイル・システム内の領域を、転送を要求する MQMD ユーザー名に基づいて制限することができます。

ユーザー・サンドボックスは、エージェントがプロトコル・ブリッジ・エージェント または Connect:Direct ブリッジ・エージェントである場合はサポートされません。

ユーザー・サンドボックスを有効にするには、制限を設定するエージェントの `agent.properties` ファイルに、次のプロパティを追加します。

```
userSandboxes=true
```

このプロパティが存在し `true` に設定されている場合、エージェントは `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name/UserSandboxes.xml` ファイル内の情報を使用して、転送を要求したユーザーがファイル・システムのどの部分にアクセス可能であるかを判別します。

`UserSandboxes.xml` XML は、`<sandbox>` エlementを0個以上含んでいる1つの`<agent>` Elementで構成されています。これらのElementは、どの規則がどのユーザーに適用されるかを記述します。`<sandbox>` Elementの `user` 属性は、要求のMQMDユーザーと突き合わせるために使用されるパターンです。

ファイル `UserSandboxes.xml` はエージェントによって定期的に再ロードされ、ファイルに有効な変更が加えられていれば、それはエージェントの動作に反映されます。デフォルトの再ロード間隔は30秒です。この間隔は、`agent.properties` ファイルの `xmlConfigReloadInterval` エージェント・プロパティを指定することにより変更できます。

`userPattern="regex"` 属性または値を指定する場合、`user` 属性はJava正規表現として解釈されます。詳しくは、[659 ページの『MFTが使用する正規表現』](#)を参照してください。

`userPattern="regex"` 属性も値も指定しない場合、`user` 属性は次のワイルドカード文字を持つパターンとして解釈されます。

- アスタリスク。0個以上の文字を表します。
- 疑問符(?)。ちょうど1文字を表します。

マッチングは、ファイル内で`<sandbox>` Elementがリストされている順序で実行されます。最初のマッチングのみが使用され、ファイル内にあるかもしれないそれ以降の他のマッチングはすべて無視されます。ファイルで指定された`<sandbox>` Elementが、転送要求メッセージに関連付けられたMQMDユーザーとマッチングしない場合、その転送はファイル・システムにアクセスできません。MQMDユーザー名と`user` 属性の間にマッチングが検出された場合、そのマッチング項目を基にして、転送に適用される規則セットが`<sandbox>` Element内で識別されます。この規則セットを使用して、転送の一環として読み取りまたは書き込みが可能であるファイルやデータ・セットが判別されます。

規則セットごとに、読み取り可能なファイルを識別する`<read>` Elementおよび書き込み可能なファイルを識別する`<write>` Elementを指定できます。規則セットから`<read>` または`<write>` Elementを省略した場合、その規則セットに関連付けられたユーザーは、それぞれ読み取りまたは書き込みの実行を許可されないと想定されます。

注: `UserSandboxes.xml` ファイルの中で、`<read>` Elementは`<write>` Elementよりも前に、`<include>` Elementは`<exclude>` Elementよりも前に配置する必要があります。

`<read>` または`<write>` のそれぞれのElementには、ファイルがサンドボックス内にあるかどうか、転送可能であるかどうかを決定するために使用されるパターンが1つ以上含まれています。これらのパターンは、`<include>` および`<exclude>` Elementを使用して指定します。`<include>` または`<exclude>` Elementの `name` 属性は、突き合わせ対象となるパターンを指定します。オプションの `type` 属性は、名前値がファイルまたはキュー・パターンであるかを指定します。`type` 属性が指定されない場合、エージェントはパターンをファイルまたはディレクトリー・パス・パターンとして扱います。以下に例を示します。

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

エージェントは <include> および <exclude> name パターンを使用して、ファイル、データセット、またはキューの読み取りまたは書き込みが可能であるかどうかを判別します。操作は、正規のファイル・パス、データ・セット、またはキュー名が、包含パターンの少なくとも1つにマッチングし、除外パターンに1つもマッチングしない場合にのみ許可されます。 <include> および <exclude> エレメントの name 属性を使用して指定するパターンには、エージェントを実行しているプラットフォームに適したパス分離文字および規則が使用されます。相対ファイル・パスを指定する場合、パスはエージェントの transferRoot プロパティを基準にして相対的に解決されます。

キューの制限を指定する場合、QUEUE@QUEUEMANAGER の構文がサポートされ、以下のルールが使用されます。

- アットマーク文字 (@) が項目から欠落している場合、パターンはいずれかのキュー・マネージャーでアクセスされるキュー名として扱われます。例えば、パターンが name である場合、name@** と同じように扱われます。
- アットマーク文字 (@) が項目の最初の文字である場合、パターンはキュー・マネージャー名として扱われ、キュー・マネージャーにあるすべてのキューにアクセスできます。例えば、パターンが @name である場合、**@name と同じように扱われます。

以下のワイルドカード文字は、<include> および <exclude> エレメントの name 属性の一部として指定した場合、特別な意味を持ちます。

単一のアスタリスクは、ディレクトリー名、またはデータ・セット名またはキュー名の修飾子の中の0個以上の文字と一致します。

?

疑問符 (?) は、ディレクトリー名、またはデータ・セット名かキュー名の修飾子の中の1文字にのみ一致します。

2つのアスタリスク文字は、ゼロ個以上のディレクトリー名、またはデータ・セット名またはキュー名のゼロ個以上の修飾子に一致します。また、パス分離文字で終わるパスには、パスの終わりに暗黙の "*" が追加されています。したがって、/home/user/ は /home/user/** と同じになります。

以下に例を示します。

- /**/test/** は、パス中に test ディレクトリーが含まれるすべてのファイルにマッチングします。
- /test/file? は、/test ディレクトリー内の、ストリング file で始まり任意の1文字が続くすべてのファイルにマッチングします。
- c:\test*.txt は、c:\test ディレクトリー内のすべてのファイルを .txt 拡張子で一致させます。
- c:\test***.txt は、'c:\test ディレクトリー内の任意のファイル、または .txt 拡張子を持つサブディレクトリーの1つに一致します。
-  // 'TEST.*.DATA' は、最初の修飾子が TEST で、任意の2番目の修飾子があり、さらに3番目の修飾子が DATA であるすべてのデータ・セットにマッチングします。
- *@QM1 は、単一修飾子を持つキュー・マネージャー QM1 上のすべてのキューと一致します。
- TEST.*.QUEUE@QM1 は、最初の修飾子が TEST で、任意の2番目の修飾子があり、さらに3番目の修飾子が QUEUE である、キュー・マネージャー QM1 にあるすべてのキューにマッチングします。
- **@QM1 は、キュー・マネージャー QM1 上の任意のキューと一致します。

シンボリック・リンク

UserSandboxes.xml ファイル内のファイル・パスで使用するシンボリック・リンクは、<include> および <exclude> エレメント内でハード・リンクを指定して、完全に解決する必要があります。例えば、/var が /SYSTEM/var にマップするシンボリック・リンクがある場合は、このパスを <tns:include name="/SYSTEM/var"/>として指定する必要があります。そうしないと、意図した転送はユーザー・サンドボックス・セキュリティ・エラーで失敗します。

例

この例は、以下の <sandbox> エレメントを AGENT_JUPITER の構成ディレクトリー内のファイル UserSandboxes.xml に追加することにより、MQMD ユーザー名 guest を持つユーザーが、エージェント AGENT_JUPITER が実行されているシステム上の /home/user/public ディレクトリーまたはそのサブディレクトリーから任意のファイルを転送できるようにする方法を示しています。

```
<?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>
```

例

次の例は、account に 1 つの数字が続く MQMD ユーザー名を持つユーザー (例: account4) に、以下のアクションの実行を許可する方法を示しています。

- /home/account ディレクトリーまたはそのいずれかのサブディレクトリー (エージェント AGENT_SATURN を実行しているシステム上の /home/account/private ディレクトリーを除く) から任意のファイルを転送します。
- エージェント AGENT_SATURN を実行しているシステム上の /home/account/output ディレクトリーまたはその任意のサブディレクトリーに、任意のファイルを転送します。
- 接頭部 ACCOUNT. で始まる、ローカル・キュー・マネージャーにあるキューからのメッセージを読み取ります。ただし、ACCOUNT.PRIVATE. で始まるものは除きます (つまり第 2 レベルに PRIVATE があるものです)。
- 任意のキュー・マネージャーにある接頭部 ACCOUNT.OUTPUT. で始まるキューにデータを転送します。

MQMD ユーザー名 account のユーザーがこれらのアクションを実行できるようにするには、AGENT_SATURN の構成ディレクトリーにあるファイル UserSandboxes.xml に以下の <sandbox> エレメントを追加します。

```
<?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>
```

関連資料

[50 ページの『ワイルドカード転送の追加検査』](#)

Continuous Delivery の IBM MQ 9.0.1 から、および Long Term Support の IBM MQ 9.0.0 Fix Pack 1 からは、エージェントがファイルを転送できる場所を制限するためにユーザー・サンドボックスまたはエージェン

ト・サンドボックスを使用してエージェントが構成されている場合、そのエージェントのワイルドカード転送に対して追加の検査を行うように指定できます。

47 ページの『MFT ユーザー・サンドボックスの処理』

ファイルの転送先および転送元とすることが可能なファイル・システム内の領域を、転送を要求する MQMD ユーザー名に基づいて制限することができます。

550 ページの『MFT agent.properties ファイル』

各 Managed File Transfer Agent には、独自のプロパティ・ファイル `agent.properties` があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。`agent.properties` ファイルには、エージェントの動作を変更するプロパティを含めることもできます。

MFT ロガーの構成プロパティ

Managed File Transfer ロガーには、一連の構成プロパティがあります。`logger.properties` ファイルでこれらのプロパティを指定します。このファイルは、`MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` ディレクトリーにあります。

IBM WebSphere MQ 7.5 以降の場合は、ファイルまたはディレクトリーの場所を表す Managed File Transfer の一部のプロパティで環境変数を使用できます。これにより、製品の一部の実行時に使用されるファイルまたはディレクトリーの場所を、環境の変更(プロセスを実行しているユーザーなど)に合わせて変えることができます。詳しくは、[535 ページの『MFT プロパティでの環境変数の使用』](#)を参照してください。

注: Windows 上でファイル・パスを指定する場合、円記号 (¥) 分離文字は、二重の円記号 (¥¥) (つまり、エスケープ・シーケンスの ¥) で表す必要があります。代わりに、分離文字として単一スラッシュ文字 (/) を使用できます。Oracle の Java プロパティ・ファイルでの文字のエスケープ・シーケンスについて詳しくは、[Javadoc for the Properties class](#) を参照してください。

バインディング・モード接続のプロパティ

表 47. バインディング・モードの接続プロパティ		
プロパティ名	説明	デフォルト値
<code>wmqfte.logger.type</code>	使用しているロガー・タイプ (file または database)。この値は FILE または DATABASE に設定してください。	デフォルト値なし
<code>wmqfte.max.transaction.messages</code>	トランザクションがコミットされる前にトランザクションで処理されるメッセージの最大数。循環ロギング・モードでは、未完了データに使用可能な固定スペース量はキュー・マネージャーが持っています。使用可能なスペースを使い尽くさないよう十分に低い値をこのプロパティに設定するようにしてください。	50
<code>wmqfte.max.transaction.time</code>	トランザクションのコミット間に経過する時間の最大長 (ミリ秒)。	5000
<code>wmqfte.max.consecutive.reject</code>	連続して (つまり、有効メッセージが出現することなく) リジェクト可能なメッセージの最大数。 この数値を超えると、ロガーは問題がメッセージ自体にあるのではなく構成にあると見なします。例えば、使用しているすべてのエージェント名よりデータベース内のエージェント名列を狭く作ると、エージェントを参照しているすべてのメッセージがリジェクトされます。	50
<code>wmqfte.reject.queue.name</code>	ロガーがメッセージを処理できない場合にロガーがメッセージを置くキューの名前。データベース・ロガーを使用している場合、このキューに置かれる可能性のあるメッセージについて詳しくは、 データベース・ロガーのエラー処理とリジェクト を参照してください。	SYSTEM.FTE.LOG.RJCT. <code>logger_name</code>

表 47. バインディング・モードの接続プロパティ (続き)		
プロパティ名	説明	デフォルト値
wmqfte.command.queue.name	ローガーがその動作を制御するコマンド・メッセージを読み取る際に、その読み取り元となるキューの名前。	SYSTEM.FTE.LOG.CMD.logger_name
wmqfte.queue.manager	ローガーの接続先キュー・マネージャー。このパラメーターは必須で、キュー・マネージャーへのバインディング・モード接続に必要となるものはこのパラメーターのみです。(リモート・キュー・マネージャーに接続するためのプロパティについては、 603 ページの表 48 を参照してください。)	デフォルト値なし
wmqfte.message.source.type	次のいずれかの値。 automatic subscription デフォルト値。ローガーは、SYSTEM.FTE/Log/# で定義される独自の永続的な管理対象サブスクリプションをキュー・マネージャー上に作成して使用します。これは、ほとんどのシナリオで適切な値となります。 administrative subscription 自動サブスクリプションが適切でない場合は、(例えば、IBM MQ Explorer、MQSC、または PCF を使用して) 異なるサブスクリプションを定義し、ローガーにそのサブスクリプションを使用するよう命令できます。例えば、この値を使用してログ・スペースを分割し、1 番目のローガーは A から H のエージェントを、2 番目のローガーは I から P のエージェントを、3 番目のローガーは Q から Z のエージェントを処理するようにします。 queue IBM MQ トポロジーを使用するとローガーのサブスクリプションを作成することが不都合であることがわかった場合は、代わりにキューを使用できます。IBM MQ を構成して、通常は調整キュー・マネージャー上の SYSTEM.FTE/Log/# へのサブスクリプションによって受信されるメッセージをキューが受信するようにします。	automatic subscription
wmqfte.message.source.name	メッセージ送信元タイプが administrative subscription または queue である場合に使用するサブスクリプションまたはキューの名前。このプロパティは、送信元タイプが automatic subscription である場合は無視されます。	デフォルト値なし
wmqfte.database.credentials.file	データベースに接続するためのユーザー名とパスワードを格納するファイル。 IBM WebSphere MQ 7.5 以降の場合は、このプロパティの値に環境変数を含めることができます。 詳しくは、 810 ページの『MFT 資格情報ファイル・フォーマット』 を参照してください。	 認証資格情報ファイルの作成に関する詳細は、 MFT 資格情報ファイルの作成を参照してください  このファイルの場所およびアクセス権については、 MQMFTCredentials.xml の構成を参照してください 。  認証資格情報ファイルの作成について詳しくは、 MFT と IBM MQ の接続認証を参照してください 。

表 47. バインディング・モードの接続プロパティ (続き)

プロパティ名	説明	デフォルト値
<p>wmqfte.database.driver</p>	<p>データベースの JDBC ドライバー・クラスの場合。これは通常、JAR ファイルのパスおよびファイル名です。例えば、AIX システム上の Db2 用の Type 2 ドライバーは、ファイル /opt/IBM/db2/V9.5/java/db2jcc.jar を必要とします。Windows システムでは、パス分離文字をスラッシュ文字 (/) として指定します (例: C:/Program Files/IBM/SQLLIB/java/db2jcc.jar)。</p> <p>z/OS z/OS の場合は、db2jcc.jar ファイルの絶対パスを指定します。例えば、wmqfte.database.driver=/db2/db2v10/jdbc/classes/db2jcc.jar などです。</p> <p>z/OS z/OS システムの場合、以下のすべての JAR ファイルを参照する必要があります。</p> <ul style="list-style-type: none"> • db2jcc.jar • db2jcc_license_cisuz.jar • db2jcc_javax.jar <p>データベース・ドライバーが複数の JAR ファイルで構成されている場合 (例えば、Db2 9.1 の場合は、ドライバー JAR ファイルとライセンス JAR ファイルが必要)、これらの JAR ファイルをすべてこのプロパティに含めます。複数のファイル名は、ご使用のプラットフォームのクラスパス分離文字、つまり、Windows システムの場合はセミコロン文字 (;)、他のプラットフォームの場合はコロン文字 (:) を使用して分離します。</p>	<p>デフォルト値なし</p>
<p>wmqfte.database.exclude .duplicate.metadata</p>	<p>データベース・ロガー・スキーマ内の他の表にある情報を含むメタデータ表にエントリを保管するかどうかを制御します。この値は true または false に設定してください。これらのメタデータ・エントリは、既存のデータと重複しており、データベース・ストレージ容量を無駄に占有することになるため、デフォルトでは保管されません。同じデータが出現するプロパティ・エントリおよび表は以下のとおりです。</p> <ul style="list-style-type: none"> • com.ibm.wmqfte.SourceAgent TRANSFER_EVENT または CALL_REQUEST • com.ibm.wmqfte.DestinationAgent TRANSFER_EVENT • com.ibm.wmqfte.MqmdUser TRANSFER_EVENT または CALL_REQUEST • com.ibm.wmqfte.OriginatingUser TRANSFER_EVENT または CALL_REQUEST • com.ibm.wmqfte.OriginatingHost TRANSFER_EVENT または CALL_REQUEST • com.ibm.wmqfte.TransferId TRANSFER または CALL_REQUEST • com.ibm.wmqfte.JobName TRANSFER または CALL_REQUEST <p>このプロパティの値を false に設定すると、これらのメタデータ・エントリがメタデータ表に保管されます。</p>	<p>true</p>

表 47. バインディング・モードの接続プロパティ (続き)

プロパティ名	説明	デフォルト値
wmqfte.database.host	<p>Db2 のみ:</p> <p>IBM WebSphere MQ 7.5 以降では、Type 4 JDBC ドライバーを使用して接続するデータベース・サーバーのホスト名。このプロパティの値を指定する場合は、<code>wmqfte.database.port</code> の値も指定する必要があります。両方のプロパティが定義されていない場合は、データベース・ロガーがデフォルトの Type 2 JDBC ドライバーを使用して接続されます。</p> <p>このプロパティに値を指定する場合は、データベースに接続するユーザー名とパスワードを定義するために、このロガー用の資格情報ファイル (<code>wmqfte.database.credentials.file</code> プロパティでファイル・パスを定義) が存在しており、アクセス可能である必要があります。資格情報ファイルは、データベースがローカル・システム上にある場合でも必要です。</p>	デフォルト値なし
wmqfte.database.name	Managed File Transfer ログ表を含むデータベース・インスタンス (または、Db2 for z/OS を使用する場合はサブシステム) の名前。	デフォルト値なし
wmqfte.database.type	使用中のデータベース管理システム。Db2 または Oracle。この値を <code>db2</code> または <code>oracle</code> に設定してください。	db2
wmqfte.database.port	<p>Db2 のみ:</p> <p>IBM WebSphere MQ 7.5 以降では、Type 4 JDBC ドライバーを使用して接続するデータベース・サーバーのポート番号。このプロパティの値を指定する場合は、<code>wmqfte.database.host</code> の値も指定する必要があります。両方のプロパティが定義されていない場合は、データベース・ロガーがデフォルトの Type 2 JDBC ドライバーを使用して接続されます。</p> <p>このプロパティに値を指定する場合は、データベースに接続するユーザー名とパスワードを定義するために、このロガー用の資格情報ファイル (<code>wmqfte.database.credentials.file</code> プロパティでファイル・パスを定義) が存在しており、アクセス可能である必要があります。資格情報ファイルは、データベースがローカル・システム上にある場合でも必要です。</p>	デフォルト値なし
wmqfte.database.schema	Managed File Transfer ログイン表を含むデータベース・スキーマ。ほとんどの場合、デフォルト値が適切ですが、サイト固有のデータベースに関する考慮事項に応じて代替値を指定する必要がある場合があります。	FTELOG
wmqfte.database.native.library.path	<p>選択したデータベース・ドライバーが必要とするネイティブ・ライブラリーを含むパス (該当する場合)。例えば、AIX システム上の Db2 用の Type 2 ドライバーは、<code>/opt/IBM/db2/V9.5/lib32/</code> からのライブラリーを必要とします。このプロパティの代わりに、他の方法を使用して <code>java.library.path</code> システム・プロパティを設定できます。</p> <p>Solaris および HP-UX システムの場合、fteStartLogger コマンドを実行する前に、パスを含めるように <code>LD_LIBRARY_PATH</code> 環境変数を設定し、エクスポートする必要もあります。</p>	デフォルト値なし
wmqfte.file.logger.fileDirectory	ファイル・ロガーのログ・ファイルを配置するディレクトリー。	<code>mqft/logs/coordination_dir/loggers/logger_name/logs</code>

表 47. バインディング・モードの接続プロパティ (続き)

プロパティ名	説明	デフォルト値
wmqfte.file.logger.fileSize	ログ・ファイルが大きくなることが許可される最大サイズ。サイズの値は、ゼロより大きい正整数で、単位 KB、MB、GB、m (分)、h (時)、d (日)、w (週) のいずれか 1 つを後に付けます。例えば、 <code>wmqfte.file.logger.fileSize=5MB</code> のように指定します。最大ファイル・サイズは 5MB です。 <code>wmqfte.file.logger.fileSize=2d</code> ファイルの最大サイズを 2 日分のデータに指定します。	10MB
wmqfte.file.logger.fileCount	作成するログ・ファイルの最大数。データ量がこの数のファイルに保管できる最大量を超えた場合、最も古いファイルが削除されます。そのため、ファイル数が指定された値より大きくなることはありません。	3

表 47. バインディング・モードの接続プロパティー (続き)

プロパティー名	説明	デフォルト値
wmqfte.file.logger.mode	<p>使用しているローガー・モード (circular または linear)。この値は CIRCULAR または LINEAR に設定してください。</p> <p>CIRCULAR - ファイル・ローガーは、wmqfte.file.logger.fileSize プロパティーを使用して定義される最大サイズにファイルが到達するまで、そのファイルに情報を書き込みます。最大サイズに到達すると、ファイル・ローガーは新規ファイルを開始します。このモードで書き込まれるファイルの最大数は、wmqfte.file.logger.fileCount プロパティーを使用して定義される値によって制御されます。このファイル最大数に到達すると、ファイル・ローガーは最初のファイルを削除して再作成し、それを現行のアクティブ・ファイルとして使用します。wmqfte.file.logger.fileSize プロパティーで定義された値が固定サイズ・バイト単位 (例えば、KB、MB、GB) である場合、このモードで使用されるディスク・スペースの上限は、fileSize に fileCount を乗算した値です。wmqfte.file.logger.fileSize プロパティーで定義された値が時間単位 (例えば、m、h、d、w) である場合、最大サイズは、ご使用のシステムにおけるその時間内のログ・メッセージのスループットによって決まります。このモードで実行される場合に使用されるログ・ファイル命名規則は、<code>logger_namenumber-timestamp.log</code> です。それぞれの説明は以下のとおりです。</p> <ul style="list-style-type: none"> • <code>logger_name</code> は、fteCreateLogger コマンドでローガーに付けられた名前です。 • <code>number</code> は、セット内のファイルの番号です。 • <code>timestamp</code> は、ファイルが作成されたときのタイム・スタンプです。 <p>LOGGER1-20111216123430147.log に例を示します</p> <p>LINEAR - ファイル・ローガーは、wmqfte.file.logger.fileSize プロパティーを使用して定義される最大サイズにファイルが到達するまで、そのファイルに情報を書き込みます。最大サイズに到達すると、ファイル・ローガーは新規ファイルを開始します。以前に書き込まれたファイルは削除されないため、それらをログ・メッセージの履歴レコードとして保持できます。linear モードで実行している場合、ファイルは削除されず、作成可能なファイル数の上限もないため、wmqfte.file.logger.fileCount プロパティーは無視されます。このモードで実行している場合には上限がないため、ディスク・スペースが少なくならないように、ログ・ファイルによって使用されているディスク・スペースの量を追跡する必要があります。このモードで実行される場合に使用されるログ・ファイル命名規則は、<code>logger_name-timestamp.log</code> です。それぞれの説明は以下のとおりです。</p> <ul style="list-style-type: none"> • <code>logger_name</code> は、fteCreateLogger コマンドでローガーに付けられた名前です。 • <code>timestamp</code> は、ファイルが作成されたときのタイム・スタンプです。 <p>LOGGER-20111216123430147.log に例を示します</p>	デフォルト値なし

表 47. バインディング・モードの接続プロパティ (続き)

プロパティ名	説明	デフォルト値
wmqfte.max.retry.interval	<p>ロガーが永続エラーを検出した場合の、再試行間の最大時間 (ミリ秒)。</p> <p>エラー条件によっては (例えば、データベース接続が失われた場合)、ロガーが続行できなくなります。このタイプの条件が発生した場合、ロガーは現行トランザクションをロールバックし、ある一定の時間待機した後、再試行します。ロガーの待機時間は最初非常に短いため、一時的なエラーであればすぐに克服できます。しかし、ロガーの再試行のたびに、待機時間は増大します。これにより、エラー条件がより長く続く場合、例えば、データベースが保守のためのサービス停止になっている場合に、あまりに多くの不要な作業が発生しないようになっています。</p> <p>このプロパティを使用して待機時間の長さに制限を設けると、エラー条件が解決される妥当な時間で再試行が発生します。</p>	600
loggerQMgrRetryInterval	ロガーのプロセス・コントローラーがキュー・マネージャーの使用可否を検査する間隔 (秒単位)。	30
maxRestartCount	maxRestartInterval プロパティの値によって指定された時間間隔内に行える再開の最大回数。この値を超えると、ロガーのプロセス・コントローラーはロガーの再開を中止し、その代わりに maxRestartDelay プロパティの値に基づいたアクションを実行します。	4
maxRestartInterval	ロガーのプロセス・コントローラーがロガーの再開数を測定する間隔 (秒単位)。この間隔内に行われた再開の回数が maxRestartCount プロパティの値を超えると、ロガーのプロセス・コントローラーはそのロガーの再開を中止します。ロガーのプロセス・コントローラーは、その代わりに maxRestartDelay プロパティの値に基づいたアクションを実行します。	120
maxRestartDelay	ロガーの再開率が maxRestartCount プロパティおよび maxRestartInterval プロパティの値を超過した場合にロガーのプロセス・コントローラーがとる動作を決定します。ゼロ以下の値を指定すると、ロガーのプロセス・コントローラーは停止します。ゼロより大きな値を指定した場合、この秒数だけ待機した後、ロガーのプロセス・コントローラーが保持する再開履歴情報がリセットされ、ロガーが再開されます。	-1
wmqfte.oracle.port	ロガーが Oracle インスタンスへの接続に使用するポート。このポートを TNS リスナーとも言います。	1521
wmqfte.oracle.host	ロガーが Oracle インスタンスへの接続に使用するホスト。	localhost
armELEMTYPE	オプションのプロパティ。ロガーを自動リスタート・マネージャー (ARM) で再始動するように構成する場合は、関連付けられた ARM ポリシー内で指定された ARM ELEMTYPE パラメーター値をこのプロパティに設定します。ロガーの場合、ELEMTYPE は SYSBFGLG に設定します。	設定なし
armELEMENT	オプションのプロパティ。ロガーを自動リスタート・マネージャー (ARM) で再始動するように構成する場合は、関連付けられた ARM ポリシー内で指定された ARM ELEMENT パラメーター値をこのプロパティに設定します。ELEMENT 値には、ロガー名に対応する値を設定できます。	設定なし

プロパティ名	説明	デフォルト値
loggerQMgrAuthenticationCredentialsFile	ロガーの調整キュー・マネージャーに接続するためのMQ 接続資格情報が入ったファイルへのパス。	<p>Z/OS 認証資格情報ファイルの作成に関する詳細は、MFT 資格情報ファイルの作成を参照してください</p> <p>ULW このファイルの場所およびアクセス権については、MQMFTCredentials.xml の構成を参照してください。</p> <p>ULW 認証資格情報ファイルの作成について詳しくは、MFT と IBM MQ の接続認証を参照してください。</p>
トレース	<p>オプションのプロパティ。ロガーの開始時にトレースを有効にしてロガーを実行するトレース仕様。トレース仕様は、クラス、等号、およびトレース・レベルのコンマ区切りのリストです。</p> <p>例えば、<code>com.ibm.wmqfte.databaselogger</code>、<code>com.ibm.wmqfte.databaselogger.operation=all</code> です。</p> <p>コロン区切りのリストで、複数のトレース仕様を指定できます。例えば、<code>com.ibm.wmqfte.databaselogger=moderate:com.ibm.wmqfte.databaselogger.operation=all</code></p>	なし
traceFiles	オプションのプロパティ。保持するトレース・ファイルの合計数。この値は、ロガーのプロセス・コントローラーに加え、ロガー自体にも適用されます。	5
traceSize	オプションのプロパティ。トレースが次のファイルで折り返す前の、各トレース・ファイルの最大サイズ (MB 単位)。この値は、ロガーのプロセス・コントローラーとロガー自体に適用されます。	20

クライアント・モード接続のプロパティ

V 9.0.4

次に表に、ロガー・キュー・マネージャーへのクライアント・モード接続をサポートするために必要な追加のロガー・プロパティを示します。

プロパティ名	説明	デフォルト値
wmqfte.queue.manager.host	ロガー・キュー・マネージャーのホスト名または IP アドレス。	デフォルト値なし
wmqfte.queue.manager.port	ロガー・キュー・マネージャーが listen するポート。	1414
wmqfte.queue.manager.channel	ロガー・キュー・マネージャーのサーバー接続チャンネルの名前。	SYSTEM.DEF.SVRCONN
wmqfte.Ssl.CipherSuite	<p>ロガーとロガー・キュー・マネージャーのデータ交換方法に関する TLS の動作を指定します。</p> <p>wmqfte.Ssl.CipherSuite の値は、CipherSuite 名です。この CipherSuite 名は、ロガー・キュー・マネージャー・チャンネルで使用される CipherSpec 名にマップされます。</p> <p>詳しくは、CipherSuite と CipherSpec の名前のマッピングを参照してください。</p>	デフォルト値なし

表 48. クライアント・モードの接続プロパティ (続き)		
プロパティ名	説明	デフォルト値
wmqfte.Ssl.PeerName	ロガー・キュー・マネージャーによって提供される名前と合致しなければならない識別名の骨組みを指定します。その識別名に基づいて、接続時にキュー・マネージャーによって提示される識別用証明書が検査されます。	デフォルト値なし
wmqfte.Ssl.TrustStore	ロガーが信頼する証明書の場所を指定します。 wmqfte.Ssl.TrustStore の値は、ファイル・パスです。ファイル・パスが Windows のファイル・パスの場合、円記号 (¥) はさらに円記号を追加してエスケープ (¥¥) する必要があります。このプロパティの値には環境変数を含めることができる点にご注意ください。	デフォルト値なし
wmqfte.Ssl.TrustStoreCredentialsFile	wmqfte.Ssl.TrustStore 資格情報が含まれているファイルのパス。 このプロパティの値には環境変数を含めることができる点にご注意ください。	デフォルト値なし
wmqfte.Ssl.TrustStoreType	使用する SSL 鍵ストアのタイプ。JKS および PKCS#12 の鍵ストアがサポートされています。このプロパティの値は、jks または pkcs12 のいずれかになります。	jks
wmqfte.Ssl.KeyStore	ロガーの秘密鍵の場所を指定します。 wmqfte.Ssl.KeyStore の値はファイル・パスです。ファイル・パスが Windows のファイル・パスの場合、円記号 (¥) はさらに円記号を追加してエスケープ (¥¥) する必要があります。このプロパティの値には環境変数を含めることができる点にご注意ください。	デフォルト値なし
wmqfte.Ssl.KeyStore.CredentialsFile	wmqfte.Ssl.KeyStore 資格情報が含まれているファイルのパス。 このプロパティの値には環境変数を含めることができる点にご注意ください。	デフォルト値なし
wmqfte.Ssl.KeyStoreType	使用する SSL 鍵ストアのタイプ。JKS および PKCS#12 の鍵ストアがサポートされています。このプロパティの値は、jks または pkcs12 のいずれかになります。	jks
wmqfte.Ssl.FipsRequired	ロガーのレベルで FIPS サポートを使用可能にすることを指定します。このプロパティの値は、true または false のいずれかになります。詳しくは、667 ページの『MFT の FIPS サポート』を参照してください。	false

関連資料

535 ページの『MFT プロパティでの環境変数の使用』

IBM WebSphere MQ 7.5 以降、ファイルまたはディレクトリーの場所を表す Managed File Transfer プロパティで環境変数を使用できます。これにより、製品の一部の実行時に使用されるファイルまたはディレクトリーの場所を、環境の変更に合わせて変えることができます。例えば、プロセスを実行するユーザーの変更などです。

605 ページの『MFT の SSL プロパティ』

SSL または TLS を IBM MQ および Managed File Transfer とともに使用して、エージェントとキュー・マネージャーとの間の許可されない接続を防止し、エージェントとキュー・マネージャーとの間のメッセージ・トラフィックを暗号化します。

MFT Java システム・プロパティ

いくつかの Managed File Transfer コマンド・プロパティとエージェント・プロパティは Java システム・プロパティとして定義する必要があります。これは、コマンド・プロパティおよびエージェント・プロパティのメカニズムを使用できない初期の関数用の構成を定義するためです。

システム・プロパティおよび Managed File Transfer コマンドを実行する JVM のその他の JVM オプションは、環境変数 `BFG_JVM_PROPERTIES` を定義することで定義します。例えば、UNIX-type タイプのプラットフォームで `com.ibm.wmqfte.maxConsoleLineLength` プロパティを設定するには、次のように変数を定義します。

```
export BFG_JVM_PROPERTIES="-Dcom.ibm.wmqfte.maxConsoleLineLength=132"
```

エージェントを Windows サービスとして実行している場合は、**fteModifyAgent** コマンドで `-sj` パラメーターを指定することにより、エージェントの Java システム・プロパティを変更できます。

プロパティ名	説明	値
<code>com.ibm.wmqfte.maxConsoleLineLength</code>	コンソールに書き込むことができる行の最大長。この長さを超える行は折り返されます。この値は、文字数ではなくバイト数で表します。	 i のデフォルト長は 132 バイトです。 UNIX、Linux、Windows、z/OS では、長さは無制限です。
<code>com.ibm.wmqfte.daemon.windows.windowsServiceLogFilesm</code>	(Windows のみ。) 保持する Windows サービス・ログ・ファイルの最大数を指定します。エージェントおよびデータベース・ロガーが Windows サービスとして稼働している場合、これらのアプリケーションのログ・ディレクトリに Windows サービス・ログ・ファイルが作成されます。Windows サービス・ログ・ファイルは、名前に接頭部 <code>service</code> が付いており、サービスの開始および停止に関するメッセージを含んでいます。	5

関連概念

[57 ページの『MFT 構成オプション \(Multiplatforms\)』](#)

Managed File Transfer は、セットアップに関する重要な情報が含まれた、操作に必要ないくつかのプロパティ・ファイルを提供します。これらのプロパティ・ファイルは、製品のインストール時に定義される構成ディレクトリにあります。

[272 ページの『MFT の使用のヒント』](#)

Managed File Transfer を最大限に活用するために役立ついくつかの提案を以下に示します。

MFT の SSL プロパティ

SSL または TLS を IBM MQ および Managed File Transfer とともに使用して、エージェントとキュー・マネージャーとの間の許可されない接続を防止し、エージェントとキュー・マネージャーとの間のメッセージ・トラフィックを暗号化します。

以下の MFT プロパティ・ファイルには、SSL プロパティが含まれています。

- [agent.properties](#) ファイル
- [coordination.properties](#) ファイル
- [command.properties](#) ファイル
-  [logger.properties](#) ファイル

Managed File Transfer で SSL を使用する方法については、51 ページの『MFT の SSL または TLS 暗号化の構成』を参照してください。

IBM WebSphere MQ 7.5 以降の場合は、ファイルまたはディレクトリーの場所を表す Managed File Transfer の一部のプロパティーで環境変数を使用できます。これにより、製品の一部の実行時に使用されるファイルまたはディレクトリーの場所を、環境の変更(プロセスを実行しているユーザーなど)に合わせて変更することができます。詳しくは、535 ページの『MFT プロパティーでの環境変数の使用』を参照してください。

agent.properties ファイルの SSL プロパティー

エージェントの `agent.properties` ファイルは、`MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` ディレクトリーにあります。これに含まれるプロパティーには、以下の SSL プロパティーがあります。

プロパティー名	説明	デフォルト値
agentSslCipherSpec	<p>エージェントとエージェント・キュー・マネージャーとの間でデータが交換される時に使用されるプロトコル、ハッシュ・アルゴリズム、および暗号化アルゴリズム、および暗号鍵で使用されるビット数を指定します。</p> <p>agentSslCipherSpec の値は、CipherSpec 名です。この CipherSpec 名は、エージェント・キュー・マネージャー・チャンネルで使用されている名前と同じです。有効な CipherSpec 名のリストは、IBM MQ classes for Java の SSL/TLS CipherSpec および CipherSuite および IBM MQ classes for JMS の SSL/TLS CipherSpec および CipherSuite に記載されています。</p> <p>agentSslCipherSpec は、agentSslCipherSuite とよく似ています。agentSslCipherSuite と agentSslCipherSpec の両方を指定した場合は、agentSslCipherSpec の値が使用されます。</p>	なし
agentSslCipherSuite	<p>エージェントとエージェント・キュー・マネージャーのデータ交換方法に関する SSL の動作を指定します。</p> <p>agentSslCipherSuite の値は、CipherSuite 名です。この CipherSuite 名は、エージェント・キュー・マネージャー・チャンネルで使用される CipherSpec 名にマップされます。詳しくは、CipherSuite と CipherSpec の名前のマッピングを参照してください。</p> <p>agentSslCipherSuite は、agentSslCipherSpec とよく似ています。agentSslCipherSuite と agentSslCipherSpec の両方を指定した場合は、agentSslCipherSpec の値が使用されます。</p>	なし
agentSslPeerName	<p>エージェント・キュー・マネージャーによって提供される名前と合致しなければならない識別名の骨組みを指定します。その識別名に基づいて、接続時にキュー・マネージャーによって提示される識別用証明書が検査されます。</p>	なし
agentSslTrustStore	<p>エージェントが信頼する証明書の場所を指定します。agentSslTrustStore の値は、ファイル・パスです。Windows のファイル・パスの場合、円記号 (&#xa5;) はエスケープ (&#xa5;&#xa5;) する必要があります。</p> <p>IBM WebSphere MQ 7.5 以降の場合は、このプロパティーの値に環境変数を含めることができます。</p>	なし

表 50. agent.properties ファイルの SSL プロパティ (続き)		
プロパティ名	説明	デフォルト値
agentSslKeyStore	エージェントの秘密鍵の場所を指定します。 agentSslKeyStore の値は、ファイル・パスです。 Windows ファイル・パスの場合は、バックスラッシュ文字 (\) をエスケープ (\\) してください。このプロパティは、エージェント・キュー・マネージャーがクライアント認証を必要とする場合にのみ、必要です。 IBM WebSphere MQ 7.5 以降の場合は、このプロパティの値に環境変数を含めることができます。	なし
agentSslFipsRequired	エージェントのレベルで FIPS サポートを使用可能にすることを指定します。このプロパティの値は、true または false のいずれかになります。詳しくは、667 ページの『MFT の FIPS サポート』を参照してください。	false
agentSslKeyStoreType	使用する SSL 鍵ストアのタイプ。JKS および PKCS#12 の鍵ストアがサポートされています。このプロパティの値は、jks または pkcs12 のいずれかになります。	jks
agentSslKeyStoreCredentialsFile	agentSslKeyStore 資格情報が含まれているファイルのパス。 IBM WebSphere MQ 7.5 以降の場合は、このプロパティの値に環境変数を含めることができます。	このプロパティのデフォルト値は、Windows の場合は %USERPROFILE%\MQMFTCredentials.xml、その他のプラットフォームの場合は \$HOME/MQMFTCredentials.xml です。
agentSslTrustStoreType	使用する SSL 鍵ストアのタイプ。JKS および PKCS#12 の鍵ストアがサポートされています。このプロパティの値は、jks または pkcs12 のいずれかになります。	jks
agentSslTrustStoreCredentialsFile	agentSslTrustStore 資格情報が含まれているファイルのパス。 IBM WebSphere MQ 7.5 以降の場合は、このプロパティの値に環境変数を含めることができます。	このプロパティのデフォルト値は、Windows の場合は %USERPROFILE%\MQMFTCredentials.xml、その他のプラットフォームの場合は \$HOME/MQMFTCredentials.xml です。

coordination.properties ファイルの SSL プロパティ

coordination.properties ファイルは MQ_DATA_PATH/mqft/config/coordination_qmgr_name ディレクトリにあります。これに含まれるプロパティには、以下の SSL プロパティがあります。

表 51. coordination.properties ファイルの SSL プロパティ		
プロパティ名	説明	デフォルト値
coordinationSslCipherSpec	コマンドと調整キュー・マネージャー間でのデータの交換時に使用されるプロトコル、ハッシュ・アルゴリズム、および暗号化アルゴリズムと、そのときに暗号鍵で使用されるビット数を指定します。 coordinationSslCipherSpec の値は、CipherSpec 名です。この CipherSpec 名は、調整キュー・マネージャー・チャンネルで使用されている CipherSpec 名と同じです。有効な CipherSpec 名のリストは、 IBM MQ classes for Java の SSL/TLS CipherSpec および CipherSuite および IBM MQ classes for JMS の SSL/TLS CipherSpec および CipherSuite に記載されています。 coordinationSslCipherSpec は、coordinationSslCipherSuite とよく似ています。coordinationSslCipherSuite と coordinationSslCipherSpec の両方を指定した場合は、coordinationSslCipherSpec の値が使用されます。	なし

表 51. coordination.properties ファイルの SSL プロパティ (続き)

プロパティ名	説明	デフォルト値
coordinationSslCipherSuite	<p>コマンドと調整キュー・マネージャーのデータ交換方法に関する SSL の動作を指定します。</p> <p>coordinationSslCipherSuite の値は、CipherSuite 名です。この CipherSuite 名は、エージェント・キュー・マネージャー・チャンネルで使用される CipherSpec 名にマップされます。詳しくは、CipherSuite と CipherSpec の名前のマッピングを参照してください。</p> <p>coordinationSslCipherSuite は、coordinationSslCipherSpec とよく似ています。coordinationSslCipherSuite と coordinationSslCipherSpec の両方を指定した場合は、coordinationSslCipherSpec の値が使用されます。</p>	なし
coordinationSslPeerName	<p>調整キュー・マネージャーによって提供される名前と一致しなければならない識別名の骨組みを指定します。その識別名に基づいて、接続時に調整キュー・マネージャーによって提示される識別用証明書が検査されます。</p>	なし
coordinationSslTrustStore	<p>コマンドが信頼する証明書の場所を指定します。coordinationSslTrustStore の値は、ファイル・パスです。Windows のファイル・パスの場合、円記号 (&#xa5;) はエスケープ (&#xa5;&#xa5;) する必要があります。</p> <p>IBM WebSphere MQ 7.5 以降の場合は、このプロパティの値に環境変数を含めることができます。</p>	なし
coordinationSslTrustStoreType	<p>使用する SSL 鍵ストアのタイプ。JKS および PKCS#12 の鍵ストアがサポートされています。このプロパティの値は、jks または pkcs12 のいずれかになります。</p>	jks
coordinationSslTrustStoreCredentialsFile	<p>coordinationSslTrustStore 資格情報が含まれているファイルのパス。</p> <p>IBM WebSphere MQ 7.5 以降の場合は、このプロパティの値に環境変数を含めることができます。</p>	このプロパティのデフォルト値は、Windows の場合は %USERPROFILE%\MQMFTCredentials.xml、その他のプラットフォームの場合は \$HOME/MQMFTCredentials.xml です。
coordinationSslKeyStore	<p>コマンドの秘密鍵の場所を指定します。coordinationSslKeyStore の値は、ファイル・パスです。Windows ファイル・パスの場合は、バックスラッシュ (¥) をエスケープ (¥¥) してください。このプロパティは、調整キュー・マネージャーがクライアント認証を必要とする場合にのみ、必要です。</p> <p>IBM WebSphere MQ 7.5 以降の場合は、このプロパティの値に環境変数を含めることができます。</p>	なし
coordinationSslKeyStoreType	<p>使用する SSL 鍵ストアのタイプ。JKS および PKCS#12 の鍵ストアがサポートされています。このプロパティの値は、jks または pkcs12 のいずれかになります。</p>	jks
coordinationSslKeyStoreCredentialsFile	<p>coordinationSslKeyStore 資格情報が含まれているファイルのパス。</p> <p>IBM WebSphere MQ 7.5 以降の場合は、このプロパティの値に環境変数を含めることができます。</p>	このプロパティのデフォルト値は、Windows の場合は %USERPROFILE%\MQMFTCredentials.xml、その他のプラットフォームの場合は \$HOME/MQMFTCredentials.xml です。
coordinationSslFipsRequired	<p>調整キュー・マネージャーのレベルで FIPS サポートを使用可能にすることを指定します。このプロパティの値は、true または false のいずれかになります。詳しくは、667 ページの『MFT の FIPS サポート』を参照してください。</p>	false

command.properties ファイルの SSL プロパティ

`command.properties` ファイルは `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` ディレクトリにあります。これに含まれるプロパティには、以下の SSL プロパティがあります。

プロパティ名	説明	デフォルト値
<code>connectionSslCipherSpec</code>	<p>コマンドとコマンド・キュー・マネージャー間でのデータの交換時に使用されるプロトコル、ハッシュ・アルゴリズム、および暗号化アルゴリズムと、そのときに暗号鍵で使用されるビット数を指定します。</p> <p><code>connectionSslCipherSpec</code> の値は、CipherSpec 名です。この CipherSpec 名は、コマンド・キュー・マネージャー・チャンネルで使用されている名前と同じです。有効な CipherSpec 名のリストは、IBM MQ classes for Java の SSL/TLS CipherSpec および CipherSuite および IBM MQ classes for JMS の SSL/TLS CipherSpec および CipherSuite に記載されています。</p> <p><code>connectionSslCipherSpec</code> は、<code>connectionSslCipherSuite</code> とよく似ています。<code>connectionSslCipherSuite</code> と <code>connectionSslCipherSpec</code> の両方を指定した場合は、<code>connectionSslCipherSpec</code> の値が使用されません。</p>	なし
<code>connectionSslCipherSuite</code>	<p>コマンドとコマンド・キュー・マネージャーのデータ交換方法に関する SSL の動作を指定します。</p> <p><code>connectionSslCipherSuite</code> の値は、CipherSuite 名です。この CipherSuite 名は、エージェント・キュー・マネージャー・チャンネルで使用される CipherSpec 名にマップされます。詳しくは、CipherSuite と CipherSpec の名前のマッピングを参照してください。</p> <p><code>connectionSslCipherSuite</code> は、<code>connectionSslCipherSpec</code> とよく似ています。<code>connectionSslCipherSuite</code> と <code>connectionSslCipherSpec</code> の両方を指定した場合は、<code>connectionSslCipherSpec</code> の値が使用されません。</p>	なし
<code>connectionSslPeerName</code>	<p>コマンド・キュー・マネージャーによって提供される名前と合致しなければならぬ識別名の骨組みを指定します。その識別名に基づいて、接続時にコマンド・キュー・マネージャーによって提示される識別用証明書が検査されます。</p>	なし
<code>connectionSslTrustStore</code>	<p>コマンドが信頼する証明書の場所を指定します。</p> <p><code>connectionSslTrustStore</code> の値は、ファイル・パスです。Windows のファイル・パスの場合、円記号 (&#xa5;) はエスケープ (&#xa5;&#xa5;) する必要があります。</p> <p>IBM WebSphere MQ 7.5 以降の場合は、このプロパティの値に環境変数を含めることができます。</p>	なし
<code>connectionSslTrustStoreType</code>	<p>使用する SSL トラストストアのタイプ。JKS および PKCS#12 の鍵ストアがサポートされています。このプロパティの値は、jks または pkcs12 のいずれかになります。</p>	jks
<code>connectionSslTrustStoreCredentialsFile</code>	<p><code>connectionSslTrustStore</code> 資格情報が含まれているファイルのパス。</p> <p>IBM WebSphere MQ 7.5 以降の場合は、このプロパティの値に環境変数を含めることができます。</p>	このプロパティのデフォルト値は、Windows の場合は <code>%USERPROFILE%\MQMFTCredentials.xml</code> 、その他のプラットフォームの場合は <code>\$HOME/MQMFTCredentials.xml</code> です。
<code>connectionSslKeyStore</code>	<p>コマンドの秘密鍵の場所を指定します。<code>connectionSslKeyStore</code> の値は、ファイル・パスです。Windows ファイル・パスの場合は、バックslash文字 (\) をエスケープ (\\) してください。このプロパティは、コマンド・キュー・マネージャーがクライアント認証を必要とする場合にのみ、必要です。</p> <p>IBM WebSphere MQ 7.5 以降の場合は、このプロパティの値に環境変数を含めることができます。</p>	なし

表 52. <i>command.properties</i> ファイルの SSL プロパティ (続き)		
プロパティ名	説明	デフォルト値
connectionSslKeyStoreType	使用する SSL 鍵ストアのタイプ。JKS および PKCS#12 の鍵ストアがサポートされています。このプロパティの値は、jks または pkcs12 のいずれかになります。 IBM WebSphere MQ 7.5 以降の場合は、このプロパティの値に環境変数を含めることができます。	jks
connectionSslKeyStoreCredentialsFile	connectionSslKeyStore 資格情報が含まれているファイルのパス。 IBM WebSphere MQ 7.5 以降の場合は、このプロパティの値に環境変数を含めることができます。	このプロパティのデフォルト値は、Windows の場合は %USERPROFILE%\MQMFTCredentials.xml、その他のプラットフォームの場合は \$HOME/MQMFTCredentials.xml です。
connectionSslFipsRequired	コマンド・キュー・マネージャーのレベルで FIPS サポートを使用可能にすることを指定します。このプロパティの値は、true または false のいずれかになります。詳しくは、667 ページの『MFT の FIPS サポート』を参照してください。	false

logger.properties ファイルの SSL プロパティ

V 9.0.4

logger.properties ファイルは、*MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name* ディレクトリにあります。ロガー・キュー・マネージャーへのクライアント・モード接続をサポートするために必要なプロパティには、以下の SSL プロパティが含まれます。

表 53. <i>logger.properties</i> ファイルの SSL プロパティ		
プロパティ名	説明	デフォルト値
wmqfte.Ssl.CipherSuite	ロガーとロガー・キュー・マネージャーのデータ交換方法に関する TLS の動作を指定します。 wmqfte.Ssl.CipherSuite の値は、CipherSuite 名です。この CipherSuite 名は、ロガー・キュー・マネージャー・チャンネルで使用される CipherSpec 名にマップされます。 詳しくは、 CipherSuite と CipherSpec の名前のマッピング を参照してください。	デフォルト値なし
wmqfte.Ssl.PeerName	ロガー・キュー・マネージャーによって提供される名前と合致しなければならない識別名の骨組みを指定します。その識別名に基づいて、接続時にキュー・マネージャーによって提示される識別用証明書が検査されます。	デフォルト値なし
wmqfte.Ssl.TrustStore	ロガーが信頼する証明書の場所を指定します。 wmqfte.Ssl.TrustStore の値は、ファイル・パスです。ファイル・パスが Windows のファイル・パスの場合、円記号 (¥) はさらに円記号を追加してエスケープ (¥¥) する必要があります。 このプロパティの値には環境変数を含めることができる点にご注意ください。	デフォルト値なし
wmqfte.Ssl.TrustStoreCredentialsFile	wmqfte.Ssl.TrustStore 資格情報が含まれているファイルのパス。 このプロパティの値には環境変数を含めることができる点にご注意ください。	デフォルト値なし

表 53. <i>logger.properties</i> ファイルの SSL プロパティ (続き)		
プロパティ名	説明	デフォルト値
<code>wmqfte.Ssl.TrustStoreType</code>	使用する SSL 鍵ストアのタイプ。JKS および PKCS#12 の鍵ストアがサポートされています。このプロパティの値は、jks または pkcs12 のいずれかになります。	jks
<code>wmqfte.Ssl.KeyStore</code>	ロガーの秘密鍵の場所を指定します。 wmqfte.Ssl.KeyStore の値はファイル・パスです。ファイル・パスが Windows のファイル・パスの場合、円記号 (¥) はさらに円記号を追加してエスケープ (¥¥) する必要があります。 このプロパティの値には環境変数を含めることができる点にご注意ください。	デフォルト値なし
<code>wmqfte.Ssl.KeyStore.CredentialsFile</code>	wmqfte.Ssl.KeyStore 資格情報が含まれているファイルのパス。 このプロパティの値には環境変数を含めることができる点にご注意ください。	デフォルト値なし
<code>wmqfte.Ssl.KeyStoreType</code>	使用する SSL 鍵ストアのタイプ。JKS および PKCS#12 の鍵ストアがサポートされています。このプロパティの値は、jks または pkcs12 のいずれかになります。	jks
<code>wmqfte.Ssl.FipsRequired</code>	ロガーのレベルで FIPS サポートを使用可能にすることを指定します。このプロパティの値は、true または false のいずれかになります。詳しくは、 667 ページの『MFT の FIPS サポート』 を参照してください。	false

MFT で使用する SHA-2 の CipherSpec と CipherSuite

Managed File Transfer は、SHA-2 CipherSpec および CipherSuite をサポートします。

IBM MQ 8.0 で、エージェントと IBM MQ キュー・マネージャーの間の接続で SHA-2 CipherSpec および CipherSuite を使用できるようにするには、IBM JRE (6.0 SR13 FP2、7.0 SR4 FP2、またはそれ以降) を使用する必要があります。

Managed File Transfer V8 で、FTPS モードのプロトコル・ブリッジを使用して FTPS サーバーに接続するために SHA-2 CipherSpec および CipherSuite を使用できるようにするには、IBM JRE (6.0 SR13 FP2、7.0 SR4 FP2、またはそれ以降) を使用する必要があります。

エージェントと IBM MQ キュー・マネージャーの間の接続で使用可能な CipherSpec および CipherSuite について詳しくは、[SSL CipherSpecs と CipherSuites](#) を参照してください。

プロトコル・ブリッジ・エージェントおよび FTPS サーバーで使用する CipherSpec および CipherSuite の構成について詳しくは、[665 ページの『プロトコル・ブリッジによる FTPS サーバーのサポート』](#) および [577 ページの『プロトコル・ブリッジ・プロパティ・ファイルのフォーマット』](#) を参照してください。

MQ 8.0 での CipherSpec の指定で説明されている新しい暗号は、IBM i JVM ではサポートされていません。したがって、IBM i プラットフォームでの SHA-2 サポートは、[MQ 7.5 での CipherSpec の指定](#)で説明されている暗号に限られます。

SP 800-131A に準拠する必要がある場合は、以下の要件を満たす必要があります。

- 適切な方法で構成した FTPS を使用する必要があります。SFTP はサポートされていません。
- リモート・サーバーは、SP 800-131A に準拠した暗号スイートのみを送信しなければなりません。

関連資料

[605 ページの『MFT の SSL プロパティ』](#)

SSL または TLS を IBM MQ および Managed File Transfer とともに使用して、エージェントとキュー・マネージャーとの間の許可されない接続を防止し、エージェントとキュー・マネージャーとの間のメッセージ・トラフィックを暗号化します。

MFT ファイル・ロガーの構成ファイル

logger.properties ファイルに加えて、Managed File Transfer スタンドアロン・ファイル・ロガーの構成ディレクトリーにも XML 構成ファイルがあります。この構成ファイルは、FileLoggerFormat.xml という名前であり、ファイル・ロガーがログ・ファイルにメッセージを書き込むために使用する形式を定義します。このファイルの内容は、FileLoggerFormat.xsd ファイルで定義されている XML スキーマに準拠する必要があります。

MFT スタンドアロン・ファイル・ロガーのデフォルトのログ・フォーマット

Managed File Transfer スタンドアロン・ファイル・ロガーのデフォルトのログ・ファイル・フォーマット定義です。

```
<?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>
        </inserts>
      </format>
    </monitorCreate>
  </messageTypes>
</logFormatDefinition>
```



```

    <separator>;</separator>
  </format>
</scheduleSkipped>
<scheduleSubmitInfo>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/schedulelog/action/@time</insert>
      <insert type="user" width="48" ignoreNull="false">/schedulelog/@ID</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="3" ignoreNull="false">/schedulelog/status/@resultCode</insert>
      <insert type="user" width="0" ignoreNull="false">/schedulelog/sourceAgent/@agent</insert>
      <insert type="user" width="12" ignoreNull="false">/schedulelog/action</insert>
      <insert type="user" width="0" ignoreNull="false">/schedulelog/originator/userID</insert>
      <insert type="user" width="0" ignoreNull="true">/schedulelog/schedule/submit</insert>
      <insert type="user" width="0" ignoreNull="true">/schedulelog/schedule/submit/@timezone</
insert>
      <insert type="user" width="3" ignoreNull="true">/schedulelog/schedule/repeat/frequency</
insert>
      <insert type="user" width="12" ignoreNull="true">/schedulelog/schedule/repeat/frequency/
@interval</insert>
      <insert type="user" width="3" ignoreNull="true">/schedulelog/schedule/repeat/expireCount</
insert>
      <insert type="user" width="0" ignoreNull="true">/schedulelog/status/supplement</insert>
    </inserts>
    <separator>;</separator>
  </format>
</scheduleSubmitInfo>
<scheduleSubmitTransfer>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/schedulelog/action/@time</insert>
      <insert type="user" width="48" ignoreNull="false">/schedulelog/@ID</insert>
      <insert type="system" width="10" ignoreNull="false">type</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/sourceAgent/@agent |
/transaction/sourceWebUser/@webGatewayAgentName |
/transaction/sourceWebGateway/@webGatewayAgentName</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/sourceAgent/@QMgr |
/transaction/sourceWebUser/@webGatewayAgentQMgr |
/transaction/sourceWebGateway/@webGatewayAgentQMgr</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/destinationAgent/@agent |
/transaction/destinationWebUser/@webGatewayAgentName |
/transaction/destinationWebGateway/@webGatewayAgentName</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/destinationAgent/@QMgr |
/transaction/destinationWebUser/@webGatewayAgentQMgr |
/transaction/destinationWebGateway/@webGatewayAgentQMgr</insert>
    </inserts>
    <separator>;</separator>
  </format>
</scheduleSubmitTransfer>
<scheduleSubmitTransferSet>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/schedulelog/action/@time</insert>
      <insert type="user" width="48" ignoreNull="false">/schedulelog/@ID</insert>
      <insert type="system" width="10" ignoreNull="false">type</insert>
      <insert type="user" width="0" ignoreNull="false">source/file | source/queue</insert>
      <insert type="user" width="5" ignoreNull="true">source/@type</insert>
      <insert type="user" width="6" ignoreNull="true">source/@disposition</insert>
      <insert type="user" width="0" ignoreNull="false">destination/file | destination/queue</
insert>
      <insert type="user" width="5" ignoreNull="true">destination/@type</insert>
      <insert type="user" width="9" ignoreNull="true">destination/@exist</insert>
    </inserts>
    <separator>;</separator>
  </format>
</scheduleSubmitTransferSet>
<transferStarted>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/transaction/action/@time</insert>
      <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="3" ignoreNull="true">/transaction/status/@resultCode</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/sourceAgent/@agent |
/transaction/sourceWebUser/@webGatewayAgentName |
/transaction/sourceWebGateway/@webGatewayAgentName</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/sourceAgent/@QMgr |
/transaction/sourceWebUser/@webGatewayAgentQMgr |
/transaction/sourceWebGateway/@webGatewayAgentQMgr</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/sourceAgent/@agentType |
/transaction/sourceWebUser/@webGatewayAgentType |
/transaction/sourceWebGateway/@webGatewayAgentType</insert>

```



```

<insert type="system" width="6" ignoreNull="false">type</insert>
<insert type="user" width="3" ignoreNull="true">/transaction/status/@resultCode</insert>
<insert type="user" width="0" ignoreNull="false">/transaction/sourceAgent/@agent |
/transaction/sourceWebUser/@webGatewayAgentName |
/transaction/sourceWebGateway/@webGatewayAgentName</insert>
<insert type="user" width="0" ignoreNull="true">/transaction/sourceAgent/@QMgr |
/transaction/sourceWebUser/@webGatewayAgentQMgr |
/transaction/sourceWebGateway/@webGatewayAgentQMgr</insert>
<insert type="user" width="0" ignoreNull="true">/transaction/sourceAgent/@agentType |
/transaction/sourceWebUser/@webGatewayAgentType |
/transaction/sourceWebGateway/@webGatewayAgentType</insert>
<insert type="user" width="0" ignoreNull="false">/transaction/destinationAgent/@agent |
/transaction/destinationWebUser/@webGatewayAgentName |
/transaction/destinationWebGateway/@webGatewayAgentName</insert>
<insert type="user" width="0" ignoreNull="true">/transaction/destinationAgent/@QMgr |
/transaction/destinationWebUser/@webGatewayAgentQMgr |
/transaction/destinationWebGateway/@webGatewayAgentQMgr</insert>
<insert type="user" width="0" ignoreNull="true">/transaction/destinationAgent/@agentType |
/transaction/destinationWebUser/@webGatewayAgentType |
/transaction/destinationWebGateway/@webGatewayAgentType</insert>
<insert type="user" width="0" ignoreNull="true">/transaction/originator/userID</insert>
<insert type="user" width="0" ignoreNull="true">/transaction/job/name</insert>
<insert type="user" width="0" ignoreNull="true">/transaction/status/supplement</insert>
</inserts>
<separator>;</separator>
</format>
</transferDelete>
<transferProgress>
<format>
<inserts>
<insert type="user" width="19" ignoreNull="false">/transaction/action/@time</insert>
<insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
<insert type="system" width="6" ignoreNull="false">type</insert>
<insert type="user" width="3" ignoreNull="true">status/@resultCode</insert>
<insert type="user" width="0" ignoreNull="false">source/file | source/queue</insert>
<insert type="user" width="0" ignoreNull="false">source/file/@size | source/queue/@size</
insert>
<insert type="user" width="5" ignoreNull="true">source/@type</insert>
<insert type="user" width="6" ignoreNull="true">source/@disposition</insert>
<insert type="user" width="0" ignoreNull="true">source/file/@alias | source/queue/@alias</
insert>
<insert type="user" width="0" ignoreNull="true">source/file/@filesystem | source/queue/
@filesystem</insert>
<insert type="user" width="0" ignoreNull="true">source/@correlationBoolean1</insert>
<insert type="user" width="0" ignoreNull="true">source/@correlationNum1</insert>
<insert type="user" width="0" ignoreNull="true">source/@correlationString1</insert>
<insert type="user" width="0" ignoreNull="false">destination/file | destination/queue</
insert>
<insert type="user" width="0" ignoreNull="false">destination/file/@size | destination/queue/
@size</insert>
<insert type="user" width="5" ignoreNull="true">destination/@type</insert>
<insert type="user" width="9" ignoreNull="true">destination/@exist</insert>
<insert type="user" width="0" ignoreNull="true">destination/file/@alias | destination/queue/
@alias</insert>
<insert type="user" width="0" ignoreNull="true">destination/file/@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>

```

関連資料

108 ページの『MFT スタンドアロン・ファイル・ロガー・フォーマット』

ファイル・ロガーにより書き込まれるメッセージ情報のフォーマットは、FileLoggerFormat.xml ファイルで定義できます。

616 ページの『スタンドアロン・ファイル・ロガーのフォーマット XSD』

スタンドアロン・ファイル・フォーマットのスキーマです。

スタンドアロン・ファイル・ロガーのフォーマット XSD

スタンドアロン・ファイル・フォーマットのスキーマです。

スキーマ

```
<?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:sequence>
</xsd:complexType>

```

```

        <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>

```

関連資料

108 ページの『MFT スタンドアロン・ファイル・ロガー・フォーマット』

ファイル・ロガーにより書き込まれるメッセージ情報のフォーマットは、FileLoggerFormat.xml ファイルで定義できます。

612 ページの『MFT スタンドアロン・ファイル・ロガーのデフォルトのログ・フォーマット』

Managed File Transfer スタンドアロン・ファイル・ロガーのデフォルトのログ・ファイル・フォーマット定義です。

SYSTEM.FTE トピック

SYSTEM.FTE トピックは、Managed File Transfer が転送を記録し、エージェント、モニター、スケジュール、およびテンプレートに関する情報を保管するために使用する調整キュー・マネージャー上のトピックです。

トピックの構造

```
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

このトピックには Managed File Transfer ネットワークのエージェント、およびそのプロパティを記述した保存パブリケーションが含まれます。このトピックに関するメッセージは、エージェント状況によって定期的に更新されます。詳しくは、[734 ページの『MFT エージェント状況メッセージ・フォーマット』](#)を参照してください。

SYSTEM.FTE/monitors/agent_name

このトピックにはエージェント *agent_name* と関連付けられたリソース・モニターを記述した保存パブリケーションが含まれます。保存パブリケーションの XML は、スキーマ MonitorList.xsd に準拠します。詳しくは、[736 ページの『MFT モニター・リスト・メッセージ・フォーマット』](#)を参照してください。

SYSTEM.FTE/Scheduler/agent_name

このトピックには、エージェント *agent_name* と関連付けられたすべてのアクティブ・スケジュールを記述する保存パブリケーションが含まれます。保存パブリケーションの XML は、スキーマ ScheduleList.xsd に準拠します。詳しくは、[741 ページの『MFT スケジュール・リスト・メッセージ・フォーマット』](#)を参照してください。

SYSTEM.FTE/Templates

このトピックには Managed File Transfer トポロジーに定義されたすべてのテンプレートを記述した保存パブリケーションが含まれます。

- 各テンプレートと関連付けられたパブリケーションは、SYSTEM.FTE/Templates/template_ID という名前です。サブトピックにパブリッシュされます。

この保存パブリケーションの内容例については、[744 ページの『MFT テンプレート XML メッセージの例』](#)を参照してください。

SYSTEM.FTE/Transfers/agent_name

このトピックにはエージェント *agent_name* で発生した転送の状況を記述したパブリケーションが含まれます。各転送と関連付けられたパブリケーションは、SYSTEM.FTE/Transfers/agent_name/transfer_ID という名前でサブトピックにパブリッシュされます。これらのパブリケーションは、個々の転送に関する進行情報を提供するために、IBM MQ Explorer・プラグインによって使用されます。パブリケーションのXMLはスキーマ TransferStatus.xsd に準拠します。詳しくは、[745 ページ](#)の『ファイル転送状況メッセージ・フォーマット』を参照してください。

SYSTEM.FTE/Log/agent_name

このトピックには、エージェント *agent_name* で発生した転送、モニター、およびスケジュールに関する情報を記録したパブリケーションが含まれます。これらのパブリケーションは、使用中の Managed File Transfer ネットワークで発生したイベントの監査レコードを提供するために、データベース・ロガーによって記録することができます。

- 各転送と関連付けられたパブリケーションは、SYSTEM.FTE/Log/agent_name/transfer_ID という名前でサブトピックにパブリッシュされ、パブリケーションのXMLはスキーマ TransferLog.xsd に準拠します。詳しくは、[748 ページ](#)の『ファイル転送ログ・メッセージ・フォーマット』を参照してください。
- 各スケジュール済み転送と関連付けられたパブリケーションは、SYSTEM.FTE/Log/agent_name/schedule_ID という名前でサブトピックにパブリッシュされ、パブリケーションのXMLはスキーマ ScheduleLog.xsd に準拠します。詳しくは、[769 ページ](#)の『スケジュール済みファイル転送ログ・メッセージ・フォーマット』を参照してください。
- 各モニターと関連付けられたパブリケーションは、SYSTEM.FTE/Log/agent_name/Monitors/monitor_name/monitor_ID という名前でサブトピックにパブリッシュされ、パブリケーションのXMLはスキーマ MonitorLog.xsd に準拠します。詳しくは、[774 ページ](#)の『MFT モニター・ログ・メッセージ・フォーマット』を参照してください。

MFT エージェント・キュー設定

fteCreateAgent コマンドで生成される MQSC コマンド・スクリプトは、次の値に設定されたパラメーターでエージェント・キューを作成します。キューを作成するために指定された MQSC スクリプトを使用せず、キューを手動で作成する場合、指定された値に次のパラメーターを設定する必要があります。

エージェント操作キュー

エージェントの操作キューの名前は、以下のとおりです。

- 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

パラメーター	値 (該当する場合)
DEFPRTY	0
DEFSOPT	SHARED
GET	ENABLED
MAXDEPTH	5000
MAXMSGL	4194304
MSGDLVSQ	PRIORITY
PUT	ENABLED

表 54. エージェント操作キューのパラメーター (続き)

パラメーター	値 (該当する場合)
RETINTVL	999999999
SHARE	
NOTRIGGER	
USAGE	NORMAL
REPLACE	

エージェント権限キュー

エージェントの権限キューの名前は、以下のとおりです。

- 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

表 55. エージェント権限キューのパラメーター

パラメーター	値 (該当する場合)
DEFPRTY	0
DEFSOPT	SHARED
GET	ENABLED
MAXDEPTH	0
MAXMSGL	0
MSGDLVSQ	PRIORITY
PUT	ENABLED
RETINTVL	999999999
SHARE	
NOTRIGGER	
USAGE	NORMAL
REPLACE	

関連資料

388 ページの『fteCreateAgent (MFT エージェントの作成)』

fteCreateAgent コマンドは、Managed File Transfer Agent とその関連構成を作成します。

MFT システム・キューおよびシステム・トピック

Managed File Transfer には、内部使用専用のいくつかのシステム・キューおよび1つのシステム・トピックがあります。

名前が SYSTEM.FTE で始まるキューは、Managed File Transfer (MFT) の内部システム・キューです。これらのキューは削除しないでください。削除すると、IBM MQ MFT が正しく動作しなくなります。622 ページの表 56 は、各キューにあるメッセージのタイプを示しています。

表 56. キュー名、タイプ、および使用法		
キュー名	キュー・タイプ	使用法
SYSTEM.FTE.AUTHAGT1.agent_name	Authority	転送要求を送受信する権限を構成するためのキュー。
SYSTEM.FTE.AUTHTRN1.agent_name	Authority	管理対象転送を開始およびキャンセルする権限を構成するためのキュー。また、管理対象呼び出しを開始します。
SYSTEM.FTE.AUTHMON1.agent_name	Authority	同じユーザーによって作成されたリソース・モニターをユーザーが作成または削除できるようにする権限を構成するためのキュー。
SYSTEM.FTE.AUTHOPS1.agent_name	Authority	別のユーザーによって作成されたリソース・モニターおよびスケジュールされた転送を削除する権限を構成するためのキュー。
SYSTEM.FTE.AUTHSCH1.agent_name	Authority	同じユーザーによって作成されたスケジュール済み転送を作成または削除する権限を構成するためのキュー。
SYSTEM.FTE.AUTHADM1.agent_name	Authority	fteStopAgent コマンドの -m オプションを使用してエージェントをシャットダウンする権限を構成するためのキュー。
SYSTEM.FTE.COMMAND.agent_name	Operation	コマンド要求をエージェントに送信するためのキュー。
SYSTEM.FTE.DATA.agent_name	Operation	ソース・エージェントによって送信されたデータを保持するために宛先エージェントによって使用されるキュー。
SYSTEM.FTE.REPLY.agent_name	Operation	宛先エージェントから応答を受信するためのキュー。
SYSTEM.FTE.STATE.agent_name	Operation	転送要求の状況を保持するためのキュー。
SYSTEM.FTE.EVENT.agent_name	Operation	リソース・モニター・ヒストリーを保持するためのキュー。

エージェントがメッセージからファイルへの転送またはファイルからメッセージへの転送に関与している場合は、SYSTEM.FTE.STATE.agent_name キューの定義を変更して、これらの管理対象転送を実行できるようにする必要がある場合があります。この設定について詳しくは、メッセージ・サイズに関連した MQ 属性および MFT プロパティを設定するためのガイダンスを参照してください。



重要: 他のシステム・キューの定義は変更しないでください。

また、SYSTEM.FTE トピックは内部使用専用であるため、変更したり削除したりしないでください。

一時キュー

Managed File Transfer は、いくつかの目的で一時キューを作成します。各キューの名前は WMQFTE で始まります。このグループにデフォルトで追加されます (ピリオドはデフォルトの接頭部の一部です。) この接頭部を変更する場合は、`command.properties` ファイルまたは `coordination.properties` ファイル、あるいはその両方の **dynamicQueuePrefix** プロパティを使用します。 `command.properties` ファイルのプロパティは、エージェントからの応答が必要なコマンドに対する応答のために作成される一時キューの接頭部を設定するために使用されます。 `coordination.properties` ファイルのプロパティは、その他の目的のために作成される一時キューの接頭部を設定するために使用されます。例えば、`WMQFTE.FTE.TIMECHECK.QUEUE` という一時キューの場合、`WMQFTE` が **dynamicQueuePrefix** プロパティによって定義された値です。

関連資料

359 ページの『MFT エージェント・アクションのユーザー権限の制限』

グループを使用してリソースに対するアクセスを管理することに加えて、追加レベルのセキュリティーを使用可能にして、ユーザーが実行できる Managed File Transfer エージェント・アクションを制限することができます。エージェントの権限キューに対する権限をユーザーに付与して、ユーザーが特定のエージェント・アクションを実行できるようにします。

MFT オブジェクト命名規則

Managed File Transfer オブジェクトに以下の命名規則を使用してください。

- エージェントおよびログガーの名前:
 - 最大 28 文字の長さにすることができ、大/小文字の区別はありません。
 - 小文字または大/小文字混合で入力された名前は、大文字に変換されます。
 - 標準 IBM MQ オブジェクト命名規則に準拠していなければなりません。これらの規則の詳細については、[IBM MQ オブジェクトの命名規則](#)を参照してください。
- IBM MQ オブジェクト命名規則に加えて、以下の規則が適用されます。
 - スラッシュ (/) 文字をエージェント名またはログガー名に使用することはできません。
 - パーセント (%) 文字をエージェント名またはログガー名に使用することはできません。
- プロパティ・ファイルのプロパティ名は大/小文字を区別します。
- キュー・マネージャー名は、大/小文字を区別します。
- 一部のプラットフォームでは、ファイル名で大/小文字を区別します。
- リソース・モニターおよび転送テンプレートの名前:
 - 大/小文字の区別はありません。
 - 小文字または大/小文字混合で入力された名前は、大文字に変換されます。
 - アスタリスク (*), パーセント (%), および疑問符 (?) を含めることはできません。
- プロトコル・ファイル・サーバー名:
 - 2 文字以上の長さでなければならず、最大長の制限はありません。
 - 大/小文字の区別はありません。
 - 標準 IBM MQ オブジェクト命名規則に準拠していなければなりません。これらの規則の詳細については、[IBM MQ オブジェクトの命名規則](#)を参照してください。

IBM i 統合ファイル・システム (IFS) のファイル

IBM i IFS のファイル名には次の文字を使用できません。

- 円記号 (¥)
- 順方向斜線 (/)
- コロン (:)
- アスタリスク (*)

- 疑問符 (?)
- 引用符 (")
- より小記号 (<)
- より大記号 (>)
- 垂直バー (|)

名前にこれらの文字のいずれかを含むファイルを IBM i IFS に転送しようとする、これらのファイルの転送は失敗します。

データ・セット名

z/OS データ・セットには、データ・セット名で名前の最大長と使用できる文字に影響する命名上の制約があります。PDS データ・セット・メンバー名は、最大 8 文字で、ドット (.) 文字を使用することはできません。データ・セットに転送する場合、明示的に名前を指定する必要があります。つまり、これらの命名上の制約によって問題が発生しないことを意味します。ただしファイルから PDS メンバーに転送する場合には、ファイル・パスが PDS メンバー名にマップされない可能性があります。PDS データ・セットに転送する場合、個々のソース・ファイルが PDS メンバーとなり、各メンバー名がソースの名前から生成されます。

PDS メンバー名は z/OS の非修飾名で、以下の正規表現で定義されます。

```
[a-zA-Z$#@][a-zA-Z0-9$#@]{0-7}
```

次の方式が、ソース・データ・セットまたはソース・ファイル名を有効な PDS メンバー名に変換するために使用されます。以下の順に考慮事項が適用されます。

1. 名前のうち、最後のスラッシュ (/)、最後のバックスラッシュ (¥)、最後のコロン (:) 以降の文字のみが使用されます。つまり、ファイル・パスの名前部分のみが使用されます。
2. ソース・ファイル (データ・セット、PDS メンバー以外) の場合、最後のドット (.) 文字以降 (ドットを含む) が無視されます。
3. 8 文字より長い名前については、最後の 8 文字のみが使用されます。
4. ドット文字はアットマーク (@) 文字に置き換えられます。
5. 無効文字はアットマーク (@) 文字に置き換えられます。
6. この変換により文字が生成されない場合、PDS メンバー名は @ となります。

MFT 管理参照

MFT エージェントの状況値

fteListAgents と **fteShowAgentDetails** のいずれかのコマンドを実行すると、エージェント状況情報が生成されます。この状況として生成される可能性がある値は、以下のとおりです。

ACTIVE

エージェントは実行中であり、ファイルを送信または受信しています。エージェントは、その状況を一定の間隔で公開しています。最新のアップデートは、予期した期間内に受け取りました。

準備完了

エージェントは実行中ですが、ファイルを送信または受信していません。エージェントは、その状況を一定の間隔で公開しています。最新のアップデートは、予期した期間内に受け取りました。

STARTING

エージェントは開始していますが、転送を実行する準備ができていません。

UNREACHABLE

エージェント状況のアップデートを予期した時間間隔で受け取っていません。エージェントは、エラーのために実行を停止したか、突然にシャットダウンしたか、実行しているものの通信問題が発生して

いる可能性があります。詳しくは、[304 ページの『エージェントが UNKNOWN 状態にあると表示される場合の処理』](#)を参照してください。

STOPPED

エージェントが停止しました。It was shut down in a controlled manner.

ENDED UNEXPECTEDLY

エージェントが予期せず終了しました。maxRestartInterval 期間内に maxRestartCount を超える再始動が行われ、maxRestartDelay 値が 0 以下である場合を除き、エージェントは自動的に再始動されます。これらのプロパティについて詳しくは、[550 ページの『MFT agent.properties ファイル』](#)を参照してください。

NO_INFORMATION

エージェントのバージョンが IBM WebSphere MQ File Transfer Edition 7.0.2 以前である可能性があります。エージェントは、このコマンドが処理できる形式では更新を公開していません。

不明

エージェントの状況を判別できません。このツールでは認識できない状況を公開した可能性があります。ネットワーク上で製品バージョンの混合がある場合には、このツールのインストール・バージョンをアップグレードすれば、この問題を解決できるかもしれません。

V 9.0.2 IBM MQ 9.0.2 では、コマンドを実行したときや、調整マネージャーに接続している各エージェントとそれぞれの個々のプロパティのリストを表示するときに、エージェントの **Status Age** 値が新たに表示されるようになりました。この値は、最後に報告されたそれらの状況の経過時間を示します。詳しくは、[304 ページの『エージェントが UNKNOWN 状態にあると表示される場合の処理』](#)を参照してください。

PROBLEM

エージェント・コマンド・ハンドラーが機能していない可能性があります。エージェントは状況メッセージを公開していますが、これらの状況メッセージは最新のものではありません。

関連資料

[473 ページの『fteListAgents: 調整キュー・マネージャーの MFT エージェントのリスト』](#)

fteListAgents コマンドは、特定の調整キュー・マネージャーに登録されているすべての Managed File Transfer のエージェントをリストするために使用します。

[518 ページの『fteShowAgentDetails』](#)

fteShowAgentDetails コマンドは、特定の Managed File Transfer Agent の詳細を表示するために使用します。これらは、エージェントの Managed File Transfer 調整キュー・マネージャーによって保管される詳細です。

[314 ページの『ファイル転送が停止したと思われる場合の対処法』](#)

負荷の重いシステムであったり、ソース・エージェントと宛先エージェントとの間でネットワーク問題があったりする場合には、転送が待機状態またはリカバリー状態で停止しているように見えることがあります。この原因となるいくつかの要因があります。

[304 ページの『エージェントが UNKNOWN 状態にあると表示される場合の処理』](#)

エージェントが実行中であり、**ftePingAgent** コマンドに正常に応答し、項目が正常に転送されています。ただし、**fteListAgents** コマンドと **fteShowAgentDetails** コマンド、および IBM MQ Explorer Managed File Transfer プラグインは、エージェントが UNKNOWN 状態であることを報告します。

Multi MFT プロセス・コントローラーの概要

IBM MQ Managed File Transfer (MFT) プロセス・コントローラーは、何らかの理由で終了した場合に、MFT エージェントを開始し、そのプロセスを再始動します。エージェント・プロセスごとに 1 つのプロセス・コントローラーがあります。

注: プロセス・コントローラーは IBM MQ for Multiplatforms のみに適用できます。

z/OS IBM MQ for z/OS では、エージェント・プロセスは自動再始動マネージャー (ARM) によって再始動されます。詳しくは、[z/OS 自動リスタート・マネージャー \(ARM\) 用の MFT の構成](#)を参照してください。

プロセス・コントローラーの動作

fteStartAgent コマンドを実行すると、そのエージェントのプロセス・コントローラーのインスタンスが開始され、プロセス・コントローラーがエージェント・プロセスを開始します。

fteStopAgent コマンドを実行すると、そのエージェントのプロセス・コントローラーに接続し、停止要求を送信します。プロセス・コントローラーは要求を受け取り、エージェント・プロセスを停止してからシャットダウンします。

プロセス・コントローラーはエージェント・プロセスをモニターします。エージェント・プロセスが予期せずに停止した場合、プロセス・コントローラーはエージェント・プロセスを再始動します。

デフォルトでは、エージェント・プロセスが2分以内に5回停止した場合、プロセス・コントローラーはそれ自体をシャットダウンし、エージェントの再始動を再試行しません。この場合は、**fteStartAgent** コマンドを使用してエージェントを手動で再始動する必要があります。

この動作を変更するには、以下のエージェント・プロパティーを変更します。

- **maxRestartCount**
- **maxRestartDelay**
- **maxRestartInterval**

BINDINGS トランスポートを使用してエージェント・キュー・マネージャーに接続するようにエージェントを構成した場合、プロセス・コントローラーは始動時にこのキュー・マネージャーへの接続を作成します。その後、プロセス・コントローラーはこの接続をモニターします。

キュー・マネージャーが使用不可になったために接続が切断された場合、プロセス・コントローラーはエージェントを停止してから、定期的な間隔で接続の再確立を試行します。

再接続の試行間隔は、エージェント・プロパティー **agentQMgrRetryInterval** によって決定されます。キュー・マネージャーが再び使用可能になり、プロセス・コントローラーがそのキュー・マネージャーに接続できるようになると、プロセス・コントローラーはエージェント・プロセスを再始動します。

注: エージェントが CLIENT トランスポートを使用してエージェント・キュー・マネージャーに接続するように構成されている場合、エージェント・プロセスは、キュー・マネージャーから切断されてもアクティブのままです。この場合、エージェント・プロセスは定期的に再接続を試行します。

このセクションに記載されている4つのプロパティーについては、「*MFT agent.properties* ファイル」トピックの「[拡張エージェント・プロパティー: プロセス・コントローラー](#)」セクションを参照してください。

プロセス・コントローラーのログ・ファイル

プロセス・コントローラーは、通知メッセージをイベント・ログに書き込みます。これは、`pceventN.log` という名前のファイルです。ここで、Nは数値です。これは、ディレクトリー `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name/logs/` にあります。

各プロセス・コントローラー・イベント・ログ・ファイルのサイズ、およびヒストリカル・ファイルの数は、エージェント・プロパティー **outputLogSize** および **outputLogFiles** によって決定されます。

このセクションで言及されているプロパティーについては、「*MFT agent.properties* ファイル」トピックの「[拡張エージェント・プロパティー: トレースおよびロギング](#)」セクションを参照してください。

注: これらのプロパティーは、プロセス・コントローラー・ログ・ファイルだけでなく、エージェント・ログ・ファイル (`outputN.log` と呼ばれる) のサイズと数を決定するためにも使用されます。

プロセス・コントローラー・イベント・ログに書き込まれるメッセージには、プロセス・コントローラーのプロセス ID、およびエージェント・プロセスのプロセス ID が含まれます。これらのメッセージの例をいくつか以下に示します。

```
[21/06/2022 16:17:40.000 GMT Daylight Time] 00000000000049e0
ProcessContro I BFGPC0003I: IBM MQ Managed File Transfer process controller started.
```

Log files located at: C:\ProgramData\IBM\MQ\mqft\logs\QM1\agents\AGENT1.

```
[21/06/2022 16:17:55.000 GMT Daylight Time] 00000000000049e0
ProcessContro I BFGPC0007I: IBM MQ Managed File Transfer process controller with process
identifier 18736 started AGENT1@QM1 with process identifier 1748.
```

```
[21/06/2022 16:19:20.000 GMT Daylight Time] 00000000000049e0
ProcessContro I BFGPC0027W: Process has ended with return code 1 and will be
restarted to attempt to recover the problem.
```

```
[21/06/2022 16:19:20.000 GMT Daylight Time] 00000000000049e0
ProcessContro I BFGPC0007I: IBM MQ Managed File Transfer process controller with process
identifier 18736 started AGENT1@QM1 with process identifier 1304.
```

ここで、エージェント AGENT1 に関連付けられたプロセス・コントローラーは、プロセス ID 18736 で実行されていました。

最初はエージェント・プロセスを開始しました。このプロセスのプロセス ID は 1748 でした。

エージェントが開始された直後に、プロセス・コントローラーは予期せずに停止したことを検出したため、プロセス・コントローラーを再始動しました。再始動後のエージェント・プロセスのプロセス ID は 1304 です。

関連資料

629 ページの『MFT プロセス・コントローラーの終了コード』

Managed File Transfer プロセス・コントローラーが終了すると、プロセス・コントローラーが終了した理由を示す終了コードを含んだ BFGPC0004I メッセージが生成されます。

MFT エージェント・プロセス・コントローラーの状況値

fteShowAgentDetails コマンドは、エージェント・プロセス・コントローラーの状況情報を生成します。この状況として生成される可能性がある値は、以下のとおりです。

WAITING

エージェント・プロセス・コントローラーは、キュー・マネージャーが使用可能になってからエージェントを開始するため、待機しています。

開始済み

エージェント・プロセス・コントローラーは、エージェント・プロセスを開始しました。

STOPPED

エージェントを停止するように要求されたか、あるいは再開間隔の間に再開されたエージェント・プロセスが多すぎたため、エージェント・プロセス・コントローラーは停止されました。

RECOVERING

エージェント・プロセスが予期せずに停止したため、プロセス・コントローラーはそのプロセスの再開を試行します。

ISTOPPING

エージェント・プロセスが即時シャットダウン要求を受け取りました。エージェント・プロセスが停止すると、プロセス・コントローラーは停止します。

CSTOPPING

エージェント・プロセスは、制御された方法でのシャットダウン要求を受け取りました。エージェント・プロセスが停止すると、プロセス・コントローラーは停止します。

不明

エージェント・プロセス・コントローラーの状況を判別できません。エージェント・プロセス・コントローラーは実行されていないか、**fteShowAgentDetails** コマンドが実行されたシステムとは異なる別のシステムで実行されている可能性があります。

関連資料

518 ページの『fteShowAgentDetails』

fteShowAgentDetails コマンドは、特定の Managed File Transfer Agent の詳細を表示するために使用します。これらは、エージェントの Managed File Transfer 調整キュー・マネージャーによって保管される詳細です。

MFT ロガーの状況値

fteShowLoggerDetails コマンドは、ロガーの状況情報を生成します。この状況として生成される可能性がある値は、以下のとおりです。

ACTIVE

ロガーは実行中であり、ファイルを送信または受信しています。ロガーは、その状況を一定の間隔で公開しています。最新のアップデートは、予期した期間内に受け取りました。

準備完了

ロガーは実行中ですが、ファイルを送信または受信していません。ロガーは、その状況を一定の間隔で公開しています。最新のアップデートは、予期した期間内に受け取りました。

STARTING

ロガーは開始中ですが、転送を実行する準備はまだできていません。

UNREACHABLE

予期された時間間隔では、ロガー状況の更新を受信しませんでした。ロガーはエラーのために実行を停止したか、突然にシャットダウンしたか、または実行しているものの通信問題が発生している可能性があります。

STOPPED

ロガーは停止しました。It was shut down in a controlled manner.

ENDED UNEXPECTEDLY

ロガーが予期せずに終了しました。maxRestartInterval 期間内に maxRestartCount 回を超える再始動があり、maxRestartDelay 値が 0 以下である場合を除き、ロガーは自動的に再始動されます。これらのプロパティの詳細は、[118 ページの『MFT ロガーの構成プロパティ』](#)を参照してください。

fteShowLoggerDetails コマンドの場合、この状況の詳細には、ロガー処理出口コードである状況コードが含まれます。既知の出口コードのリストについては、『[処理出口コード](#)』を参照してください。

NO_INFORMATION

ロガーのバージョンが IBM WebSphere MQ File Transfer Edition 7.0.2 以前である可能性があります。ロガーは、このコマンドが処理できる形式では更新を公開していません。

不明

ロガーの状況を判別できません。このツールでは認識できない状況を公開した可能性があります。ネットワーク上で製品バージョンの混合がある場合には、このツールのインストール・バージョンをアップグレードすれば、この問題を解決できるかもしれません。

PROBLEM

ロガー・コマンド・ハンドラーが機能していない可能性があります。ロガーは状況メッセージを公開していますが、これらの状況メッセージは最新のものではありません。

関連資料

[525 ページの『fteShowLoggerDetails』](#)

fteShowLoggerDetails コマンドは、特定の Managed File Transfer ロガーの詳細を表示するために使用します。

MFT ロガー・プロセス・コントローラーの状況値

fteShowLoggerDetails コマンドは、ロガー・プロセス・コントローラーの状況情報を生成します。この状況として生成される可能性がある値は、以下のとおりです。

WAITING

ロガー・プロセス・コントローラーは、キュー・マネージャーが使用可能になってからロガーを開始するため、待機しています。

開始済み

ロガー・プロセス・コントローラーは、ロガー・プロセスを開始しました。

STOPPED

ロガーを停止するように要求されたか、あるいは再開間隔の間に再開されたロガー・プロセスが多すぎたため、ロガー・プロセス・コントローラーは停止されました。

RECOVERING

ロガー・プロセスが予期せずに停止したため、プロセス・コントローラーはそのプロセスの再開を試行します。

ISTOPPING

ロガー・プロセスが即時シャットダウン要求を受け取りました。ロガー・プロセスが停止すると、プロセス・コントローラーは停止します。

CSTOPPING

ロガー・プロセスは、制御された方法でのシャットダウン要求を受け取りました。ロガー・プロセスが停止すると、プロセス・コントローラーは停止します。

不明

ロガー・プロセス・コントローラーの状況を判別できません。ロガー・プロセス・コントローラーは実行されていないか、`fteShowLoggerDetails` コマンドが実行されたシステムとは異なる別のシステムで実行されている可能性があります。

関連資料

525 ページの『[fteShowLoggerDetails](#)』

fteShowLoggerDetails コマンドは、特定の Managed File Transfer ロガーの詳細を表示するために使用します。

MFT プロセス・コントローラーの終了コード

Managed File Transfer プロセス・コントローラーが終了すると、プロセス・コントローラーが終了した理由を示す終了コードを含んだ BFGPC0004I メッセージが生成されます。

プロセス・コントローラーが終了したことを示す次のメッセージが表示されます。

BFGPC0004I IBM MQ Managed File Transfer プロセス・コントローラーは終了コード `reason_code` で終了しました。

ここで、`reason_code` は、プロセス・コントローラーが終了した理由を示しています。

注：プロセス・コントローラーからの終了コードのほとんどはオペレーティング・システムの標準終了コードを反映していますが、一部の終了コードは特定の目的で定義されたものであり、プロセス・コントローラー・ログ・ファイルにも具体的なメッセージが必ず書き込まれます。

理由コード	説明
RC_SUCCESS = 0	プロセス・コントローラーは正常に終了しました。
RC_FAILURE = 1	プロセス・コントローラー障害の汎用戻りコード (一般的には返されることはない)。
RC_EXIT = 2	プロセス・コントローラーは強制的に終了させられました (例えば、診断システムがプロセス・コントローラーの強制終了を要求した)。
RC_ABEND = 70	プロセス・コントローラーは、リカバリー不能な問題が発生したため、強制的に終了しています。
RC_QMUNAVAIL = 75	プロセス・コントローラーのキュー・マネージャーを使用できないため、プロセス・コントローラーを続行できません。
RC_CONFIG = 78	始動構成データに問題があるため、プロセス・コントローラーは続行できません。

これらの終了コードは、`pceventX.log` に書き込まれます。ここで、`X`は何らかの数値です。例えば、ログ・ファイル名は `pcevent0.log` のようになります。

関連情報

Managed File Transfer 診断メッセージ: BFGPC0001 - BFGPC9999

ファイルの転送に関するガイドライン

転送を行うオペレーティング・システムや、バイナリー・モードとテキスト・モードのどちらで転送するかに応じて、予期される動作についてのガイドラインが存在します。

詳しくは、関連トピックをお読みください。

関連資料

[631 ページの『z/OS と分散システム間のファイルおよびデータ・セットの転送』](#)

Managed File Transfer を使用して、z/OS と分散ファイル・システムの間でファイルやサポートされるデータ・セット・タイプを転送することができます。以下の性質を注意深く検討してください。これは、転送元と転送先のシステムのタイプによって異なります。

[632 ページの『z/OS でのデータ・セット間の転送』](#)

Managed File Transfer を使用して、z/OS データ・セット間の転送を行うことができます。以下の性質を注意深く検討し、データ・セットの転送が正しく行われるようにしてください。

[634 ページの『Connect:Direct ノードとの間のデータ・セット転送』](#)

Connect:Direct ブリッジを使用して、Managed File Transfer エージェントと IBM Sterling Connect:Direct ノードの間でデータ・セット転送を実行できます。転送のソースと宛先のいずれかまたは両方としてデータ・セットを指定できます。

[637 ページの『Connect:Direct プロセス・ステートメント・パラメーターと BPXWDYN キーのマッピング』](#)

ソースまたは宛先のいずれかが Connect:Direct ノードになっている状況でデータ・セットの転送要求を送信すると、指定したサポート対象の BPXWDYN キーが、Connect:Direct プロセスによって受け入れられる形式に変換されます。

[642 ページの『MFT で使用できない BPXWDYN のプロパティー』](#)

いくつかの BPXWDYN オプションは、**fteCreateTemplate** コマンド、**fteCreateTransfer** コマンド、または `agent.properties` ファイル内の **bpxdynAllocAdditionalOptions** プロパティーを使用するときに指定してはなりません。

[643 ページの『MFT でのテキスト・ファイルの転送』](#)

テキスト・ファイルの転送には、あるコード・ページから別のコード・ページへの、ファイルのコード・ページの変換が伴います。テキスト・ファイルの転送には、システム間の CRLF (復帰改行) 文字の変換も伴います。このトピックでは Managed File Transfer のテキスト・ファイル転送動作をまとめています。

[645 ページの『Connect:Direct と MFT の間のテキスト・ファイル転送』](#)

テキストの転送には、あるコード・ページから別のコード・ページへの、ファイルのコード・ページの変換が伴います。テキスト・ファイルの転送には、システム間の CRLF (復帰改行) 文字の変換も伴います。このトピックでは、Managed File Transfer Agent と Connect:Direct ノードの間のテキスト・ファイル転送の動作をまとめます。

[646 ページの『プロトコル・ブリッジ・エージェントとの間のファイル転送』](#)

プロトコル・ブリッジ・エージェントを使用すると、Managed File Transfer ネットワークの外部にある FTP ファイル・サーバーまたは SFTP ファイル・サーバーとの間でファイルを転送できます。

[647 ページの『IBM i システム間のファイル転送』](#)

テキスト・モードで Managed File Transfer を使って IBM i システム間のファイル転送を行い、ファイル内のデータを変換する場合は、このトピックの情報を検討してください。

[651 ページの『IBM i の QSYS.LIB にある保存ファイルの転送』](#)

Managed File Transfer は、QSYS.LIB ファイル・システムにある保存ファイルを 2 つの IBM i システム間で転送することをサポートします。保存ファイルのファイル転送を要求する際は、以下の情報を検討してください。

[652 ページの『世代別データ・グループ \(GDG\) の転送』](#)

Managed File Transfer は、z/OS で、ソース・データ・セットと宛先データ・セットの世代別データ・グループ (GDG) をサポートしています。絶対および相対 GDG 名がサポートされています。新しい世代に書き込む場合には、基本 GDG が存在する必要があります。

653 ページの『MFTでのワイルドカード文字の使用』

ファイル転送用にソース・ファイル名およびソース・ファイル・パスを指定するときに、ワイルドカード文字を使用できます。これにより、複数のファイルを同時に選択することができるようになります。

z/OS

z/OS と分散システム間のファイルおよびデータ・セットの転送

Managed File Transfer を使用して、z/OS と分散ファイル・システムの間でファイルやサポートされるデータ・セット・タイプを転送することができます。以下の性質を注意深く検討してください。これは、転送元と転送先のシステムのタイプによって異なります。

Managed File Transfer は、z/OS で、ソース・データ・セットと宛先データ・セットの世代別データ・グループ (GDG) をサポートしています。絶対および相対 GDG 名がサポートされています。新しい世代に書き込む場合には、基本 GDG が存在している必要があります。

ファイルまたはデータ・セットを磁気テープに転送すると、磁気テープ内の既存データ・セットが置き換えられます。新しいデータ・セットの属性は、転送定義で渡された属性を基にして設定されます。属性は、何も指定されていなければ、ソース・データ・セットと同じ属性またはデフォルト値 (ソースがファイルである場合) に設定されます。既存の磁気テープ・データ・セットの属性は無視されます。

ファイルからデータ・セットへの転送 - バイナリー転送

宛先レコード長は、宛先データ・セットの形式によって決まります。データ・セットが宛先システムに存在することを確認してください。または、データ・セットが正しく作成されるように、宛先データ・セットを正しい属性で指定してください。属性を指定しない場合、形式が不定形式レコードで、(DEVTYPE マクロによって返された) デバイスに合った最大ブロック・サイズ (BLKSIZE) の物理順次データ・セットというデフォルトがシステムによって指定されます。例えば、DASD ならサイズは 6144 で、テープならサイズは 32760 です。分散システムのファイルをバイナリー・モードで z/OS データ・セットに転送する場合は、以下の性質に注意してください。

物理順次 (PS) 宛先データ・セット

- 分散システム上のソース・ファイルが順次読み取られ、各レコードまたはブロックが埋められます。
- 可変長フォーマットのデータ・セットの場合は、容量が満杯になるまで各レコードが埋められます。

区分データ・セット (PDS) 宛先データ・セット

- 各ソース・ファイルは同じまたは同等の名前で PDS メンバーにコピーされます。ファイル名がメンバー名の許容最大長より長い場合は、ファイル名は有効なメンバー名に変換されます。メンバー名の詳細については、『オブジェクトの命名規則』を参照してください。ソース・ファイルがディレクトリーの場合、そのディレクトリーの各ファイルが PDS のメンバーになります。
- PDS メンバーが存在する場合、転送について既存の宛先ファイルの上書きが指定されていれば、メンバーは上書きされます。上書きを指定しない場合、転送が失敗します。
- 分散システム上のソース・ファイルが順次読み取られ、メンバーの各レコードまたはブロックが埋められます。
- 可変長フォーマットの PDS メンバーの場合は、容量が満杯になるまで各レコードが埋められます。

ファイルからデータ・セットへの転送 - テキスト転送

宛先レコード長は、宛先データ・セットの形式によって決まります。データ・セットが宛先システムに存在することを確認してください。または、データ・セットが正しく作成されるように、宛先データ・セットを正しい属性で指定してください。分散システムのファイルをテキストとして z/OS データ・セットに転送する場合は、以下の性質に注意してください。

物理順次 (PS) 宛先データ・セット

- テキストの各行がレコード (または不定形式レコード (RECFM=U) データ・セットの場合はブロック) になります。データ・セット・レコードに行末文字は存在しません (非 ASA データ・セットの場合のみ)。
- 宛先データ・セットで ASA フォーマット制御文字が使用される場合は、行末文字は事実上、同等の ASA フォーマット制御コードに変換されます。
- 行がレコードより長い場合、レコードの境界で行が分割され、次のレコードに続きます。

PDS 宛先データ・セット

- 各ソース・ファイルは同じまたは同等の名前で PDS メンバーにコピーされます。ファイル名がメンバー名の許容最大長より長い場合は、ファイル名は有効なメンバー名に変換されます。メンバー名の詳細については、『[オブジェクトの命名規則](#)』を参照してください。ソース・ファイルがディレクトリーの場合、そのディレクトリーの各ファイルが PDS のメンバーになります。
- PDS メンバーが存在する場合、転送について既存の宛先ファイルの上書きが指定されていれば、メンバーは上書きされます。上書きを指定しない場合、転送が失敗します。
- テキストの各行がレコード (または不定形式レコード (RECFM=U) データ・セットの場合はブロック) になります。メンバー・レコードに行末文字は存在しません (非 ASA データ・セットの場合のみ)。
- 宛先データ・セットで ASA フォーマット制御文字が使用される場合は、行末文字は事実上、同等の ASA フォーマット制御コードに変換されます。
- 行がレコードより長い場合、レコードの境界で行が分割され、次のレコードに続きます。

データ・セットからファイルへの転送 - バイナリーおよびテキスト転送

データ・セットからファイルにバイナリーまたはテキストとして転送する場合は、以下の性質に注意してください。

- 各レコードの内容はバイナリー形式でファイルに転送されます。レコード、ブロック・フォーマット情報、または ASA フォーマット制御文字は転送されません。
- テキスト転送のみの場合、各データ・セット・レコードは、宛先エージェントのコード・ページに変換されたテキストを持つ行になります。つまり、Windows 宛先システムでは復帰改行 (CRLF) が追加され、UNIX 宛先システムでは復帰 (CR) が追加されます。
- 非 VSAM および PS ソース・データ・セット。** ソース・データ・セットのレコードは宛先ファイルに転送され、連結してまとめられます。宛先ファイルが存在する場合には、ファイル転送で指定した宛先ファイル動作オプションに応じて、ファイルが上書きされます。ファイルではなくディレクトリーとして宛先を指定した場合、宛先ファイル名は、高位修飾子 (HLQ) を除いたデータ・セット名になります。
- PDS ソース・データ・セット。** 指定された各メンバー (メンバーが指定されていない場合はすべてのメンバー) が宛先に抽出されます。宛先としてディレクトリーを指定する場合、メンバーは別個のファイルに抽出されます。そのように指定しない場合、指定されたメンバーはそれぞれ宛先ファイルに書き込まれ、事実上 1 人のメンバーのみが転送されます。メンバーの宛先ファイルが存在する場合には、ファイル転送で指定した宛先ファイル動作オプションに応じて、ファイルが上書きされます。

関連資料

630 ページの『[ファイルの転送に関するガイドライン](#)』

転送を行うオペレーティング・システムや、バイナリー・モードとテキスト・モードのどちらで転送するかに応じて、予期される動作についてのガイドラインが存在します。

632 ページの『[z/OS でのデータ・セット間の転送](#)』

Managed File Transfer を使用して、z/OS データ・セット間の転送を行うことができます。以下の性質を注意深く検討し、データ・セットの転送が正しく行われるようにしてください。

437 ページの『[fteCreateTransfer: 新規ファイル転送の開始](#)』

fteCreateTransfer コマンドは、コマンド行から新規のファイル転送を作成して開始します。このコマンドは、ファイル転送の即時開始、将来の日時におけるファイル転送のスケジュール、ファイル転送の 1 回以上の繰り返し、および特定の条件に基づくファイル転送の起動を実行できます。

z/OS でのデータ・セット間の転送

Managed File Transfer を使用して、z/OS データ・セット間の転送を行うことができます。以下の性質を注意深く検討し、データ・セットの転送が正しく行われるようにしてください。

Managed File Transfer では、ディスクの場合も磁気テープの場合も、アンカタログ・データ・セットはサポートされていません。既存データ・セットはカタログされる必要があり、新規データ・セットはカタログされます。

以下のケースを検討してください。

z/OS システム間でデータ・セットをコピーまたは移動するときに、宛先が存在しない場合。

デフォルトで、宛先データ・セットはソースと同じ特性で作成されます。宛先データ・セットの属性を指定し、デフォルトの特性をオーバーライドすることができます。これを行う場合、確実に転送を行えるように互換性チェックが実行されます。

z/OS システム間でデータ・セットをコピーまたは移動するときに、宛先が既に存在する場合。

- 宛先データ・セットの属性を指定し、デフォルトの特性をオーバーライドする場合、互換性チェックを実行して、必要な方法で宛先データ・セットにアクセスできるようにします。ただし、以下の属性をオーバーライドすることはできません。
 - 基本データ・セット編成およびタイプ
 - 論理レコード長 (LRECL)
 - ブロック・サイズ (BLKSIZE)

データ・セットを磁気テープに転送する場合。

データ・セットを磁気テープに転送すると、磁気テープの既存データ・セットが置き換えられます。新しいデータ・セットの属性は、転送定義で渡された属性を基にして設定されます。属性は、何も指定されていなければ、ソース・データ・セットと同じ属性またはデフォルト値 (ソースがファイルである場合) に設定されます。既存の磁気テープ・データ・セットの属性は無視されます。

さらに、実行中の宛先エージェントのユーザー ID が、磁気テープをマウントするための正しい権限を持っている必要があります。これを行う方法については、お客様の企業で使用している外部セキュリティー・マネージャーの資料を参照してください。

磁気テープからデータ・セットに転送する場合。

磁気テープ上のデータ・セットにアクセスするためには、実行中のソース・エージェントのユーザー ID が、磁気テープをマウントするための適切な権限を持っている必要があります。これを行う方法については、お客様の企業で使用している外部セキュリティー・マネージャーの資料を参照してください。

データ・セットの互換性

以下のデータ・セットの互換性の性質と制限を検討してください。

レコード・フォーマットおよび長さの違い:

- 可変長フォーマット・レコードは、レコード・データ内で 4 バイト・レコード長のフィールドを使用します。このため、固定のレコードから可変のレコード・データ・セットへの転送では、可変レコード長は固定レコード長に 4 を加えた長さと同じかそれより大きくなります。可変長フォーマットのレコード・データ・セットから固定長フォーマットのレコード・データ・セットへの転送では、固定長フォーマットのレコード・データ・セットは可変レコード長から 4 を引いた長さと同じかそれより大きくなります。

ブロック・サイズの違い:

- 固定形式および可変長フォーマットのレコード・データでは、ブロック・サイズの違いにより、ソース・データ・セットと宛先データ・セットのレイアウトが変わります。
- 不定形式レコードでは、宛先のブロック・サイズがソース・データ・セットのブロック・サイズ以上の場合に、データ・セットを転送できます。
- 不定形式データ・セットでは、ソースのブロック・サイズが宛先のブロック・サイズより大きい場合、転送を行えません。

区分データ・セット (PDS) および区分データ・セット拡張 (PDSE) データ・セット。

以下の性質と制限は PDS と PDSE に対して同じように適用されます。

- PDS または PDSE メンバーを宛先 PDS または PDSE に転送する場合、宛先 PDS または PDSE のメンバーが作成されます。宛先 PDS または PDSE メンバーが既に存在する場合、そのメンバーは上書きされます。PDS または PDSE メンバーを非 PDS または非 PDSE 宛先データ・セットに転送すると、宛先データ・セットが作成され、メンバー・データを格納します。宛先データ・セットが既に存在する場合、そのデータ・セットが上書きされます。

- PDS または PDSE から非 PDS または非 PDSE の宛先に転送しようとする、PDS または PDSE のすべてのメンバーが非 PDSE の宛先に書き込まれます。それ以降メンバーの転送が行われるたびに、転送オプションに応じて、非 PDSE の宛先の以前の内容が上書きされるか失敗します。
- PDS または PDSE から宛先 PDS または PDSE に転送すると、宛先に PDS または PDSE 全体のコピーが作成されます。宛先 PDS または PDSE が既に存在する場合、ソースからのメンバーが追加されます。PDS または PDSE メンバーが宛先に既に存在する場合、そのメンバーが上書きされます。
- 非 PDS または非 PDSE から宛先 PDS または PDSE に転送すると、非 PDS または非 PDSE の内容が PDS または PDSE の新規メンバーとして追加されます。PDS メンバーが既に存在する場合、そのメンバーが上書きされます。新規メンバーに名前を指定しないと、ソース・データ・セットまたは DD 名から名前が生成されます。
- ディスク・スペースが限られているシステムで PDS データ・セットと PDSE データ・セットに転送する場合は、既知の制約事項があります。詳しくは、[MFT の一般的な問題の z/OS セクション](#)を参照してください。
- 注：PDS または PDSE を宛先の PDS または PDSE に転送すると、メンバー情報および統計は保持されません。例えば、PDS として保管されているロード・ライブラリーを転送すると、宛先 PDS はロード・ライブラリーとして使用することができません。

バイナリー転送とテキスト転送

データ・セットのバイナリー転送は、デフォルトのレコード・フォーマット (タイプ=レコード) を使ってデータ・セットから読み取ったとおりの、バイナリー形式のレコード・データと定義されます。データの読み取りと書き込みはレコード単位で行われます。システム・サービスは、必要なレコードおよびブロック変換 (データ・セットのレコードおよびブロックの設定が異なる場合) および必要な ASA およびマシン制御コード変換を実行します。一方のデータ・セットが ASA フォーマット制御文字に対して定義され、もう一方がそれに適していない場合、C/C++ システム・ライブラリー関数の動作を使って通常の制御コードへの変換が実行されます。

世代別データ・グループ (GDG)

Managed File Transfer は、z/OS で、ソース・データ・セットと宛先データ・セットの世代別データ・グループ (GDG) をサポートしています。絶対および相対 GDG 名がサポートされています。新しい世代に書き込む場合には、基本 GDG が既に存在している必要があります。

関連資料

630 ページの『[ファイルの転送に関するガイドライン](#)』

転送を行うオペレーティング・システムや、バイナリー・モードとテキスト・モードのどちらで転送するかに応じて、予期される動作についてのガイドラインが存在します。

652 ページの『[世代別データ・グループ \(GDG\) の転送](#)』

Managed File Transfer は、z/OS で、ソース・データ・セットと宛先データ・セットの世代別データ・グループ (GDG) をサポートしています。絶対および相対 GDG 名がサポートされています。新しい世代に書き込む場合には、基本 GDG が存在している必要があります。

634 ページの『[Connect:Direct ノードとの間のデータ・セット転送](#)』

Connect:Direct ブリッジを使用して、Managed File Transfer エージェントと IBM Sterling Connect:Direct ノードの間でデータ・セット転送を実行できます。転送のソースと宛先のいずれかまたは両方としてデータ・セットを指定できます。

631 ページの『[z/OS と分散システム間のファイルおよびデータ・セットの転送](#)』

Managed File Transfer を使用して、z/OS と分散ファイル・システムの間でファイルやサポートされるデータ・セット・タイプを転送することができます。以下の性質を注意深く検討してください。これは、転送元と転送先のシステムのタイプによって異なります。

z/OS **Connect:Direct ノードとの間のデータ・セット転送**

Connect:Direct ブリッジを使用して、Managed File Transfer エージェントと IBM Sterling Connect:Direct ノードの間でデータ・セット転送を実行できます。転送のソースと宛先のいずれかまたは両方としてデータ・セットを指定できます。

データ・セット名の指定

転送要求で Connect:Direct ノードにあるデータ・セットを指定する場合は、Managed File Transfer エージェント間のデータ・セット転送の場合と同じ構文を使用しますが、2つの変更点があります。

- データ・セット名の接頭部として、Connect:Direct ノード名とコロン (:) を使用する必要があります。構文は次のようになります。

```
cdNode:data_set_name{;attrib1;...;attribN}
```

例えば、Connect:Direct ノード CD_NODE1 が配置されているシステムにある OBJECT.LIB という区分データ・セットを指定する場合は、以下の構文を使用します。

```
CD_NODE1:// 'OBJECT.LIB';RECFM(F,B);BLKSIZE(800);LRECL(80)
```

この例では、3つのオプション属性が RECFM(F,B);BLKSIZE(800);LRECL(80) というテキストで指定されています。

- 指定したデータ・セット名は、単一引用符文字で囲むかどうかにかかわらず、完全修飾データ・セット名として解釈されます。システムによって接頭部が追加されることはありません。エージェントを実行するためのユーザー ID などの接頭部を指定する場合は、データ・セット名の一部として指定する必要があります。この動作は、Managed File Transfer エージェントだけが関わっているデータ・セット転送の動作とは異なります。後者の場合は、指定したデータ・セット名を単一引用符文字で囲んでいないと、システムによって宛先エージェントのデフォルトの高位修飾子の接頭部が追加されます。

この2つの変更点を除けば、データ・セット名とオプション属性を指定するときの構文は、Managed File Transfer エージェント間のデータ・セット転送の場合と同じです。この構文には、以下のようなルールがあります。

- データ・セット名の接頭部として、2つのスラッシュ文字 (//) を使用する必要があります。
- データ・セットの属性を指定する場合は、データ・セット名の後にそれらの属性をセミコロンで区切って指定します。属性を指定するときには、BPXWDYN に適した *key(value)* という形式にする必要があります。

転送要求でデータ・セットを指定する方法の詳細については、[437 ページの『fteCreateTransfer: 新規ファイル転送の開始』](#)と [421 ページの『fteCreateTemplate: 新規ファイル転送テンプレートの作成』](#)を参照してください。

転送要求で使用するパラメーター

Connect:Direct ノードにあるデータ・セットがかかっているほとんどの転送要求では、Managed File Transfer エージェントだけが関わっているデータ・セット転送の場合と同じ要領でソースと宛先のデータ・セットを指定できます。**fteCreateTransfer** コマンドまたは **fteCreateTemplate** コマンドで、**source_specification**、**-ds**、**-dp** の各パラメーターを使用します。この構文は、以下のような状況に対応しています。

- 転送にかかっているすべてのエージェントが IBM WebSphere MQ File Transfer Edition 7.0.4 以降の場合
- ソース・エージェントが Connect:Direct ブリッジ・エージェントであり (したがって IBM WebSphere MQ File Transfer Edition 7.0.4 以降であり)、宛先エージェントが IBM WebSphere MQ File Transfer Edition 7.0.3 以前の場合

宛先エージェントが Connect:Direct ブリッジ・エージェントで、ソース・エージェントが IBM WebSphere MQ File Transfer Edition 7.0.3 以前の場合は、転送要求を以下のように変更する必要があります。

- 転送の宛先として順次データ・セットまたは区分データ・セット (PDS) のメンバーを指定する場合は、**-df** パラメーターを使用します。
- 転送の宛先として PDS を指定する場合は、**-dd** パラメーターを使用します。

ソース・エージェントが IBM WebSphere MQ File Transfer Edition 7.0.4 以降になっている転送で、通常の **-ds** パラメーターと **-dp** パラメーターの代わりにこの構文を使用することもできます。例えば、ソース・エージェントが IBM WebSphere MQ File Transfer Edition 7.0.3 以前になる状況が部分的に存在する場合に、すべての状況で使用する構文を統一するのであれば、**-df** パラメーターと **-dd** パラメーターを使用します。

注：転送の宛先が PDS で、宛先エージェントが Connect:Direct ブリッジ・エージェントの場合は、**-de** パラメーターで `overwrite` という値を指定する必要があります。

データ・セットの属性の指定

データ・セットの一部の属性は、Managed File Transfer によって設定され、Connect:Direct の **COPY** プロセスにパラメーターとして渡されます。さらに、転送要求で適切な BPXWDYN キーを使用して、特定の属性を指定することもできます。Connect:Direct のプロパティに対応しているキーは、Connect:Direct ブリッジによって、Connect:Direct で必要な形式に変換されます。例えば、データ・セットの指定が `CD_NODE1:/'OBJECT.LIB';RECFM(F,B);BLKSIZE(800);LRECL(80)` になっている場合は、`RECFM(F,B);BLKSIZE(800);LRECL(80)` という属性が `DCB=(RECFM=FB,BLKSIZE=800,LRECL=80)` に変換されます。

この 2 つのタイプのパラメーターの対応関係に関する詳細と、Connect:Direct の転送で利用できる BPXWDYN キーの詳細については、637 ページの『[Connect:Direct プロセス・ステートメント・パラメーターと BPXWDYN キーのマッピング](#)』を参照してください。すべての BPXWDYN キーに Connect:Direct プロセス・パラメーターが対応しているわけではなく、すべての Connect:Direct プロセス・パラメーターに BPXWDYN キーが対応しているわけでもありません。

その他の考慮事項

- 転送の宛先が Connect:Direct ノードにある区分データ・セットの場合は、区分データ・セットを作成してから転送を実行してください。Connect:Direct ノードによって区分データ・セットが作成されることはありません。

関連概念

230 ページの『[Connect:Direct ブリッジ](#)』

既存の IBM Sterling Connect:Direct ネットワークとの相互間で、ファイルを転送することができます。Managed File Transfer のコンポーネントである Connect:Direct ブリッジを使用して、MFT と IBM Sterling Connect:Direct の間でファイルを転送します。

関連タスク

 [z/OS 上の Connect:Direct ノードへのデータ・セットの転送](#)

Windows または Linux のシステムにある Connect:Direct ブリッジを使用して、z/OS 上の Managed File Transfer エージェントから z/OS 上の Connect:Direct ノードにデータ・セットを転送できます。

関連資料

632 ページの『[z/OS でのデータ・セット間の転送](#)』

Managed File Transfer を使用して、z/OS データ・セット間の転送を行うことができます。以下の性質を注意深く検討し、データ・セットの転送が正しく行われるようにしてください。

437 ページの『[fteCreateTransfer: 新規ファイル転送の開始](#)』

fteCreateTransfer コマンドは、コマンド行から新規のファイル転送を作成して開始します。このコマンドは、ファイル転送の即時開始、将来の日時におけるファイル転送のスケジュール、ファイル転送の 1 回以上の繰り返し、および特定の条件に基づくファイル転送の起動を実行できます。

421 ページの『[fteCreateTemplate: 新規ファイル転送テンプレートの作成](#)』

fteCreateTemplate コマンドは、将来の使用のために保持できるファイル転送テンプレートを作成します。必須パラメーターは **-tn** (*template_name*) パラメーターのみです。その他のパラメーターはすべてオプションです。ただし、ソース・ファイル仕様を指定する場合には、宛先ファイルも提供する必要があります。同様に、宛先ファイルを指定する場合には、ソース・ファイル仕様も指定する必要があります。

348 ページの『[ダブルスラッシュで指定される Connect:Direct ファイル・パス](#)』

ファイル転送の一部として、Connect:Direct ノードにあるファイルを、ダブルスラッシュ (//) で始まるファイル・パスを使用して指定した場合、そのファイルはデータ・セットとして扱われます。

z/OS **Connect:Direct プロセス・ステートメント・パラメーターと BPXWDYN キーのマッピング**

ソースまたは宛先のいずれかが Connect:Direct ノードになっている状況でデータ・セットの転送要求を送信すると、指定したサポート対象の BPXWDYN キーが、Connect:Direct プロセスによって受け入れられる形式に変換されます。

IBM Sterling Connect:Direct 処理ステートメントについて詳しくは、「[Connect:Direct Process Language Reference Guide](#)」を参照してください。

表 58. *Connect:Direct* の **COPY** ステートメントのパラメーターと、*Managed File Transfer* で使用する **BPXWDYN** キーとの対応関係

Connect:Direct COPY ステートメントのパラメーター	BPXWDYN キー
DSN	DSN (データ・セットの転送で有効)。このキーを指定すると、転送要求に組み込まれているソース・ファイルまたは宛先ファイルの指定に基づいて、Managed File Transfer によって割り当てられるパラメーター値がオーバーライドされます。
FILE	データ・セットの場合は、対応関係がありません。
PNODE	対応関係がありません。転送の 1 次ノードは、Managed File Transfer によって識別されます。このパラメーターの値を指定しようとすると、エラーが生成されます。
SNODE	対応関係がありません。転送の 2 次ノードは、Managed File Transfer によって識別されます。このパラメーターの値を指定しようとすると、エラーが生成されます。
DCB	DCB のサブパラメーターの対応関係を参照
DISP	COPY From ステートメントの DISP のサブパラメーターの対応関係 と COPY To ステートメントの DISP のサブパラメーターの対応関係 を参照
RESGDG	対応関係がありません
LABEL	LABEL のサブパラメーターの対応関係を参照
MSVGP	対応関係がありません
単位	単位
VOL	VOL のサブパラメーターの対応関係を参照
ALIAS	対応関係がありません
EXCLUDE	対応関係がありません
PDS.DIR	対応関係がありません。このプロセス・パラメーターの値は、Managed File Transfer によって N に設定されるので、ディレクトリーに含まれているユーザー関連情報は送信されません。

表 58. Connect:Direct の **COPY** ステートメントのパラメーターと、Managed File Transfer で使用する **BPXWDYN** キーとの対応関係 (続き)

Connect:Direct COPY ステートメントのパラメーター	BPXWDYN キー
REPLACE NOREPLACE	対応する BPXWDYN はありません。宛先データ・セットが宛先システムに既に存在している場合の動作は、転送要求の -de (destination_file_behavior) パラメーターの値で定義します。宛先データ・セットが既に存在する場合の Managed File Transfer のデフォルトの動作の詳細については、632 ページの『z/OS でのデータ・セット間の転送』を参照してください。
SELECT	対応する BPXWDYN はありません。コピー対象として選択するデータ・セット・メンバーは、転送要求のソース・ファイルの指定で定義します。
BUFND	対応関係がありません
IOEXIT	対応関係がありません
DATAEXIT	対応関係がありません
SYSOPTS	SYSOPTS のサブパラメーターの対応関係を参照
タイプ	対応関係がありません
AVGREC	対応関係がありません
DATACLAS	DATACLAS
DSNTYPE	DSNTYPE。このキーで PDS の値を指定すると、Managed File Transfer によって割り当てられるパラメーター値 (LIBRARY) がオーバーライドされます。その他の値 (EXTPREF、EXTREQ、BASIC、LARGE) には、対応関係がありません。それらのサポートされていない値のいずれかを指定すると、エラーが生成されます。順次データ・セットの場合に PDS または LIBRARY を指定すると、エラーが生成されます。
KEYLEN	対応関係がありません
KEYOFF	対応関係がありません
LIKE	LIKE
LRECL	対応関係がありません
MGMTCLAS	MGMTCLAS
RECORG	対応関係がありません
SECMODEL	対応関係がありません
STORCLAS	STORCLAS
SPACE	SPACE のサブパラメーターの対応関係を参照
SYSOUT	対応関係がありません
CKPT	対応関係がありません
COMPRESS	対応関係がありません
SECURE	対応関係がありません

表 59. Connect:Direct の **COPY** ステートメントの **DCB** パラメーターのサブパラメーターと、Managed File Transfer で使用する BPXWDYN キーとの対応関係

DCB パラメーターのサブパラメーター	BPXWDYN キー
model-file-name	対応関係がありません
BLKSIZE	BLKSIZE
NCP	BUFNO
DEN	対応関係がありません
DSORG	DSORG
KEYLEN	対応関係がありません
LIMCT	対応関係がありません
LRECL	LRECL
OPTCD	対応関係がありません
RECFM	RECFM
RKP	対応関係がありません
TRTCH	TRTCH

表 60. Connect:Direct の **COPY From** ステートメントの **DISP** パラメーターのサブパラメーターと、Managed File Transfer で使用する BPXWDYN キーとの対応関係

DISP パラメーターのサブパラメーター (COPY From ステートメントの場合)	BPXWDYN キー	詳細
[OLD SHR]	[OLD SHR]	転送前のデータ・セットの状況を指定します。Managed File Transfer は、このサブパラメーターを SHR に設定します。
[KEEP DELETE]	[KEEP DELETE] または PATHDISP	転送が正常に完了した後のデータ・セットの状況を指定します。Managed File Transfer によって設定される値は、 -sd パラメーターで定義するソース・ファイルの処置によって決まります。
[KEEP DELETE]	[KEEP DELETE] または PATHDISP	転送が異常な状態で完了した後のデータ・セットの状況を指定します。Managed File Transfer は、このサブパラメーターを KEEP に設定します。

表 61. Connect:Direct の **COPY To** ステートメントの **DISP** パラメーターのサブパラメーターと、Managed File Transfer で使用する BPXWDYN キーとの対応関係

DISP パラメーターのサブパラメーター (COPY To ステートメントの場合)	BPXWDYN キー	詳細
[NEW OLD MOD RPL SHR]	[NEW OLD MOD SHR]	転送前のデータ・セットの状況を指定します。Managed File Transfer によって設定される値は、転送要求の -de (destination_file_behavior) パラメーターの値によって決まります。宛先データ・セットが存在しなければ、サブパラメーター値は NEW になります。データ・セットが存在すれば、サブパラメーター値は RPL になります。Managed File Transfer では、 RPL というキーを転送要求で指定できません。
[KEEP CATLG]	[KEEP CATLOG] または PATHDISP	このサブパラメーターは、転送が正常に完了した後のデータ・セットの状況を指定します。Managed File Transfer によって CATLOG に設定されます。
[KEEP CATLG DELETE]	[KEEP DELETE] または PATHDISP	このサブパラメーターは、転送が異常な状態で完了した後のデータ・セットの状況を指定します。Managed File Transfer によって DELETE に設定されます。

表 62. Connect:Direct の **COPY** ステートメントの **LABEL** パラメーターのサブパラメーターと、Managed File Transfer で使用する BPXWDYN キーとの対応関係

LABEL パラメーターのサブパラメーター (COPY ステートメントの場合)	BPXWDYN キー	詳細
file-sequence-number	SEQUENCE	
[SL AL BLP LTM NL]	LABEL(type)	type の有効な値は、NL、SL、NSL、SUL、BLP、LTM、AL、AUL です。Connect:Direct では、これらの値のサブセットが有効です。Connect:Direct でサポートされていない値を指定すると、Connect:Direct によってエラー・メッセージが生成されます。
[PASSWORD NOPWREAD]	対応関係がありません	
[IN OUT]	対応関係がありません	
[RETPD EXPDT]	RETPD	EXPDT がサポートされない

表 63. Connect:Direct の COPY ステートメントの VOL パラメーターのサブパラメーターと、Managed File Transfer で使用する BPXWDYN キーとの対応関係	
VOL パラメーターのサブパラメーター (COPY ステートメントの場合)	BPXWDYN キー
PRIVATE	対応関係がありません
RETAIN	対応関係がありません
volume-sequence-no	対応関係がありません
volume-count	MAXVOL
SER	VOL
REF	対応関係がありません

表 64. Connect:Direct の COPY ステートメントの SYSOPTS パラメーターのサブパラメーターと、Managed File Transfer で使用する BPXWDYN キーとの対応関係	
SYSOPTS パラメーターのサブパラメーター (COPY ステートメントの場合)	BPXWDYN キー
DBCS	対応関係がありません
コード・ページ	値は Managed File Transfer の転送オプションによって異なります。詳しくは、643 ページの『 MFT でのテキスト・ファイルの転送 』を参照してください。
DATATYPE	対応関係がありません。データ・セットとの間のテキスト転送の場合は、Managed File Transfer によってこの値が TEXT に設定され、そうでない場合は BINARY に設定されます。
XLATE	対応関係がありません。Managed File Transfer は、 DATATYPE の値が TEXT の場合、この値を NO に設定します。
STRIP.BLANKS	対応関係がありません。Managed File Transfer は、 DATATYPE の値が TEXT の場合にこの値を YES に設定します。
PERMISS	対応関係がありません
PRECOMP	対応関係がありません
UNIQUE	対応関係がありません
SYSOUT	対応関係がありません

表 65. Connect:Direct の COPY ステートメントの SPACE パラメーターのサブパラメーターと、Managed File Transfer で使用する BPXWDYN キーとの対応関係	
SPACE パラメーターのサブパラメーター (COPY ステートメントの場合)	BPXWDYN キー
CYL	CYL
TRK	TRACKS
blk	BLOCKS
av-rec-len	対応関係がありません

表 65. Connect:Direct の **COPY** ステートメントの **SPACE** パラメーターのサブパラメーターと、Managed File Transfer で使用する BPXWDYN キーとの対応関係 (続き)

SPACE パラメーターのサブパラメーター (COPY ステートメントの場合)	BPXWDYN キー
prim, [sec], [dir]	SPACE(prim[,sec]), DIR
RLSE	RELEASE
CONTIG	対応関係がありません
ROUND	対応関係がありません

関連概念

230 ページの『Connect:Direct ブリッジ』

既存の IBM Sterling Connect:Direct ネットワークとの相互間で、ファイルを転送することができます。Managed File Transfer のコンポーネントである Connect:Direct ブリッジを使用して、MFT と IBM Sterling Connect:Direct の間でファイルを転送します。

関連タスク

z/OS z/OS 上の Connect:Direct ノードへのデータ・セットの転送

Windows または Linux のシステムにある Connect:Direct ブリッジを使用して、z/OS 上の Managed File Transfer エージェントから z/OS 上の Connect:Direct ノードにデータ・セットを転送できます。

関連資料

Connect:Direct ノードとの間のデータ・セット転送

Connect:Direct ブリッジを使用して、Managed File Transfer エージェントと IBM Sterling Connect:Direct ノードの間でデータ・セット転送を実行できます。転送のソースと宛先のいずれかまたは両方としてデータ・セットを指定できます。

z/OS **MFT** で使用できない BPXWDYN のプロパティ

いくつかの BPXWDYN オプションは、**fteCreateTemplate** コマンド、**fteCreateTransfer** コマンド、または **agent.properties** ファイル内の **bpxwdynAllocAdditionalOptions** プロパティを使用するときに指定してはなりません。

いくつかの BPXWDYN オプションはエージェントで使用されるため、またはサポートされていないため、Managed File Transfer では指定できません。これらのオプションを使用すると、予測不能な動作が起こる可能性があります。該当するオプションを以下の表にリストします。

BPXWDYN オプション	説明
DA DSN	割り振るデータ・セット名を指定します。
FI DD	割り振る DDName を指定します。
FILEDATA	データをテキストとして扱うかバイナリーとして扱うかを、順次アクセス・メソッド・サービスに指定します。
OLD SHR MOD NEW SYSOUT	データ・セットの状況を指定します。
REUSE	機能が実行される前に、指定されたデータ・セットが解放されることを指定します。
HOLD	ユーザーまたはオペレーターにより出力データ・セットが解放されるまで、その出力データ・セットを保留することを指定します。
KEEP DELETE CATALOG UNCATALOG	解放されたデータ・セットの後処理を指定します。
RECORG(LS)	VSAM 線形データ・セットを作成します。
MSG	割り振りメッセージの送信先を指定します。 注: このオプションは使用可能ですが、Managed File Transfer は、このオプションを使用してエラー情報を転送ログに送るため、これを使用すると予測不能な動作が起こる場合があります。

MFTでのテキスト・ファイルの転送

テキスト・ファイルの転送には、あるコード・ページから別のコード・ページへの、ファイルのコード・ページの変換が伴います。テキスト・ファイルの転送には、システム間の CRLF (復帰改行) 文字の変換も伴います。このトピックでは Managed File Transfer のテキスト・ファイル転送動作をまとめています。

別の指定をしなければ、変換は、ファイルのソース・システムのデフォルト・コード・ページから、その宛先システムのデフォルト・コード・ページへと行われます。また、テキスト・ファイルの転送では改行の変換が実行されます。これは、宛先ファイルの改行文字が宛先プラットフォームに固有のものになることを意味します。ソース・ファイルを読み取るため、および宛先ファイルに書き込むために使用するコード・ページを指定することにより、システムのデフォルト・コード・ページの使用をオーバーライドすることができます。また、宛先ファイルに使用する行末文字シーケンスを指定することもできます。詳しくは、437 ページの『fteCreateTransfer: 新規ファイル転送の開始』および 161 ページの『転送定義ファイルの使用』のトピックを参照してください。

テキスト・ファイルの転送では、コード・ページ間の単純なコード・ポイント置換が実行されます。テキスト・ファイルの転送では、データの複雑な転送や変換 (例えば、BIDI データまたはテキスト・シェーピングのビジュアル形式と論理形式の間の変換) は実行されません。

表 66. すべてのプラットフォームにおけるテキスト・ファイルの転送の動作

領域	デフォルトの動作	この動作を変更できるか
ソース・ファイルのエンコード	ソース・プラットフォームのエンコード	Yes ソース・ファイル・エンコードを指定する場合に、そのソースがデータ・セットのときは、エンコードは EBCDIC コード・ページである必要があります。そうでない場合、転送は失敗します。同様に、宛先がデータ・セットの場合、宛先エンコードは EBCDIC コード・ページである必要があります。
ソース・ファイルの行末文字のシーケンス	シングル (LF) または (CRLF) シーケンスを宛先の行末文字シーケンスに変換する	No
宛先ファイルのエンコード	宛先プラットフォームのエンコード	Yes ソース・ファイル・エンコードを指定する場合に、そのソースがデータ・セットのときは、エンコードは EBCDIC コード・ページである必要があります。そうでない場合、転送は失敗します。同様に、宛先がデータ・セットの場合、宛先エンコードは EBCDIC コード・ページである必要があります。
宛先ファイルの行末文字のシーケンス	宛先プラットフォームの EOL	Yes

領域	デフォルトの動作	この動作を変更できるか
ソースまたは宛先における、マップできない文字または誤った形式の文字に対するテキスト置換文字シーケンス	マップできない文字または誤った形式の文字が存在する場合に転送が失敗することを意味するブランク。textReplacementCharacterSequence プロパティを使用して、置換テキストを指定することができます。これについては、550 ページの『MFT agent.properties ファイル』で説明しています。	Yes

z/OS データ・セット

z/OS

データ・セットのレコードにテキスト・モードでアクセスした場合、各レコードは単一行で表されます。改行文字はレコードに存在しませんが、ASA フォーマットのデータ・セットの場合、改行 (あるいは他の制御文字) を表す ASA フォーマット制御コード文字が設定されます。終端の改行文字が付いたテキスト行がレコードに書き込まれると、改行文字は自動的に除去されるか、あるいは必要に応じて適切な ASA 制御コードが設定されます。レコードを読み取るときには、返されるデータに改行文字が自動的に付加されます。ASA 形式のデータ・セットの場合、レコードの ASA 制御コードに応じて、この文字は複数の改行または 1 つの用紙送りになります。

さらに、固定形式のデータ・セットの場合は、レコードが読み取られたときに、レコードの最後の文字 (スペース文字以外の文字) の後に改行が追加されます。その結果、固定形式のデータ・セットは、テキストの格納に適した形式になります。

領域	デフォルトの動作	この動作を変更できるか
行の最大長	宛先データ・セットの LRECL または BLKSIZE 設定 (必要に応じて)	No
長い行の折り返し	折り返し。必要に応じて、行は複数のレコードおよびブロックに分割されます。	No

Managed File Transfer エージェントが実行される場合、環境変数 `_EDC_ZERO_RECLEN` は常に "Y" に設定されます。この設定によって、Managed File Transfer のテキスト転送の動作は、可変ブロックと固定ブロックのデータ・セットを転送する FTP と同じ動作になります。ただし、不定形式のデータ・セットの場合、Managed File Transfer は、シングル・スペース行を空の行に変換し、空の行を保持します。FTP は空の行をシングル・スペース行に変換し、シングル・スペース行を保持します。Managed File Transfer の動作と FTP の動作の違いを表 3 にまとめます。

データ・セットの形式によって、テキストの各行をレコードに書き込む方法も決まります。ASA 形式以外のデータ・セットの場合、改行文字および復帰文字はレコードに書き込まれません。ASA 形式のデータ・セットの場合、各レコードの最初のバイトは、状況に応じて行末、用紙送り、およびその他のコードを表す ASA 制御コードになります。ASA 制御コードは各レコードの最初にあるため、ソース・テキスト・ファイルが改行文字シーケンスで始まらない場合、ブランク (「」) の ASA 制御文字シーケンス (改行と同等) が挿入されます。つまり、ASA データ・セットがファイルに転送されると、ファイルの先頭にブランク行が表示されます。

データ・セット・フォーマット	ファイル内の元のテキスト行	データ・セットのレコード	データ・セットのレコードの読み取り	FTP の読み取りの動作
固定ブロック	空の行	スペース埋め込みレコード	空の行	MFT と同じ

表 68. 各種のデータ・セットに関する Managed File Transfer の動作 (続き)

データ・セット・フォーマット	ファイル内の元のテキスト行	データ・セットのレコード	データ・セットのレコードの読み取り	FTP の読み取りの動作
固定ブロック	シングル・スペース	スペース埋め込みレコード	空の行	MFT と同じ
可変ブロック	空の行	空のレコード	空の行	MFT と同じ
可変ブロック	シングル・スペース	シングル・スペース・レコード	シングル・スペース	MFT と同じ
未定義	空の行	シングル・スペース・レコード	空の行	シングル・スペース
未定義	シングル・スペース	シングル・スペース・レコード	空の行	シングル・スペース

関連資料

630 ページの『ファイルの転送に関するガイドライン』

転送を行うオペレーティング・システムや、バイナリー・モードとテキスト・モードのどちらで転送するかに応じて、予期される動作についてのガイドラインが存在します。

645 ページの『Connect:Direct と MFT の間のテキスト・ファイル転送』

テキストの転送には、あるコード・ページから別のコード・ページへの、ファイルのコード・ページの変換が伴います。テキスト・ファイルの転送には、システム間の CRLF (復帰改行) 文字の変換も伴います。このトピックでは、Managed File Transfer Agent と Connect:Direct ノードの間のテキスト・ファイル転送の動作をまとめます。

683 ページの『MFT で使用できるコード・ページ』

この参照トピックでは、Managed File Transfer がサポートする各種のプラットフォームでのテキスト・ファイル変換に使用可能な、すべての文字エンコード・フォーマットをリストしています。

Connect:Direct と MFT の間のテキスト・ファイル転送

テキストの転送には、あるコード・ページから別のコード・ページへの、ファイルのコード・ページの変換が伴います。テキスト・ファイルの転送には、システム間の CRLF (復帰改行) 文字の変換も伴います。このトピックでは、Managed File Transfer Agent と Connect:Direct ノードの間のテキスト・ファイル転送の動作をまとめます。

Managed File Transfer のテキスト転送の動作については、[643 ページの『MFT でのテキスト・ファイルの転送』](#)を参照してください。

- Connect:Direct ブリッジ・ノードと、転送の宛先として使用する Connect:Direct ノードのネットワーク・マップに、正しいプラットフォームの記述を組み込むようにします。
 - Connect:Direct ブリッジ・ノードが Windows システムにある場合は、ネットワーク・マップに含まれているリモート・ノードごとに、「オペレーティング・システム」リストから正しい値を選択するようにしてください。
 - リモート・ノードが Windows システムにある場合は、「Windows」を選択します。
 - リモート・ノードが UNIX または Linux のシステムにある場合は、「UNIX」を選択します。
 -  リモート・ノードが z/OS システムにある場合は、「OS/390」を選択します。

Connect:Direct ブリッジは、その他のオペレーティング・システムのリモート・ノードへの転送には対応していません。

- Connect:Direct ブリッジ・エージェントの構成ディレクトリーにある `ConnectDirectNodeProperties.xml` ファイルを使用して、ファイル転送にかかわるリモート・ノードごとに、リモート Connect:Direct ノードのオペレーティング・システム・タイプを指定します。詳細については、[145 ページの『ConnectDirectNodeProperties.xml ファイルを構成して、リモート](#)

[Connect:Direct ノードに関する情報を組み込む](#) および 588 ページの『[Connect:Direct ノードのプロパティ・ファイルのフォーマット](#)』を参照してください。

Connect:Direct では、ネットワーク・マップの情報に基づいて、使用する行末を判別します。

- 転送の宛先が Managed File Transfer Agent の場合は、そのエージェントが行末の変換を実行します。
- 転送の宛先が Connect:Direct ノードの場合は、Connect:Direct ブリッジ・エージェントが行末の変換を実行します。

関連資料

[643 ページの『MFT でのテキスト・ファイルの転送』](#)

テキスト・ファイルの転送には、あるコード・ページから別のコード・ページへの、ファイルのコード・ページの変換が伴います。テキスト・ファイルの転送には、システム間の CRLF (復帰改行) 文字の変換も伴います。このトピックでは Managed File Transfer のテキスト・ファイル転送動作をまとめています。

[347 ページの『Connect:Direct ノードとの間のテキスト転送でデータが正しく変換されない場合の対策』](#)

MFT エージェントと Connect:Direct ノードの間でテキスト・モードでファイルを転送すると、コード・ページと行末文字の変換が実行されます。転送では、Connect:Direct ブリッジ・ノードのネットワーク・マップにあるオペレーティング・システム情報に基づいて、リモート・ノードの行末文字が判別されます。ネットワーク・マップにある情報が不正確であれば、行末文字の変換が正しく実行されない可能性があります。

プロトコル・ブリッジ・エージェントとの間のファイル転送

プロトコル・ブリッジ・エージェントを使用すると、Managed File Transfer ネットワークの外部にある FTP ファイル・サーバーまたは SFTP ファイル・サーバーとの間でファイルを転送できます。

プロトコル・ブリッジを使用してファイルを転送する場合、ブリッジは、転送するファイルが格納されているソースまたは宛先ディレクトリーを読み取るための権限を持っている必要があります。例えば、実行権限 (d--x-x-x) のみを持つディレクトリー /home/fte/bridge からファイルを転送すると、このディレクトリーから試行した転送は、以下のエラー・メッセージを表示して失敗します。

```
BFGBR0032E: Attempt to read filename from the protocol file server has failed with server error 550
Failed to open file.
```

ファイル転送中は、ファイルは通常、一時ファイルとして転送先に書き込まれ、転送が完了した時点でリネームされます。ただし、転送先が、書き込みを制限して構成されている (ユーザーはプロトコル・ファイル・サーバーにファイルをアップロードできるが、それらのアップロードされたファイルは決して変更できず、ユーザーが書き込めるのは事実上一度だけである) プロトコル・ファイル・サーバーの場合、転送されるファイルは転送先に直接書き込まれます。そのため、転送中に問題が発生した場合、一部しか書き込まれていないファイルが、転送先プロトコル・ファイル・サーバーに残りますが、Managed File Transfer は、これらのファイルを削除も編集もできません。このシチュエーションでは、転送は失敗します。

プロトコル・ブリッジ・エージェントに加え、Managed File Transfer ネットワークに別のエージェントがあることを確認します。プロトコル・ブリッジ・エージェントは、FTP または SFTP サーバーに対してのみのブリッジであり、転送されたファイルをローカル・ディスクに書き込むことはありません。ファイルを FTP または SFTP サーバーとの間で転送する場合は、プロトコル・ブリッジ・エージェントを (FTP または SFTP サーバーを表す) ファイル転送の宛先またはソースとして使用し、別の標準エージェントを対応するソースまたは宛先として使用する必要があります。

SFTP ファイル・サーバー上に新規ディレクトリーを作成する必要がある管理対象転送要求

Managed File Transfer プロトコル・ブリッジ・エージェントはサード・パーティーの JSch ライブラリーを使用して、SFTP プロトコルでファイル・サーバーと通信します。プロトコル・ブリッジ・エージェントがファイル・サーバー上に存在しないディレクトリーにファイルを転送しようと試み、JSch がそのディレクトリーを作成するために要求された SFTP 操作を実行できない場合、プロトコル・ブリッジ・エージェントがファイル・サーバーにログインしたときのユーザーがそのための権限を持っていないため、JSch はプロトコル・ブリッジ・エージェントに例外をスローします。その後、プロトコル・ブリッジ・エージェントは、管理対象転送に「失敗」のマークを付け、補足メッセージを生成します。JSch から失敗に関する詳細な情報が提供された場合、プロトコル・ブリッジ・エージェントは、その情報を次のように補足メッセージに含めます。

BFGTR0072E: 次の例外のため、転送を完了できませんでした:

BFGBR0119E: ブリッジ・エージェントがディレクトリー *directory name* を作成できませんでした。理由: *message from JSch exception*

V 9.0.0.9 IBM MQ 9.0.0 Fix Pack 9 以降、失敗に関する詳しい情報が JSch 例外に含まれていない場合は、プロトコル・ブリッジ・エージェントが以下の補足メッセージを生成します。

BFGTR0072E: 次の例外のため、転送を完了できませんでした:

BFGBR0209E: ブリッジ・エージェントがディレクトリー *directory name* を作成できませんでした

関連概念

214 ページの『プロトコル・ブリッジ』

プロトコル・ブリッジを使用すれば、Managed File Transfer (MFT) ネットワークから、MFT ネットワークの外部(ローカル・ドメインとリモート・ロケーションの両方)にあるファイル・サーバーに格納されているファイルにアクセスできます。このファイル・サーバーでは、FTP、FTPS、または SFTP ネットワーク・プロトコルを使用できます。それぞれのファイル・サーバーで少なくとも 1 つの専用エージェントが必要です。この専用エージェントは、プロトコル・ブリッジ・エージェントとして知られています。ブリッジ・エージェントは、複数のファイル・サーバーと相互作用できます。

IBM i IBM i システム間のファイル転送

テキスト・モードで Managed File Transfer を使って IBM i システム間のファイル転送を行い、ファイル内のデータを変換する場合は、このトピックの情報を検討してください。

IBM i システム上の各ファイルは、ファイルのデータ・エンコードを識別するコード化文字セット ID (CCSID) 値でタグ付けされます。例えば、EBCDIC データが含まれているファイルは 037 という CCSID 値を持ち、ASCII データが含まれているファイルは 819 という CCSID 値を持つ可能性があります。

Managed File Transfer は、テキスト・モードの転送では、ソース・ファイルと宛先ファイルの間のファイル・エンコードに違いがある場合にデータを変換します。ただし、Managed File Transfer は現在、IBM i システム上のファイルと関連付けられている CCSID タグを無視します。代わりに、ソース・エージェントと宛先エージェントを実行している JVM の JVM ファイル・エンコード・プロパティを使用します。このプロパティのデフォルト値はローカルに基づいています(ただし、IBM i システム上のこのデフォルト値は、SystemDefault.properties ファイルを使用してオーバーライドできます。このファイルについては、セクション 647 ページの『SystemDefault.properties ファイルの file.encoding レコードの変更』で説明されています。) このデフォルトの実装を使用する場合、テキスト・モードでファイルを転送するエージェントは、ファイル・エンコードが異なるテキスト・ファイル进行处理する際に、行えることが限られてしまいます。例えば、EBCDIC テキストを含むファイルと ASCII テキストを含むファイルを、適切な(つまり、EBCDIC または ASCII) ファイル・エンコードのオーバーライドを実施してエージェントの停止および再始動を行わずに転送する場合、同じエージェントを使用することができません。IBM i V6R1 システムでは、WRKJVMJOB のオプション 7、「Current® Java システム・プロパティの表示」を使用して、エージェント・ジョブを実行中の JVM のファイル・エンコード値を確認できます。(WRKJVMJOB コマンドは IBM i V5R4 システムには存在しません)。

ファイル・エンコードが異なるテキスト・ファイルを Managed File Transfer を使って転送する計画の場合、複数のエージェント、およびそれらのエージェントを開始する複数のユーザーを作成することを検討してください。そのタイプのデータの転送準備が整っているエージェントをそれぞれ固有のエンコードが持つようにするためです。

例えば、CCSID 値が 037 の EBCDIC テキストを含んでいるファイルを IBM i システム(ソース)から別の IBM i V6R1 システム(宛先)に転送し、宛先でファイルの内容を、CCSID 値が 819 の ASCII テキストに変換する場合は、以下のステップを実行します。

1. JVM ファイル・エンコードが Cp037 のソース・エージェントを選択します。
2. JVM ファイル・エンコードが ISO8859_1 の宛先エージェントを選択します。
3. テキスト・モードの転送、および必要に応じてその他の仕様を選択します。

SystemDefault.properties ファイルの file.encoding レコードの変更

特定のエンコード用エージェントを実行する JVM を使用可能にするには、以下のステップを実行します。

1. IBM i システムで実行されるエージェントを開始するユーザーを決定します。これは、Managed File Transfer ファイル転送要求を処理するエージェントです。

必要に応じて、そのユーザーのホーム・ディレクトリーに `SystemDefault.properties` ファイルを作成します。例えば、自らエージェントを開始する場合、Qshell を使用して以下のコマンドを実行します。

```
touch -C 819 /home/your_userID/SystemDefault.properties
```

- 必要に応じて、Qshell を使用して `/qibm/proddata/mqm/bin/fteStopAgent` コマンドを実行することによってエージェントを停止します。
- ステップ 1 で説明されている `SystemDefault.properties` ファイルを更新し、ファイルに以下のようなレコードが含まれるようにします。

```
file.encoding=java_encoding
```

ここで、Java エンコードは、ファイルに含まれているデータのタイプに対応し、表 `File.encoding` の値と `System i5® CCSID` の `file.encoding` 値と一致します。

- ステップ 1 で指定したユーザーは、以下のステップを実行する必要があります。
 - IBM i V5R4 の場合のみ: `QIBM_PASE_DESCRIPTOR_STDIO` 環境変数 (*JOB の有効範囲) を、EBCDIC ファイル・エンコードを使用している場合は「B」に、ASCII エンコードを使用している場合は「T」に追加します。以下に例を示します。

```
ADDENVVAR ENVVAR('QIBM_PASE_DESCRIPTOR_STDIO') VALUE('B') REPLACE(*YES)
```

- Qshell がアクティブになっている場合、**F3=Exit** を押して Qshell を終了します。
- 必要に応じて、Qshell を開始し、`/qibm/proddata/mqm/bin/fteStartAgent` コマンドを実行して、エージェントを再始動します。

エージェントを実行している JVM のファイル・エンコードが変更された場合、エージェント・ログがそのエンコードとともに書き込まれます。エージェント・ログの内容を読み取る場合は、そのエンコードに対応したビューアーを使用する必要があります。

データ変換のための転送定義の使用

ファイルを転送するときにデータを変換する別の方法は、ファイル・エンコードを指定した転送定義を作成することです。あるいは、**-sce** および **-dce** パラメーターを **fteCreateTransfer** コマンドで使用します。宛先が IBM i システムであるときにこれらのパラメーターを使用する場合は、ファイルの CCSID タグが正しくならない可能性があります。そのため、IBM i システムにあるファイルが関係するデータ変換の制御は、前のセクションで説明したように、`SystemDefault.properties` を使ったアプローチで行うことをお勧めします。

プロトコル・ブリッジの制約事項

IBM i では、プロトコル・ブリッジ・エージェントを使って SFTP サーバーとの間の EBCDIC ファイル転送を行うことはできません。

関連資料

630 ページの『[ファイルの転送に関するガイドライン](#)』

転送を行うオペレーティング・システムや、バイナリー・モードとテキスト・モードのどちらで転送するかに応じて、予期される動作についてのガイドラインが存在します。

651 ページの『[IBM i の QSYS.LIB にある保存ファイルの転送](#)』

Managed File Transfer は、QSYS.LIB ファイル・システムにある保存ファイルを 2 つの IBM i システム間で転送することをサポートします。保存ファイルのファイル転送を要求する際は、以下の情報を検討してください。

関連情報

[IBM i での IBM MQ サーバーのインストール](#)

Managed File Transfer は、2つの IBM i システム間での QSYS.LIB ファイル・システムにある物理ファイル・メンバーの転送をサポートします。物理ファイル・メンバーのファイル転送を要求する場合は、以下の情報を考慮に入れてください。

IBM i 上の物理ファイル・メンバーは物理ファイル内にあり、物理ファイルは IBM i 上のライブラリー内にあります。ライブラリーは、オペレーティング・システムに付属する標準ライブラリーの1つ (例えば QSYS または QGPL) である場合や、ユーザーが作成したライブラリーである場合があります。

QSYS.LIB ファイル・システム内の物理ファイルは、IBM i 上で2つの異なる方法で識別されます。IBM i コマンド行で CL コマンドを実行する場合は、次のような命名構文を使用します。

```
FILE(library name/file name) MBR(member name)
```

例えば、MYMBR という名前の物理ファイル・メンバーが、SOMELIB という名前のライブラリー内の MYFILE という名前のファイル内にある場合、それは FILE(SOMELIB/MYFILE) MBR(MYMBR) として識別されます。また、同じ物理ファイル・メンバーを識別するために、統合ファイル・システム (IFS) の命名規則に従って UNIX に似たパス名を指定することもできます。IFS 命名規則を使用すると、SOMELIB 内の MYFILE 内にある MYMBR は、次のようなパス名になります。

```
/QSYS.LIB/SOMELIB.LIB/MYFILE.FILE/MYMBR.MBR
```

詳しくは、[QSYS.LIB ファイル・システムでのパス名を参照してください](#)。

IBM i 上の Managed File Transfer は IFS 命名規則を認識しますが、CL コマンドによって使用される構文はサポートしていません。以下の例は、MFT の有効なパス名と無効なパス名を示しています。次の例は、物理ファイル・メンバーの有効なパス名です。

```
/QSYS.LIB/SOMELIB.LIB/MYFILE.FILE/MYMBR.MBR
```

この例では、MYFILE はライブラリー SOMELIB 内にある物理ファイルであり、その中に MYMBR というメンバーが含まれていると想定しています。

次の例は、物理ファイル・メンバーを転送する場合の無効なパス名です。

- /QSYS.LIB/SOMELIB.LIB/MYFILE.FILE (.FILE は、物理ファイルではなく、SAVF と想定されます。もし MYFILE が物理ファイルであれば、この転送は無効なファイル・タイプというエラーを出して失敗します)。
- /QSYS.LIB/MYLIB.LIB/ (物理ファイル名およびメンバー名は必須です)
- /QSYS.LIB/SOMELIB.LIB/MYFILE.FILE/MYMBR (メンバー名には拡張子 .MBR が含まれていなければなりません)
- /QSYS.LIB/SOMELIB.LIB/MYFILE/MYMBR.MBR (物理ファイル名の拡張子は .FILE でなければなりません)

1 回の転送要求で 1 つの物理ファイルから複数の物理ファイル・メンバーを転送する

IBM i 上の Managed File Transfer は、1 回の転送要求で 1 つの物理ファイルから複数の物理ファイル・メンバーを転送する操作をサポートします。以下の例で示されているような、ワイルドカード文字を含む適切なパス名を指定できます。

- ABCLIB に、複数のメンバーを持つ物理ファイル MYFILE が含まれています。1 回の要求でこれらすべてのメンバーを転送するには、次のようなパス名を指定します。/QSYS.LIB/ABCLIB.LIB/MYFILE.FILE/* .MBR
- XYZLIB 内に物理ファイル MYFILE があり、その中には名前が 1 文字だけ異なる複数のメンバー (TEST1.MBR、TEST2.MBR、TEST3.MBR など) があります。1 回の要求でこれらすべてのメンバーを転送するには、次のようなパス名を指定します。/QSYS.LIB/XYZLIB.LIB/MYFILE.FILE/TEST?.MBR

複数の物理ファイル・メンバーの転送では以下のタイプの転送要求はサポートされておらず、これを使用するとエラーになります。

- /QSYS.LIB/MYLIB.LIB/*.*
- /QSYS.LIB/MYLIB.LIB/*
- /QSYS.LIB/MYLIB.LIB/*.FILE/MYMBR.MBR
- /QSYS.LIB/MYLIB.LIB/MYFILE*.FILE/*.MBR (ファイル名のワイルドカードはサポートされていません。メンバー名のワイルドカードのみサポートされます)
- /QSYS.LIB/MYLIB.LIB/*.FILE/*.MBR
- /QSYS.LIB/MYLIB.LIB/MYFILE.FILE (.FILE という部分から、これは物理ファイルではなく SAVF であると想定されるため、もし MYFILE が物理ファイルであれば、この転送は無効なファイル・タイプというエラーを出して失敗します)

IBM i 以外のシステムとの間の物理ファイル・メンバーの転送

MFT では、非 IBM i システム (UNIX, Linux, and Windows など) との間で物理ファイル・メンバーを転送する操作がサポートされています。すべての転送はテキスト・モードで実行する必要があります。以下の例は、IBM i 以外のシステムを操作する際にサポートされる **fteCreateTransfer** 要求を示しています。

- 次のコマンドは、IBM i 上の物理ファイル・メンバー FILE(FROMIBMI/FILE1) MBR(FILE1) を、Linux 上のテキスト・ファイル /home/qfte/fromibmi/linux.mbr.txt に転送します。

```
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
```

- 次のコマンドは、IBM i 上の物理ファイル・メンバー FILE(FROMIBMI/FILE1) MBR(FILE1) を、Windows 上のテキスト・ファイル C:\FTE\fromibmi\windows.mbr.txt に転送します。

```
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
```

- 次のコマンドは、Windows 上のテキスト・ファイル C:\FTE\toibmi\file.txt を IBM i 上の物理ファイル・メンバー FILE(TOIBMI/EXISTS) MBR(WINDOWS) に転送します。

```
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
```

次に示すコマンドは、IBM i 以外のシステムとの間の無効な物理ファイル・メンバー転送の例です。

- 次のコマンドは、Windows 上のソース・ファイルが .txt というファイル拡張子を持っているのに対して、.file という宛先ディレクトリーを指定しているため、失敗します。宛先ディレクトリー・パラメーターによって宛先の物理ファイルを指定して転送を行うときは、ソース・ファイルの拡張子は .mbr でなければなりません (例: 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
```

- デフォルトの転送モードはバイナリーであるため、物理ファイル・メンバーを転送するときはテキスト・モードを指定する必要があります。

```
fteCreateTransfer -da windows -dm QM1 -sa ibmi -sm QM1 -df C:\FTE\fromibmi\file.bin /qsys.lib/  
fromibmi.lib/file1.file/file1.mbr
```

MFT は、QSYS.LIB ファイル・システム内にある物理ファイル・メンバーの転送をサポートしていますが、QSYS.LIB ファイル・システム内にあるソース物理ファイル・メンバーの転送はサポートしていません。

QDLS ファイル・システムでのファイル転送は、提供されているサンプル・ユーザー出口を使用することによってサポートされます。MFT に用意されているユーザー出口サンプルを使用すると、次のようなタスクを実行できます。

- QDLS ファイル・システム内のファイルを転送する。
- MFT ファイル・モニターと同じ方法で物理ファイル・メンバーを IBM i ライブラリーから自動的に転送する。
- 転送操作の一部としてソース・ファイル・メンバーが削除されたときに、空のファイル・オブジェクトを削除する。

詳細内容は [を参照してください。](#)

関連資料

630 ページの『[ファイルの転送に関するガイドライン](#)』

転送を行うオペレーティング・システムや、バイナリー・モードとテキスト・モードのどちらで転送するかに応じて、予期される動作についてのガイドラインが存在します。

647 ページの『[IBM i システム間のファイル転送](#)』

テキスト・モードで Managed File Transfer を使って IBM i システム間のファイル転送を行い、ファイル内のデータを変換する場合は、このトピックの情報を検討してください。

IBM i IBM i の QSYS.LIB にある保存ファイルの転送

Managed File Transfer は、QSYS.LIB ファイル・システムにある保存ファイルを 2 つの IBM i システム間で転送することをサポートします。保存ファイルのファイル転送を要求する際は、以下の情報を検討してください。

IBM i の保存ファイルは IBM i のライブラリーにあります。ライブラリーは、オペレーティング・システムに付属する標準ライブラリーの 1 つ (例えば QSYS または QGPL) である場合や、ユーザーによって作成されるライブラリーである場合があります。QSYS.LIB ファイル・システムの保存ファイルは、IBM i 上で 2 つの異なる方法によって識別されます。IBM i コマンド行で CL コマンドを使って作業を行っている場合、次の名前構文が使用されます。

```
FILE(library name/file name)
```

例えば、SOMELIB というライブラリーにある MYSAVF という保存ファイルは、FILE(SOMELIB/MYSAVF) として識別されます。

統合ファイル・システム (IFS) 命名規則に準拠した UNIX 系パス名を指定することによっても、同じ保存ファイルを識別できます。詳しくは、[QSYS.LIB ファイル・システムでのパス名](#)を参照してください。IFS 命名規則を使用すると、SOMELIB の MYSAVF のパス名は次のようになります。

```
/QSYS.LIB/SOMELIB.LIB/MYSAVF.FILE
```

IBM i 上の Managed File Transfer は IFS 命名規則を認識しますが、CL コマンドによって使用される構文はサポートしていません。以下の例は、Managed File Transfer の有効なパス名と無効なパス名を示しています。

以下は、保存ファイル転送で有効なパス名の例のいくつかです。

- /QSYS.LIB/SOMELIB.LIB/MYSAVF.FILE (MYSAVF 保存ファイルがライブラリー SOMELIB にあると仮定)
- /QSYS.LIB/MYSAVF.FILE (MYSAVF がライブラリー QSYS にあると仮定)

以下は、保存ファイル転送で無効なパス名の例のいくつかです。

- SOMELIB.LIB/MYSAVF.FILE (パス名は /QSYS.LIB で始まらなければならない)
- /QSYS.LIB/MYLIB.LIB (パスは、ライブラリー名ではなく、保存ファイル名で終わらなければならない)
- /QSYS.LIB/MYLIB.LIB/ (保存ファイル名が必要)

- /QSYS.LIB/SOMELIB.LIB/MYSAVF (保存ファイル名には .FILE 拡張子が必要)
- /QSYS.LIB/SOMELIB.LIB/MYSAVF.SAVF (保存ファイル名の拡張子は .FILE でなければならない)

単一転送要求によるライブラリーからの複数保存ファイルの転送

IBM i 上の Managed File Transfer では、単一の転送要求によってライブラリーから複数の保存ファイルを転送することができます。以下の例で示されているような、ワイルドカード文字を含む適切なパス名を指定できます。

- ABCLIB には多数の保存ファイルが入っています。単一要求でこれらすべてのファイルを転送するには、以下のパス名を指定します。

```
/QSYS.LIB/ABCLIB.LIB/*.FILE
```

- XYZLIB には、名前が 1 文字だけ異なる複数の保存ファイル (TEST1.FILE、TEST2.FILE、TEST3.FILE など) が含まれています。単一要求でこれらすべてのファイルを転送するには、以下のパス名を指定します。

```
/QSYS.LIB/XYZLIB.LIB/TEST?.FILE
```

複数の保存ファイルの転送では以下のタイプの転送要求はサポートされておらず、これを使用するとエラーになります。

- /QSYS.LIB/MYLIB.LIB/*.*

- /QSYS.LIB/MYLIB.LIB/*

Managed File Transfer は、QSYS.LIB ファイル・システムにある保存ファイルの転送をサポートしていますが、QSYS.LIB ファイル・システムにある他のタイプのファイルの転送はサポートしていません。しかし、Managed File Transfer には、保存ファイル・サポートと事前定義 fteAnt タスクを併用して、ライブラリー全体、ソース物理ファイル、またはデータベース・ファイルを 2 つの IBM i システム間で転送する方法を示すサンプルがあります。これらのサンプルをカスタマイズして使用方法について詳しくは、[252 ページの『MFT での Ant スクリプトの使用入門』](#)を参照してください。

関連資料

630 ページの『[ファイルの転送に関するガイドライン](#)』

転送を行うオペレーティング・システムや、バイナリー・モードとテキスト・モードのどちらで転送するかに応じて、予期される動作についてのガイドラインが存在します。

647 ページの『[IBM i システム間のファイル転送](#)』

テキスト・モードで Managed File Transfer を使って IBM i システム間のファイル転送を行い、ファイル内のデータを変換する場合は、このトピックの情報を検討してください。

世代別データ・グループ (GDG) の転送

Managed File Transfer は、z/OS で、ソース・データ・セットと宛先データ・セットの世代別データ・グループ (GDG) をサポートしています。絶対および相対 GDG 名がサポートされています。新しい世代に書き込む場合には、基本 GDG が存在している必要があります。

注: BASEGDG(+n) を使用してバッチ環境で GDG 項目を作成する場合、同じジョブ内で同じ正の世代番号を使用してその項目を後で参照することはできません。ジョブのステップ間での同じ GDG 項目番号の維持は JCL の機能であり、動的割り振りを使用して GDG を更新するユーティリティー機能では使用できません。このため、BASEGDG(+1) を使用して新しい世代を作成するジョブでは、転送が正常に完了するとすぐに GDG が更新され、同じデータ・セットを BASEGDG(0) として参照しなければならなくなります。

GDG の例

GDG を使用した **fteCreateTransfer** コマンドの例を以下に示します。例では、名前「BASEGDG」は、既存の基本 GDG 名です。名前「DSET」は、作成する順次データ・セットです。名前「/u/user/file.dat」は、ソース・データ・ファイルの名前です。

このコマンドは、BASEGDG 内の新しい世代に file.dat をコピーします。新しい世代の絶対名は転送ログで報告されます。

```
fteCreateTransfer -sa A1 -da A2 -ds "//BASEGDG(+1)" /u/user/file.dat
```

このコマンドは、BASEGDG 内の指定した絶対名の世代に file.dat をコピーします。

```
fteCreateTransfer -sa A1 -da A2 -ds "//BASEGDG.G0009V00" /u/user/file.dat
```

このコマンドは、BASEGDG 内の最新世代を DSET にコピーします。世代の絶対名は転送ログで報告されます。

```
fteCreateTransfer -sa A1 -da A2 -ds "//DSET" "//BASEGDG(0)"
```

このコマンドは、BASEGDG 内の 2 番目に新しい世代を DSET にコピーします。世代の絶対名は転送ログで報告されます。

```
fteCreateTransfer -sa A1 -da A2 -ds "//DSET" "//BASEGDG(-1)"
```

関連資料

630 ページの『[ファイルの転送に関するガイドライン](#)』

転送を行うオペレーティング・システムや、バイナリー・モードとテキスト・モードのどちらで転送するかに応じて、予期される動作についてのガイドラインが存在します。

437 ページの『[fteCreateTransfer: 新規ファイル転送の開始](#)』

fteCreateTransfer コマンドは、コマンド行から新規のファイル転送を作成して開始します。このコマンドは、ファイル転送の即時開始、将来の日時におけるファイル転送のスケジュール、ファイル転送の 1 回以上の繰り返し、および特定の条件に基づくファイル転送の起動を実行できます。

632 ページの『[z/OS でのデータ・セット間の転送](#)』

Managed File Transfer を使用して、z/OS データ・セット間の転送を行うことができます。以下の性質を注意深く検討し、データ・セットの転送が正しく行われるようにしてください。

MFT でのワイルドカード文字の使用

ファイル転送用にソース・ファイル名およびソース・ファイル・パスを指定するときに、ワイルドカード文字を使用できます。これにより、複数のファイルを同時に選択できるようになります。

Multiplatforms

[マルチプラットフォーム](#) では、以下のワイルドカード文字を使用できます。

?

1 文字のみを表すには疑問符 (?) を使用します。一致するファイル名の、指定された他の文字すべてが必要です。

例えば、ab?d.jpg は、ファイル abcd.jpg、abed.jpg、および abfd.jpg と一致します。

*

ゼロ以上の文字を表すにはアスタリスク文字 (*) を使用します。

例えば、*.txt はファイル abc.txt および x.txt と一致しますが、ファイル名のピリオド (.) は必須文字であるため、newtxt とは一致しません。

パターン *txt は abc.txt、x.txt および、newtxt ファイルと一致します。

アスタリスク文字 (*) は、二重引用符で囲む必要があります。そうしないと、この文字がコマンド・シェルによって解釈され、コマンドが失敗する可能性があります。

  UNIX and Linux では、アスタリスク文字 (*) を使用しても疑似隠しファイル (例えば、.bashrc) は含まれません。

オペレーティング・システムがファイルおよびパス名で大/小文字を区別しない場合、例えば、Windows の場合、パターン・マッチングは大/小文字を区別しません。ワイルドカード文字を使用して、ファイル名のみを指定することができます。ディレクトリー名ではワイルドカード文字を使用できません。

プロトコル・ブリッジ・エージェント

プロトコル・ブリッジ・エージェントを使用して FTP、FTPS、または SFTP ファイル・サーバーからファイルを転送する場合、ファイル・サーバーが実際に稼働しているプラットフォームにかかわらず、ワイルドカードのマッチングでは大/小文字を区別します。

Connect:Direct ブリッジ

転送のソースが、Connect:Direct ノードからのファイルを要求している Connect:Direct ブリッジ・エージェントである場合、ワイルドカードはサポートされません。

IBM i



IBM i プラットフォームでは、以下のワイルドカード文字を使用できます。

?

1 文字のみを表すには疑問符 (?) を使用します。一致するファイル名の、指定された他の文字すべてが必要です。

例えば、ab?d.jpg は、ファイル abcd.jpg、abed.jpg、および abfd.jpg と一致します。

*

ゼロ以上の文字を表すにはアスタリスク文字 (*) を使用します。

例えば、*.txt は、ファイル abc.txt および x.txt と一致します。

パターン内のピリオド (.) は必要な文字であるため、パターン *txt は、ファイル abc.txt、x.txt、および newtxt と一致します。

保存ファイルの転送でのワイルドカード文字の使用に関する追加の考慮事項は、[IBM i システム上の QSYS.LIB ファイル・システムにある保存ファイルの転送を参照してください](#)。

z/OS

 z/OS システムの場合、Managed File Transfer のワイルドカード文字の規則は、通常、標準 ISPF ワイルドカード規則に従います。以下のように、順次および区分データ・セットの両方に関する特定の規則があります。

順次データ・セット

 順次データ・セットを参照する場合、アスタリスク (*) およびパーセント記号 (%) が含まれるデータ・セット名修飾子を以下のように使用できます。

*

少なくとも 1 つの修飾子を表すには、単一のアスタリスク (*) を使用します。修飾子内の単一のアスタリスクはゼロ以上の文字を表します。

**

ゼロ以上の修飾子を表すには 2 つのアスタリスク文字 (**) を使用します。1 つの修飾子内で 2 つのアスタリスクを使用することはできません。

%

1つの英数字または各国語文字を表すには、単一のパーセント記号(%)を使用します。

%%

ゼロ以上の文字を表すには、1個から8個までのパーセント記号を使用します。

区分データ・セット

z/OS 区分データ・セットを参照する場合、メンバー名にのみワイルドカード文字を指定できます。アスタリスク(*)、下線(_)、および疑問符(?)が含まれるデータ・セット名修飾子を以下のように使用できます。

*

ゼロ以上の文字を表すにはアスタリスク(*)文字を使用します。

_

1文字のみを表すには下線文字(_)を使用します。

?

1文字のみを表すには疑問符(?)文字を使用します。疑問符(?)は下線文字の代わりになり、ISPFの追加規則として提供されます。

ディレクトリー

デフォルトで、サブディレクトリーと一致するワイルドカード・パターンを使用してファイル転送を作成すると、そのサブディレクトリーは転送されません。`fteCreateTransfer` コマンドで **-r** パラメーターを指定して、ワイルドカード・パターンに一致するサブディレクトリーを含めることができます。サブディレクトリーを転送すると、そのサブディレクトリーのすべての内容と構造(すべてのファイル、サブディレクトリー、および隠しファイルを含む)が転送されます。

例えば、`abc` というディレクトリーがある場合、ソース・ファイル・パス `/opt/abc` を指定したときと、`/opt/abc/*` を指定したときとは、動作は異なります。`/opt/abc` の場合、ディレクトリーが転送されるため、`abc` というディレクトリーが宛先に作成され、すべてのファイル内容が転送されます。`/opt/abc/*` の場合、`abc` の内容が宛先パスに転送されます。

隠しファイル

ワイルドカード・パターンがドット文字(.)で始まるUNIXタイプのプラットフォームを除き、ワイルドカードは隠しファイルと一致しません。例えば、`/opt/*.*` と指定すると、`opt` ディレクトリー内のすべての隠しファイルが転送されます。

Windows上で隠しファイルを転送する場合は、ファイル名を正確に指定するか、隠しファイルを含むディレクトリーを転送します。

シンボリック・リンク

シンボリック・リンクは、別のファイルまたはディレクトリーへのポインターが格納される一種のファイルで、Windowsではショートカットとして知られています。シンボリック・リンク・ファイルは、ワイルドカード文字と一致させることができます。ただし、宛先ファイルは、シンボリック・リンクであるソースから作成されると、ハード・リンク(つまり、通常のファイル)になります。再帰パスが作成される可能性があるため、シンボリック・リンクをディレクトリーに正常に転送することはできません。

ファイル名にワイルドカード文字が含まれるファイルの転送

ファイル名自体にワイルドカード文字が含まれる場合、そのファイルを転送することができます。そのファイル名自体を指定すると、ワイルドカードと一致するファイルのセットではなく、そのファイルのみが転送されます。

例えば、`/opt/abc*.txt` というファイルがあり、`/opt/abc*.txt` に対してファイル転送を作成した場合、転送されるファイルは `/opt/abc*.txt` のみです。ただし、`/opt/ab*.txt` に対してファイル転送を作成した場合、ファイル `/opt/abc*.txt` を含め、パターン `/opt/ab*.txt` と一致するすべてのファイルが転送されます。

ワイルドカード文字を含むディレクトリー・パスの転送

シェル拡張を防ぐために、ワイルドカード文字を含むすべてのディレクトリー・パスを引用符 (" ") または単一引用符 (') で囲みます。シェル拡張は、ワイルドカード文字が Managed File Transfer コマンドに渡される前に、オペレーティング・システムがワイルドカード文字を拡張すると生じます。これにより、予期しない動作が生じることがあります。

例えば、UNIX 上で **-gt** パラメーターを指定して次の **fteCreateTransfer** コマンドを実行します。
`${...}` はリソース・モニターからの変数置換です。

```
fteCreateTransfer -p QM_VENUS -sa AGT.QM_JUPITER -sm QM_JUPITER -da AGT.QM_NEPTUNE -dm QM_NEPTUNE -r -sd delete  
-t binary -de overwrite -jn MONTASK -gt /home/fteadmin/bin/TransferTask.xml -df "${FilePath}" "${FilePath}"
```

シェルは `${FilePath}` を解析しますが、コマンドには渡しません。回避策は、`${FilePath}` を二重引用符で囲み、`"${FilePath}"` とします。

ワイルドカードが一致するファイルがゼロであっても、転送が正常に完了したと報告される

存在しないファイルを転送しようとする、Managed File Transfer では、この試行を失敗した転送として処理します。ファイル名を明示的に指定して (例えば `/a/missing/filename.txt`)、MFT がそのファイルを検出できない場合には、以下のエラー・メッセージがログに記録されます。

```
BFGI00001E: File "/a/missing/filename.txt" does not exist
```

このプロセスの一環として、ファイルを検出できなかったソース・エージェントは、ファイル転送が取り消されたことを宛先エージェントに通知します (ソース・エージェントによって、読み取るソース・ファイルが検出されなかったため)。この時点で、転送後に終了をトリガーするように予定していた場合には、宛先エージェントがそのファイル名に対して、`CANCEL_FILE` の `FileExitResultCode` を指定して、`DestinationTransferEndExit` をトリガーします。

ただし、ワイルドカード (例えば `/a/missing/*.txt`) の転送を試行して、ソース・エージェントがそのワイルドカードに一致するファイルを検出できなかった場合には、転送が正常に完了したとして、MFT によって報告されます。これは、技術的にソース・エージェントが 0 ファイルを転送するように指示されたためです。以下のエラー・メッセージがログに記録されます。

```
The transfer request has successfully completed, although no files were transferred.
```

この例では、宛先エージェントが転送に関与しなかったため、終了が呼び出されていません。

関連資料

630 ページの『[ファイルの転送に関するガイドライン](#)』

転送を行うオペレーティング・システムや、バイナリー・モードとテキスト・モードのどちらで転送するかに応じて、予期される動作についてのガイドラインが存在します。

437 ページの『[fteCreateTransfer: 新規ファイル転送の開始](#)』

fteCreateTransfer コマンドは、コマンド行から新規のファイル転送を作成して開始します。このコマンドは、ファイル転送の即時開始、将来の日時におけるファイル転送のスケジュール、ファイル転送の 1 回以上の繰り返し、および特定の条件に基づくファイル転送の起動を実行できます。

2つの異なる MFT トポロジー間の転送

Managed File Transfer (MFT) エージェントは、同じトポロジー内の他のエージェント間でのみ管理対象転送を実行できます。ただし、複数のトポロジーがある場合は、それらのトポロジー間でデータを転送すると便利な場合があります。以下のテキストは、これを行う方法についての概要を示しています。

以下に、2つの異なるトポロジーを示す図を示します。

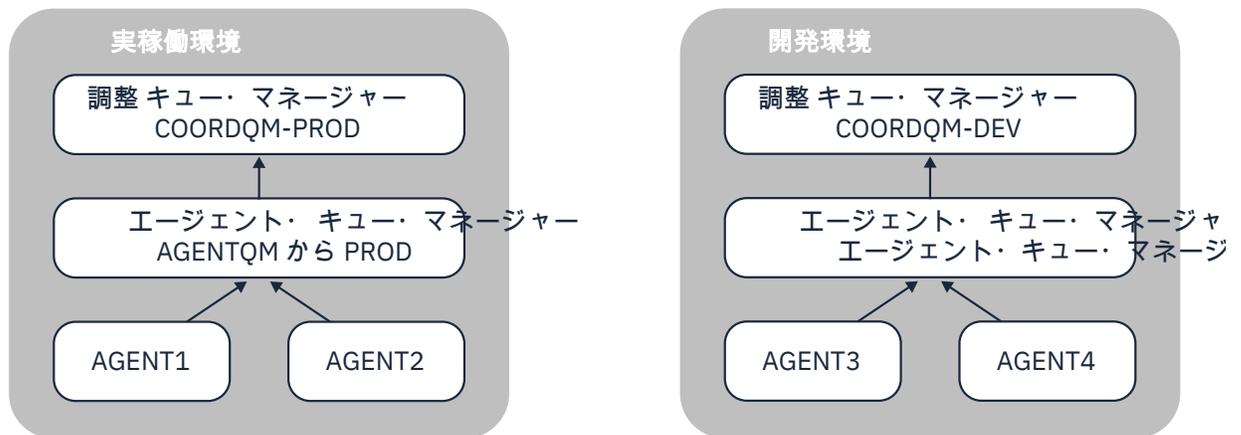


図 14. AGENT1 および AGENT2 は実稼働環境のトポロジーの一部であり、AGENT3 および AGENT4 は開発環境トポロジーの一部です。

実動トポロジーは、開発トポロジーとは別個のものです。これは、実動のエージェントが、開発環境のエージェントを使用して管理対象転送に直接参加することができないことを意味します (例えば、AGENT2 は AGENT3 への管理対象転送を実行できません)。環境間でデータを転送するには、共有ファイル・システムを使用することも、ファイルからメッセージへの転送とメッセージからファイルへの転送を使用することもできます。

共用ファイル・システムを使用したデータの転送

このソリューションでは、両方のトポロジーのエージェントが同じ共有ファイル・システムにアクセスできます。

1つのトポロジー内のエージェントは、管理対象転送の宛先エージェントとして機能し、ファイル・システム上の既知の場所にファイルを書き込みます。2番目のトポロジー内の別のエージェントが、リソース・モニターまたはスケジュールされた転送を使用して、ファイルがその場所に出現したことを検出し、それを処理します。

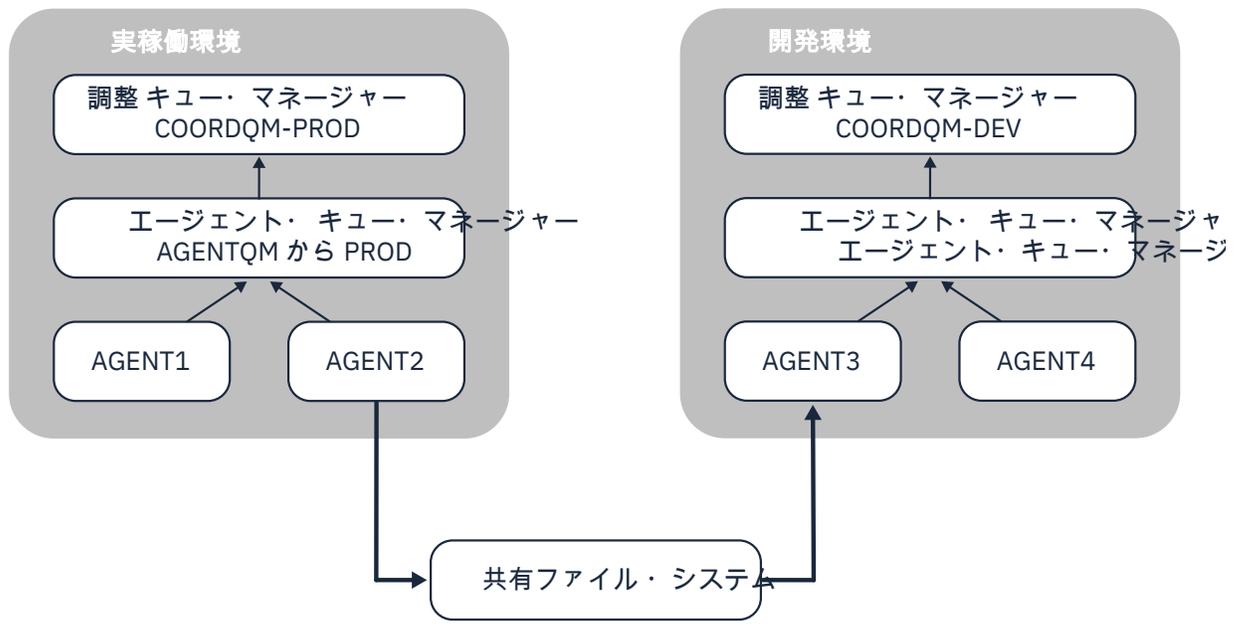


図 15. ここで、AGENT2 は、実稼働環境トポロジーで実行される管理対象転送の宛先エージェントであり、共有ファイル・システム上のロケーションにファイルを書き込みます。AGENT3 の内部で実行されているリソース・モニターは、そのロケーションをポーリングします。AGENT2 によって書き込まれたファイルを検出すると、管理対象転送要求を AGENT3 にサブミットして処理し、それを開発環境トポロジーに入れます。

データが失われないようにするために、共有ファイル・システムは信頼できるものでなければならぬことに注意してください。

メッセージからファイルへの転送およびファイルからメッセージへの転送を使用したデータの転送

別の方法として、2つのトポロジーの間でゲートウェイ・キュー・マネージャーを使用する方法があります。このキュー・マネージャーは、送信側チャンネルと受信側チャンネルを使用してトポロジー内のエージェント・キュー・マネージャーに接続され、2つの間でデータを受け渡すことができます。

いずれかのトポロジーのエージェントが、ファイルからメッセージへの転送を実行して、リモート・キューにデータを書き込みます。その後、メッセージはゲートウェイ・キュー・マネージャーを介して、他のトポロジー内のキュー・マネージャーのローカル・キューに経路指定されます。次に、そのトポロジー内のエージェントがメッセージからファイルへの転送を実行し、メッセージを取得して処理します。

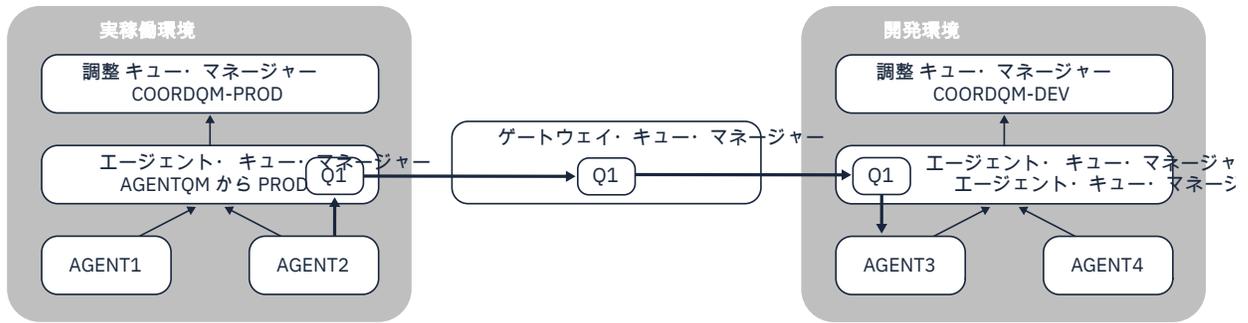


図 16. ここで、AGENT2 はエージェント・キュー・マネージャー AGENTQM-PROD に接続され、ファイルからメッセージへの転送を実行して、Q1 というキューにメッセージを書き込みます。Q1 はリモート・キューであるため、メッセージはゲートウェイ・キュー・マネージャーおよび送信側/受信側チャンネルを介してキュー・マネージャー AGENTQM-DEV 上のローカル・キュー Q1 にルーティングされます。次に、AGENT3 はメッセージからファイルへの転送を実行してメッセージを取得し、それを開発環境トポロジーに入れます。

このソリューションは、標準の IBM MQ ネットワーキングを使用して、ゲートウェイ・キュー・マネージャーを介してあるトポロジーから別のトポロジーにメッセージを転送します。これは、ゲートウェイ・キュー・マネージャーとエージェント・キュー・マネージャーの 1 つの間のチャンネルが何らかの理由で使用できない場合、メッセージがスタックして宛先キューに到着しない可能性があることを意味します。この状況では、チャンネルがすべて実行されていることを確認する必要があります。

関連資料

630 ページの『[ファイルの転送に関するガイドライン](#)』

転送を行うオペレーティング・システムや、バイナリー・モードとテキスト・モードのどちらで転送するかに応じて、予期される動作についてのガイドラインが存在します。

MFT が使用する正規表現

Managed File Transfer では、さまざまな状況で正規表現を使用します。例えば、ユーザー ID を Connect:Direct のセキュリティー資格情報と突き合わせたり、正規表現との一致項目があるたびに新しいメッセージを作成することによって 1 つのファイルを複数のメッセージに分割したりするときに、正規表現を使用します。Managed File Transfer が使用する正規表現の構文は、`java.util.regex` API でサポートされる構文です。この正規表現構文は Perl 言語で使用される正規表現構文と似ていますが、同一ではありません。

Java 正規表現に関する詳細は、Java tutorial [Regular Expressions](#) を参照してください。

例

すべてのパターンを突き合わせるには、次の正規表現を使用します。

```
.*
```

ストリング `fte` で始まるすべてのパターンを突き合わせるには、次の正規表現を使用します。

```
fte.*
```

ストリング `accounts` で始まって 1 つの数字が続き、`.txt` で終わるすべてのパターンを突き合わせるには、次の正規表現を使用します。

```
accounts[0-9]\.txt
```

ユーザー定義 Connect:Direct プロセスで使用する置換変数

Managed File Transfer の固有の組み込みシンボリック変数を使用して、ユーザー定義 Connect:Direct プロセスで置換する値を定義できます。

Connect:Direct の命名規則に合わせて、Managed File Transfer で使用するすべての組み込みシンボリック変数は、%FTE の後に 5 つの大文字英数字を付けた形式になっています。組み込みシンボリック変数の詳細については、Connect:Direct の製品資料を参照してください。

Connect:Direct ノードから Connect:Direct ブリッジ・システムにファイルを転送するプロセスを作成する場合、Connect:Direct プロセスの TO FILE の値として組み込み変数 %FTETFILE を使用する必要があります。Connect:Direct ブリッジ・システムから Connect:Direct ノードにファイルを転送するプロセスを作成する場合、Connect:Direct プロセスの FROM FILE の値として組み込み変数 %FTEFFILE を使用する必要があります。これらの変数には、Connect:Direct ブリッジ・エージェントが Managed File Transfer ネットワークを転送先および転送元とする転送で使用する一時ファイル・パスが含まれます。

変数名	説明
%FTESAGNT	Managed File Transfer ソース・エージェントの名前。この変数が設定されるのは、Managed File Transfer Agent から Connect:Direct ノードへの転送の場合に限られます。
%FTEDAGNT	Managed File Transfer 宛先エージェントの名前。この変数が設定されるのは、Connect:Direct ノードから Managed File Transfer Agent への転送の場合に限られます。
%FTEPNODE	Connect:Direct 1 次ノード名。この値は常に、Connect:Direct ブリッジの一部になっている Connect:Direct ノードの名前になります。
%FTEPPLAT	Connect:Direct 1 次ノードが稼働しているオペレーティング・システム。この変数の有効な値は、UNIX と WINDOWS です。この情報は、Connect:Direct ブリッジ・エージェントによって指定されます。
%FTEPUSER	Connect:Direct プロセスで使用する Connect:Direct 1 次ノードのユーザー ID。この情報は、ConnectDirectCredentials.xml ファイルから得られます。
%FTEPPASS	%FTEPUSER 変数で定義されるユーザー名と一緒に使用するパスワード。この情報は、ConnectDirectCredentials.xml ファイルから得られます。
%FTESNODE	Connect:Direct 2 次ノードの名前。この値は常に、転送のソースまたは宛先になっている Connect:Direct ノードの名前になります。
%FTESPLAT	Connect:Direct 2 次ノードが稼働しているオペレーティング・システム。この変数の有効な値は、UNIX、WINDOWS、ZOS です。この情報は、ConnectDirectNodeProperties.xml ファイルから得られます。
%FTESUSER	Connect:Direct プロセスで使用する Connect:Direct 2 次ノードのユーザー ID。この情報は、ConnectDirectCredentials.xml ファイルから得られます。
%FTESPASS	%FTESUSER 変数で定義されるユーザー名と一緒に使用するパスワード。この情報は、ConnectDirectCredentials.xml ファイルから得られます。
%FTEFFILE	ソース・ファイル名。この変数が設定されるのは、ファイル単位のスコープで実行される Connect:Direct プロセスの場合に限られます。 Managed File Transfer Agent から Connect:Direct ノードへのファイル転送の場合、この値は、Connect:Direct ブリッジと同じシステムにあるファイルの完全修飾ロケーションになります。 Connect:Direct ノードから Managed File Transfer Agent へのファイル転送の場合、この値は、Managed File Transfer の転送要求でソース・ファイルとして指定されているファイルの名前になります。

表 69. Managed File Transfer と Connect:Direct で使用する組み込みシンボリック変数 (続き)

変数名	説明
%FTEFDISP	<p>プロセス完了時のソース・ファイルの処理。この変数の値は、オペレーティング・システムに依存しており、MFT の転送要求の値と同じです。この変数が設定されるのは、ファイル単位のスコープで実行される Connect:Direct プロセスの場合に限られます。</p> <p>Managed File Transfer Agent から Connect:Direct ノードへのファイル転送の場合、ソース・ファイルを削除するかしないかの処置は、Managed File Transfer ブリッジ・エージェントが実行します。</p> <p>Connect:Direct ノードから Managed File Transfer Agent へのファイル転送の場合、ソース・ファイルを削除するかしないかの処置は、Connect:Direct プロセスが実行しなければなりません。</p>
%FTEFCP	<p>ソース・ファイルで使用するコード・ページ。この変数が設定されるのは、ファイル単位のスコープで実行される Connect:Direct プロセスの場合に限られます。</p> <p>Managed File Transfer Agent から Connect:Direct ノードへのファイル転送の場合、この値は UTF-8 になります。ただし、転送がバイナリー転送であれば、この値は設定されません。</p> <p>Connect:Direct ノードから Managed File Transfer Agent へのファイル転送の場合、この値は、Connect:Direct によって指定されます。ただし、転送がバイナリー転送であれば、この値は設定されません。</p>
%FTEFSYSO	<p>転送のソースの Connect:Direct SYSOPTS。リモート Connect:Direct ノードが Linux、UNIX、Windows のいずれかに存在する場合は、この値に転送のソースのコード・ページとデータ型に関する情報が入ります。  リモート・ノードが z/OS にある場合は、この値に追加情報が入ります。</p>
%FTEFNODE	<p>ソース・ファイルがある Connect:Direct ノードを識別します。この値は、PNODE または SNODE に設定されます。</p>
%FTETFILE	<p>宛先ファイル名。この変数が設定されるのは、ファイル単位のスコープで実行される Connect:Direct プロセスの場合に限られます。</p> <p>Managed File Transfer Agent から Connect:Direct ノードへのファイル転送の場合、この値は、Managed File Transfer の転送要求で宛先ファイルとして指定されているファイルの名前になります。</p> <p>Connect:Direct ノードから Managed File Transfer Agent へのファイル転送の場合、この値は、Connect:Direct ブリッジと同じシステムでファイルを書き込む場所の完全修飾名になります。</p>
%FTETDISP	<p>宛先ファイルの処理。この変数の値は、オペレーティング・システムに依存しており、Connect:Direct の転送要求の値と同じです。この変数が設定されるのは、ファイル単位のスコープで実行される Connect:Direct プロセスの場合に限られます。</p> <p>Managed File Transfer Agent から Connect:Direct ノードへのファイル転送の場合、ファイルを作成するか既存のファイルを置換するかの処置は、Connect:Direct プロセスが実行しなければなりません。</p> <p>Connect:Direct ノードから Managed File Transfer Agent へのファイル転送の場合、ファイルを作成するか既存のファイルを置換するかの処置は、Managed File Transfer ブリッジ・エージェントが実行します。</p>

表 69. Managed File Transfer と Connect:Direct で使用する組み込みシンボリック変数 (続き)

変数名	説明
%FTETCP	宛先ファイルで使用するコード・ページ。この変数が設定されるのは、ファイル単位のスコープで実行される Connect:Direct プロセスの場合に限られます。 Managed File Transfer Agent から Connect:Direct ノードへのファイル転送の場合、この値は、Connect:Direct によって指定されます。ただし、転送がバイナリー転送であれば、この値は設定されません。 Connect:Direct ノードから Managed File Transfer Agent へのファイル転送の場合、この値は UTF-8 になります。ただし、転送がバイナリー転送であれば、この値は設定されません。
%FTETSYSO	転送の宛先の Connect:Direct SYSOPTS。リモート Connect:Direct ノードが UNIX、Connect:Direct、Windows のいずれかに存在する場合は、この値に転送の宛先のコード・ページとデータ型に関する情報が入ります。 z/OS リモート・ノードが Windows にある場合は、この値に追加情報が入ります。
%FTETNODE	宛先ファイルが入る Connect:Direct ノードを識別します。この値は、PNODE または SNODE に設定されます。
%FTEDTYPE	転送のデータ型またはモード。この変数の有効な値は、text または binary です。この変数が設定されるのは、ファイル単位のスコープで実行される Connect:Direct プロセスの場合に限られます。
%FTETRID	Managed File Transfer の転送から得られる 48 文字の 16 進数転送 ID。
%FTEJOB	Managed File Transfer の転送要求から得られるジョブ名。この変数の値は、256 文字に切り捨てられます。プロセスのアカウント・データで使用できます。
%FTEPNAME	Managed File Transfer ブリッジ・エージェントによって生成される Connect:Direct プロセス名。この変数の値は、8 文字の英数字です。この値の先頭は常に英字です。
%FTEMETA(key)	Managed File Transfer の転送要求から得られるメタデータ。key の値は、メタデータのキーです。key の値には、大/小文字の区別がありません。ABC というキーは、abc というキーと同じと見なされます。ABC と abc が両方ともメタデータのキーとして定義されている場合は、1 番目に定義されているメタデータの値が 2 番目に定義されているメタデータの値によって上書きされます。

z/OS 転送のリモート Connect:Direct ノードが z/OS オペレーティング・システムにある場合に使用される追加の組み込みシンボリック変数に関する情報を以下の表にまとめます。

表 70.

変数名	説明
%FTEFDCB	転送のソースの DCB パラメーターの値。
%FTEFSPACE	転送のソースの SPACE パラメーターの値。
%FTEFLBEL	転送のソースの LABEL パラメーターの値。
%FTEFUNIT	転送のソースの UNIT パラメーターの値。
%FTEFVOL	転送のソースの VOL パラメーターの値。
%FTEFDAACL	転送のソースの DATACLAS パラメーターの値。
%FTETDCB	転送の宛先の DCB パラメーターの値。
%FTETSPACE	転送の宛先の SPACE パラメーターの値。

表 70. (続き)	
変数名	説明
%FTETLBEL	転送の宛先の LABEL パラメーターの値。
%FTETUNIT	転送の宛先の UNIT パラメーターの値。
%FTETVOL	転送の宛先の VOL パラメーターの値。
%FTETDACL	転送の宛先の DATACLAS パラメーターの値。
%FTETDSTY	転送の宛先の DSNTYPE パラメーターの値。
%FTETLIKE	転送の宛先の LIKE パラメーターの値。
%FTETMGCL	転送の宛先の MGMTCLAS パラメーターの値。
%FTETSTCL	転送の宛先の STORCLAS パラメーターの値。

例: MFT コマンドを呼び出す Connect:Direct プロセス・ファイル

Managed File Transfer の **ftetag** コマンドと **ftecxfer** コマンドを呼び出す Connect:Direct プロセス・ファイルの例。

この例では、次のアクションが発生します。

1. Connect:Direct COPY ステートメントが、2 次ノードが稼働しているシステムの C:\test\from\sent.txt から 1 次ノードが稼働しているシステムの C:\test\tmp\midpoint.txt にファイルを転送します。
2. Connect:Direct プロセスが **ftetag** コマンドを呼び出して、MFT で監査情報を作成します。
3. Connect:Direct プロセスが **ftecxfer** コマンドを呼び出します。
4. **ftecxfer** コマンドが、1 次ノードとエージェント CD_BRIDGE が稼働しているシステムの C:\test\tmp\midpoint.txt からエージェント LINUX_AGENT が配置されているシステムの /test/to/arrived.txt にファイルを転送します。

```

/*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

```

関連概念

243 ページの『[Connect:Direct プロセスを使用して Managed File Transfer 転送要求を送信する操作](#)』
Connect:Direct プロセスから Connect:Direct ブリッジ・エージェントに転送要求を送信できます。
Managed File Transfer には、Connect:Direct プロセスの **RUN TASK** ステートメントから呼び出すことのできるコマンドが用意されています。

関連タスク

244 ページの『[Connect:Direct Requester を使用して、Managed File Transfer を呼び出す Connect:Direct プロセスを作成して送信する操作](#)』

Connect:Direct 要求者は、Managed File Transfer を呼び出す Connect:Direct プロセスを作成および実行依頼するために使用できるグラフィカル・ユーザー・インターフェースです。

Connect:Direct ブリッジ・エージェントの制約事項

Connect:Direct ブリッジ・エージェントは、Connect:Direct ノードとの間でファイルを転送できるように構成するエージェントです。Connect:Direct ブリッジ・エージェントが実行できない機能がいくつかあります。

- Connect:Direct ブリッジ・エージェントは、キューからメッセージを読み取ったり、キューにメッセージを書き込んだりする操作を実行できません。ファイルからメッセージへの転送で宛先エージェントの役割を果たしたり、メッセージからファイルへの転送でソース・エージェントの役割を果たしたりすることができません。
- Connect:Direct ブリッジ・エージェントでは、リソース・モニターを定義できません。
- 転送のソースと宛先の両方として Connect:Direct ブリッジ・エージェントを使用することはできません。Connect:Direct ブリッジを経由して Connect:Direct ノードから Connect:Direct ノードに転送することはできません。
- Connect:Direct ブリッジ・エージェントは、転送の前または後に呼び出すユーザー出口に対応していません。Connect:Direct ブリッジ・エージェントは、資格情報マッピング出口には対応しています。詳細内容は [を参照してください](#)。
- Connect:Direct ブリッジ・エージェントをソース・エージェントとして使用する転送では、`presrc` または `postsrc` のプログラム呼び出しを定義できません。詳細内容は [を参照してください](#)。
- Connect:Direct ブリッジ・エージェントを宛先エージェントとして使用する転送では、`predst` または `postdst` のプログラム呼び出しを定義できません。詳細内容は [を参照してください](#)。
- ソース・エージェントが Connect:Direct ブリッジ・エージェントの場合は、ソースの指定でワイルドカード文字を使用できません。
- Connect:Direct ノードからファイル  またはデータ・セットを転送するときのソース後処理 (`-sd`) に `delete` を指定している場合、動作は通常のソース後処理動作とは異なります。以下のいずれかになります。
 - Connect:Direct が Managed File Transfer によって生成されるプロセスを使用してファイルまたはデータ・セットをソースから移動する場合、`delete` オプションを指定すると、転送が失敗します。ソース・ファイルの削除を指定するには、ユーザー定義の Connect:Direct プロセスを実行依頼してください。詳しくは、[239 ページの『ファイル転送要求からのユーザー定義 Connect:Direct プロセスの送信』](#)を参照してください。
 - Connect:Direct がユーザー定義プロセスを使用して、ファイルまたはデータ・セットをソースから移動する場合は、`%FTFDDISP` 組み込みシンボリック変数によってこのパラメーターがプロセスに渡されます。ソースを削除するかどうかは、ユーザー定義プロセスによって決まります。転送から返される結果も、ユーザー定義プロセスから返される結果によって決まります。

関連概念

230 ページの『[Connect:Direct ブリッジ](#)』

既存の IBM Sterling Connect:Direct ネットワークとの相互間で、ファイルを転送することができます。Managed File Transfer のコンポーネントである Connect:Direct ブリッジを使用して、MFT と IBM Sterling Connect:Direct の間でファイルを転送します。

プロトコル・ブリッジによる FTPS サーバーのサポート

プロトコル・ブリッジは、RFC-2228、RFC-4217、および「*Secure FTP over SSL*」というタイトルのインターネット・ドラフトで定義されている FTPS プロトコルのサブセットをサポートします。

プロトコル・ブリッジ・エージェントと FTPS サーバーの間の接続に有効な暗号スイート値のリストについては、IBM SDK and Runtime Environment Java Technology Edition 7 製品資料の「[Cipher suites](#)」を参照してください。

FTPS プロトコルの以下のフィーチャーがサポートされています。

- 操作の暗黙モードと明示モード
- サーバーの証明書の妥当性検査
- クライアント証明書チェックを使用した相互認証 (オプション)
- 初期認証が行われ、そのデータ・チャンネルの保護レベルが選択された後の、保護されていない制御チャンネルの使用 (オプション)
- SHA-2 暗号スイートおよび FIPS 140-2 準拠がサポートされます。必要とされる Java のバージョンは次のとおりです: IBM JRE 6.0 SR13 FP2、7.0 SR4 FP2、またはそれ以降。

FTPS プロトコルおよびランタイム環境の以下のフィーチャーはサポートされていません。

- 追加のセキュリティー・データ交換のための **ADAT** コマンドの使用。
- チャンネル暗号化のみ (つまりサーバーの証明書が妥当性検査されていない) の FTPS の使用。
- **PROT** コマンドを使用して、Clear、Secure、または Confidential の保護レベルを選択します。
- **MIC** コマンド、**CONF** コマンド、および **ENC** コマンドを使用した、各コマンドの暗号化。
- サーバーが明示的な FTPS をサポートしていない場合の FTP プロトコルへのフォールバック。そのようなサーバーを使用する場合は、プロトコル・ブリッジによって提供されている FTP サポートを使用してください。
- **FEAT** コマンドを使用した、使用可能な FTPS サーバー機能の判別。
- DN フィールドに対するパターン・マッチングを使用した、証明書の妥当性検査。
- 証明書の失効の検査。
- 信頼できる認証局が発行元になっている証明書の妥当性検査。
- セッション確立時の SSL ネゴシエーション・フェーズで使用可能な暗号スイートの明示的な選択。
-  オペレーティング・システムに暗号方式を組み込む、z/OS  または IBM i に固有の拡張機能の使用。特に、データ・セットなどの、鍵情報や信用情報を保管するための、z/OS 鍵リングまたは階層化されていないファイル・システムの使用。暗号ハードウェアおよびオフロード・エンジンが JVM によってトランスペアレントに管理されており、明示的なアプリケーション・コードを必要としない場合には、それらの機能が使用されます。

関連概念

214 ページの『プロトコル・ブリッジ』

プロトコル・ブリッジを使用すれば、Managed File Transfer (MFT) ネットワークから、MFT ネットワークの外部 (ローカル・ドメインとリモート・ロケーションの両方) にあるファイル・サーバーに格納されているファイルにアクセスできます。このファイル・サーバーでは、FTP、FTPS、または SFTP ネットワーク・プロトコルを使用できます。それぞれのファイル・サーバーで少なくとも 1 つの専用エージェントが必要です。この専用エージェントは、プロトコル・ブリッジ・エージェントとして知られています。ブリッジ・エージェントは、複数のファイル・サーバーと相互作用できます。

プロトコル・ブリッジによる SFTP サーバーのサポート

プロトコル・ブリッジは、「SSH File Transfer Protocol バージョン 6 ドラフト 13」というタイトルの IETF Internet Draft で定義されている SFTP プロトコルをサポートします。

プロトコル・ブリッジ・エージェントは、SFTP プロトコルを使用してファイル・サーバーに接続する際に以下の暗号をサポートします。

- blowfish-cbc

- 3des-cbc
- aes128-cbc
- aes192-cbc
- aes256-cbc
- aes128-ctr
- aes192-ctr
- aes256-ctr
- 3des-ctr
- arcfour
- arcfour128
- arcfour256

デフォルトでは、プロトコル・ブリッジ・エージェントによって使用される暗号のリストは、aes128-cbc,aes192-cbc,aes256-cbcです。異なる暗号を使用するようにプロトコル・ブリッジ・エージェントを構成する方法については、577ページの『プロトコル・ブリッジ・プロパティ・ファイルのフォーマット』を参照してください。

認証の方式

ProtocolBridgeCredentials.xml ファイル内で単一のユーザーに対して秘密鍵とサーバー・パスワードとともに IBM MQ Managed File Transfer (MFT) プロトコル・ブリッジ・エージェント・コードを指定した場合、デフォルトで MFT プロトコル・ブリッジ・エージェントは接続の確立時に SFTP ファイル・サーバーの必要に応じて両方の認証方式を使用するよう JSch ライブラリーを構成します。

ProtocolBridgeCredentials.xml ファイル内で単一のユーザーに対して秘密鍵とサーバー・パスワードの両方が構成されていても、SFTP ファイル・サーバーがそれらの認証方式のいずれかのみを要求している場合は、MFT プロトコル・ブリッジ・エージェントは、パスワードに基づく認証よりも公開/秘密鍵認証を優先的に使用するよう JSch ライブラリーを構成します。

公開/秘密鍵認証を使用する試みが SFTP ファイル・サーバーによってリジェクトされた場合、MFT プロトコル・ブリッジ・エージェントは JSch ライブラリーを使用してユーザー名とパスワードに基づく認証を試みます。

これらの認証がいずれか 1 つだけ成功すれば、SFTP ファイル・サーバーに対する接続が確立されます。

MFT プロトコル・ブリッジ・エージェントと関連付けられている ProtocolBridgeCredentials.xml ファイルに対して秘密鍵とパスワードの両方の認証を構成するには、以下を指定する必要があります。

- MFT ユーザー名をプロトコル・サーバーのユーザー名にマップするエレメントの **serverPassword** 属性 (および関連値)。
- 親エレメントで定義されている MFT ユーザーのエレメント。

例えば、構文は次のようになります。

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

キーボード対話方式

MFT プロトコル・ブリッジ・エージェントは、サード・パーティー・ライブラリーである JSch を使用して SFTP ファイル・サーバーに接続します。ProtocolBridgeCredentials.xml ファイルで秘密鍵が指定されていない場合に、このエージェントがキーボード対話方式を使用して SFTP ファイル・サーバーとの認証を試行することができるように、JSch ライブラリーを構成することができます。

キーボード対話方式を使用した認証は、SFTP ファイル・サーバーが password: スtringを使用してパスワード (大文字、または大/小文字混合のいずれか) を求めるプロンプトを出す場合にのみ機能するという

ことに注意してください。キーボード対話 認証方式を使用しているときに、SFTP ファイル・サーバーが password: 以外のストリングで応答すると、その接続の試みは失敗します。

最初の接続の試みに対して SFTP ファイル・サーバーがこのストリングで応答すると、プロトコル・ブリッジ・エージェントは、JSch ライブラリーを使用して、ProtocolBridgeCredentials.xml ファイル内の user エレメントの **serverPassword** 属性で構成されているパスワードを送信します。

関連情報

[プロトコル・ブリッジ](#)

MFT の FIPS サポート

Managed File Transfer は、エージェント、コマンド、および IBM MQ Explorer からキュー・マネージャーへのクライアント接続での、FIPS に準拠した暗号化モジュールの使用をサポートしています。キュー・マネージャーへのどの SSL 接続でも、TLS プロトコルのみが使用されます。鍵ストア・タイプは、JKS と PKCS#12 がサポートされています。

エージェント、調整キュー・マネージャー、またはコマンド・キュー・マネージャーを FIPS に対応させるかどうかは、以下のようにして指定します。

- 特定のエージェントを FIPS に対応させる場合は、そのエージェントの `agent.properties` ファイルで、適切な `agentSsl` プロパティを設定します。詳細内容は [を参照してください](#)。
- 特定の調整キュー・マネージャーを FIPS に対応させる場合は、その調整キュー・マネージャーの `coordination.properties` ファイルで、適切な `coordinationSsl` プロパティを設定します。詳細内容は [を参照してください](#)。
- 特定のコマンド・キュー・マネージャーを FIPS に対応させる場合は、そのコマンド・キュー・マネージャーの `command.properties` ファイルで、適切な `connectionSsl` プロパティを設定します。詳細内容は [を参照してください](#)。

IBM i FIPS は、Managed File Transfer for **IBM i** IBM i ではサポートされません。

プロトコル・ブリッジまたは Connect:Direct ブリッジとの接続は、FIPS に対応していません。

IBM MQ と FIPS について、および必要な構成手順については、[連邦情報処理標準 \(FIPS\)](#) を参照してください。

FIPS 準拠の暗号化モジュールを使用する場合、CipherSuite が FIPS 準拠でなければなりません。FIPS 準拠でない場合、接続は失敗します。IBM MQ でサポートされている CipherSpec については、[IBM MQ classes for Java の SSL/TLS CipherSpec および CipherSuite および IBM MQ classes for JMS の SSL/TLS CipherSpec および CipherSuite](#) を参照してください。

MFT データベース・ロガー表

ロガーをインストールして構成すると、以下のデータベース表が作成されます。

AUTH_EVENT

権限検査に関連したイベント。通常は、特権が不十分だったために要求が拒否された、というイベントです。

- **ID:** 行 ID
- **ACTION:** 実行されたアクションのタイプ。
- **COMMAND_ID:** イベントを要求した元のメッセージの IBM MQ メッセージ ID。転送要求の場合は、これが転送 ID にもなります。
- **TIME:** イベントの発生時刻。
- **ORIGINATOR_MQ_USER:** IBM MQ メッセージに組み込まれている、権限検査が実行されたユーザー ID。
- **AUTHORITY:** 要求された操作で必要であった権限。
- **ORIGINAL_XML_REQUEST:** コマンド・メッセージのペイロード。どの操作が拒否されたのかを確認できます。

- **RESULTCODE:** 結果を示す数字コード。
- **RESULT_TEXT:** 権限イベントの結果を説明するメッセージ。

CALL

Managed File Transfer によって管理される、オペレーティング・システム・コマンド、Ant スクリプト 、または z/OS JCL ジョブのリモート実行。呼び出しを転送に埋め込んだり、call_request 行で参照したりすることができます。

CALL (この表の行) は、通常の転送の一部になっている場合もあれば、スタンドアロンの独立した管理対象呼び出しの場合もあります。通常の転送の一部になっている場合は、TRANSFER_CALLS に基づいて、その呼び出しが TRANSFERS の関連項目にリンクされます。スタンドアロンの独立した管理対象呼び出しが可能なのは、Ant を使用する場合またはメッセージを直接挿入する場合に限られます。スタンドアロンの独立した管理対象呼び出しの場合は、TRANSFERS 表ではなく CALL_REQUEST 表が使用されます。TRANSFER_CALLS に相当する表は、必要ありません。1つの呼び出し要求で実行できる呼び出しは1つだけだからです。

- **ID:** 行 ID
- **COMMAND:** 実行されたコマンド。このフィールドには、コマンドに渡された引数や、コマンドが置かれているパスは含まれません。
- **TYPE:** コマンドのタイプ (Ant や JCL など)。
- **RETRIES:** 要求された再試行の数。
- **RETRY_WAIT:** 最初に要求されていた再試行間の待機間隔 (秒単位)。
- **SUCCESS_RC:** コマンドの正常終了を示す戻りコード。その他のコードが受信される場合、実行は失敗であると報告されます。
- **EXECUTED_COMMAND:** 実行されたコマンドの絶対パス名 (パスを含む)。
- **CAPPED_RETRIES:** 可能な再試行の回数。要求された再試行の回数よりもエージェントの再試行制限値のほうが小さいと、この数は、要求された数より小さくなる可能性があります。
- **CAPPED_RETRY_WAIT:** 使用された再試行の間隔。要求された再試行待機よりもエージェントで構成されている制限値のほうが小さいと、この数は、要求された数より小さくなる可能性があります。
- **OUTCOME:** 呼び出しが全体として成功したかどうか。再試行が複数回行われた場合、各再試行の結果が CALL_RESULT 表に個別に記録されます。

CALL_ARGUMENT

呼び出されるコマンドに提供された引数またはパラメーター。

- **ID:** 行 ID
- **CALL_ID:** 引数が関連付けられる呼び出し。
- **KEY:** 引数はキーと値の対のようなものか、キーそのものか、または名前です。
- **TYPE:** 引数のタイプ。オペレーティング・システム・コマンドに対する位置パラメーターであったり、Ant で使用される名前付きプロパティであったりします。
- **VALUE:** 引数の値。

CALL_REQUEST

ファイル転送の一部ではないコマンド呼び出しの手段。ManagedCall メッセージは、Ant およびダイレクト XML インジェクションを使用して実行依頼できます。

- **ID:** 管理対象呼び出し要求の 16 進 ID。
- **CALL_ID:** この呼び出しについて説明する、CALL 表の行のデータベース ID。
- **ACTION_TIME:** アクションが発生した時刻。
- **AGENT:** コマンドが実行されるエージェント。

- **AGENT_QM:** コマンドが実行されるエージェントによって使用されるキュー・マネージャー。
- **ARCHITECTURE:** エージェントが実行するシステムのマシン体系。
- **OS_NAME:** エージェントが実行しているオペレーティング・システムの名前。
- **OS_VERSION:** オペレーティング・システムのバージョン。
- **ORIGINATOR_HOST:** 呼び出し要求の送信元のマシンのホスト名。
- **ORIGINATOR_USER:** 要求 XML で報告される、呼び出し要求を送信したユーザーの名前。
- **ORIGINATOR_MQ_USER:** 要求の IBM MQ メッセージ記述子に含まれている、呼び出し要求を送信したユーザーの名前。
- **JOB_NAME:** ユーザー指定のジョブ名。
- **RESULTCODE:** 呼び出しの全体的な結果コード。
- **RESULTTEXT:** 呼び出しの全体的な結果メッセージ。

CALL_RESULT

コマンドの呼び出しの詳細結果。再試行が使用可能になっている場合、1回の呼び出しに複数の結果が存在する可能性があります。

- **ID:** 行 ID
- **CALL_ID:** この結果が適用される、CALL 表の行のデータベース ID。
- **SEQUENCE:** 試行が複数回行われた場合、この結果が適用される試行。
- **OUTCOME:** コマンドの結果 (例えば、成功または失敗)。
- **RETURN_CODE:** コマンド戻りコード。
- **TIME:** コマンドが完了した時刻。
- **STDOUT:** コマンドが開始されている場合、コマンドからの標準出力ストリーム。
- **STDERR:** コマンドが開始されている場合、コマンドからの標準エラー・ストリーム。
- **ERROR:** コマンドを開始できなかった場合、問題を説明するエラー・メッセージが Managed File Transfer によって生成されます。

FILE_SPACE_ENTRY

名前が指定されているファイル・スペースに送信されたファイルが各行に表示されます。

- **ID:** ファイル・スペース項目の ID。
- **FILE_SPACE_NAME:** ファイル・スペースの名前。ファイル・スペースの所属先のユーザーの名前です。
- **TRANSFER_ITEM_ID:** この行に関連する転送項目の ID。
- **ALIAS:** このファイル・スペース項目の別名。通常、この別名は、転送のソース・ファイルの名前になります。
- **DELETED:** ファイルがファイル・スペースから削除された時刻。ファイルが削除されていない場合は、null の値になります。

METADATA

転送に関連付けられるメタデータ。

- **ID:** 行 ID
- **TRANSFER_EVENT_ID:** このメタデータが転送に関連する場合、そのメタデータが関連付けられる transfer_event 行。メタデータがスタンドアロン管理対象呼び出しに関連付けられている場合、このフィールドはヌルです。
- **STANDALONE_CALL_ID:** メタデータがスタンドアロン管理対象呼び出しに関連付けられている場合、関係する管理対象呼び出し要求の ID。
- **KEY:** メタデータ項目の名前。

- **VALUE:** メタデータ項目の値。

MONITOR

外部条件に基づいて Managed File Transfer 操作を起動するリソース・モニター。

- **AGENT:** モニターが実行するエージェント。
- **ID:** モニターの 16 進 ID。
- **NAME:** モニターの名前。
- **QMGR:** モニターが実行するエージェントのキュー・マネージャー。

MONITOR_ACTION

各行は、モニターについて発生するアクション (例えば、作成やトリガーなど) を表します。

- **ID:** 行 ID
- **ACTION:** 実行されたアクションのタイプ。
- **JOB_NAME:** 実行依頼されたジョブの名前 (該当する場合)。
- **MONITOR:** このアクションが発生したモニター。存在しないモニターについてアクションが要求されたために、そのアクションが失敗した場合、ヌルになることがあります。
- **ORIGINAL_XML_REQUEST:** このアクションが *create* または *triggerSatisfied* アクションだった場合、モニターの起動時に開始される XML 要求。
- **ORIGINATOR_MQ_USER:** アクションを開始した IBM MQ メッセージに含まれるユーザー ID。
- **ORIGINATOR_USER:** アクションの実行要求を送信したユーザー名。
- **ORIGINATOR_HOST:** ユーザーがアクションの実行要求を送信したマシン。
- **TIME:** アクションが発生した時刻。
- **UPDATED_XML_REQUEST:** アクションが *triggerSatisfied* である場合、開始された XML 要求。この要求は、変数置換のため、最初に出された XML 要求とは異なる可能性があります。

MONITOR_EXIT_RESULT

リソース・モニター出口の実行の結果。

- **ID:** 行 ID
- **ACTION_ID:** 結果が関連付けられるモニター・アクション。
- **EXIT_NAME:** この結果を生成した出口の名前。
- **RESULTCODE:** 出口から返された値 (cancel または proceed のいずれか)。
- **RESULTTEXT:** 出口からのテキスト出力 (提供されている場合)。

MONITOR_METADATA

リソース・モニターに関連付けられるメタデータの項目。

- **ID:** 行 ID
- **ACTION_ID:** メタデータが関連付けられる `monitor_action`。
- **KEY:** メタデータ項目の名前。
- **PHASE:** このメタデータ項目が最初に送信されたデータを表すのか、それとも変数置換後に更新されたバージョンを表すのか。
- **VALUE:** メタデータ項目の値。

SCHEDULE

エージェントに登録された転送スケジュール。

- **AGENT:** このスケジュールがあるエージェントの名前。
- **CREATION_DATE:** このスケジュールが作成された時点。
- **ID:** スケジュールの固有のデータベース (エージェントではない) ID。
- **ID_ON_AGENT:** エージェントがデータベース ID として使用する ID。この ID はエージェント間で固有ではなく、エージェントの永続状態がリセットされた場合でもエージェントにおいて固有でない場合があります。
- **LATEST_ACTION:** このスケジュールの状態を変更した最新のアクション。

SCHEDULE_ACTION

スケジュール状態を変更するイベントが発生すると、アクションが記録されます。

- **ACTION_TYPE:** 発生したアクション。
- **ID:** 行 ID
- **ORIGINATOR_HOST:** 変更の原因となった要求の実行依頼元のマシン。
- **ORIGINATOR_USER:** 変更の原因となった要求が実行依頼されたときの名前を持つユーザー。
- **SCHEDULE_ID:** このアクションが適用されるスケジュール。
- **SPEC_AFTERWARDS:** アクション発生後のこのスケジュールの状態を表す `schedule_spec`。
- **STATUS_CODE:** アクションの結果を説明する数値戻りコード。
- **STATUS_TEXT:** アクションの結果のテキスト記述。通常、アクションが成功した場合はヌルです。
- **TIME:** アクションが発生した時点。

SCHEDULE_SPEC

スケジュール済みの個々の転送の詳細。

- **ID:** 行 ID
- **DESTINATION_AGENT:** ファイルの転送先のエージェント。
- **DESTINATION_QM:** 宛先エージェントによって使用されるキュー・マネージャー。
- **REPEAT_COUNT:** スケジュールが繰り返しており、それが終了時刻ではなく発生回数に結びつけられている場合、繰り返しの回数。
- **REPEAT_FREQUENCY:** スケジュールされた転送間に存在する `repeat_intervals` の数。
- **REPEAT_INTERVAL:** 転送が繰り返す場合の、繰り返しの間隔 (例えば、分または週)。
- **SOURCE_AGENT:** ファイルの転送元のエージェント。
- **SOURCE_QM:** ソース・エージェントによって使用されるキュー・マネージャー。
- **START_TIME:** スケジュール内の最初の転送が行われる時刻。
- **START_TIMEBASE:** 転送に関連した時刻の時間基準。例えば、エージェントのタイム・ゾーンで操作するか、管理者のタイム・ゾーンで操作するかを示します。
- **START_TIMEZONE:** 時間基準が対応付けられ、スケジュールの操作で使用されるタイム・ゾーン。

SCHEDULE_ITEM

各ファイル (または転送時間に一致するパターン) は `schedule_item` で表されます。

- **ID:** 行 ID
- **CHECKSUM_METHOD:** ファイルのチェックサムの計算方法。
- **DESTINATION_EXISTS_ACTION:** ファイルが宛先に既に存在する場合に宛先エージェントが実行するアクション。
- **DESTINATION_FILENAME:** ファイルの転送先のファイルまたはディレクトリ。
- **DESTINATION_QUEUE:** ファイルからメッセージへの転送のための宛先キュー名。

- **Multi** **DESTINATION_TYPE:** destination_filename 列がファイルまたはディレクトリーのどちらを参照するか。
- **z/OS** **DESTINATION_TYPE:** destination_filename 列がファイル、ディレクトリー、またはデータ・セットのどれを参照するか。
- **FILE_MODE:** ファイルが転送されるモード (例えば、テキストまたはバイナリー)。
- **RECURSIVE:** エージェントがスケジュールに従って転送を作成する場合、エージェントがソース・ディレクトリーを繰り返すか (Y)、またはそうでないか (N)。
- **SCHEDULE_SPEC_ID:** この項目が関連付けられる schedule_spec。
- **SOURCE_DISPOSITION:** 転送の完了後にソース・ファイルで実行するアクション。
- **SOURCE_FILENAME:** ソース・ファイル、ディレクトリー名、またはパターン。
- **SOURCE_QUEUE:** メッセージからファイルへの転送のためのソース・キュー名。

TRANSFER

1 つ以上のファイルの 1 回の転送。

- **TRANSFER_ID:** 転送の 16 進 ID。
- **JOB_NAME:** 転送のユーザー指定のジョブ名。
- **SCHEDULE_ID:** この転送がスケジュールの結果である場合、関係するスケジュールのデータベース行 ID。
- **START_ID:** 転送の開始を表す transfer_event の行 ID。
- **COMPLETE_ID:** 転送の終了を表す transfer_event の行 ID。
- **RESULTCODE:** 転送の全体的な結果コード。この列に指定できる値が次のトピックにリストされています: 338 ページの『MFT の戻りコード』。これらのコードは、転送全体に適用されます。個々の各項目の状況については、[TRANSFER_ITEM.RESULTCODE](#) を参照してください。
- **RESULTTEXT:** 転送の全体的な結果テキスト (存在する場合)。
- **STATUS:** 転送の状況。この列の有効な値は、started、success、partial success、failure、cancelled です。
- **RELATED_TRANSFER_ID:** この転送に関連した、前の転送の 16 進 ID。例えば、転送がファイル・ダウンロードである場合、このフィールドはファイルをアップロードした転送を参照します。

TRANSFER_CALLS

実行可能コマンド呼び出しを転送にリンクします。

- **ID:** 行 ID
- **POST_DESTINATION_CALL:** 転送の完了後に宛先で行われる呼び出し。
- **POST_SOURCE_CALL:** 転送の完了後にソース・エージェントで行われる呼び出し。
- **PRE_DESTINATION_CALL:** 転送の開始前に宛先エージェントで行われる呼び出し。
- **PRE_SOURCE_CALL:** 転送の開始前にソース・エージェントで行われる呼び出し。
- **TRANSFER_ID:** この行の中の呼び出しが関連付けられる転送。

TRANSFER_CD_NODE

転送で使用する Connect:Direct ノードに関する情報。

- **PNODE:** 転送の 1 次ノード。
- **SNODE:** 転送の 2 次ノード。
- **BRIDGE_IS_PNODE:** Connect:Direct ブリッジの一部であるノードを示す文字。この値が Y の場合は、1 次ノードがブリッジ・ノードです。この値が N の場合は、2 次ノードがブリッジ・ノードです。
- **ID:** この行の ID。

TRANSFER_CORRELATOR

転送項目に関連する相関ストリングと相関数値が各行に入ります。

- **CORRELATION_BOOLEAN:** ブール相関値。真の場合は Y、偽の場合は N という 1 文字が表示されます。
- **CORRELATION_STRING:** ストリング相関値。
- **CORRELATION_NUMBER:** 数値相関値。
- **ID:** この行の ID。

TRANSFER_EVENT

転送に関連するイベント (開始または終了)。

- **ID:** 行 ID
- **ACTION_TIME:** 転送アクションが行われた時刻。
- **SOURCE_AGENT:** ファイルの転送元のエージェントの名前。
- **SOURCE_AGENT_TYPE:** ファイルの転送元のエージェントのタイプ。有効な値は、1 = STANDARD、2 = BRIDGE、3 = WEB_GATEWAY、4 = EMBEDDED、5 = CD_BRIDGE、6 = SFG です。
注: IBM MQ 9.0 から、Managed File Transfer は Web ゲートウェイや Web エージェントをサポートしなくなりました。
- **SOURCE_QM:** ソース・エージェントによって使用されるキュー・マネージャー。
- **SOURCE_ARCHITECTURE:** ソース・エージェントをホスティングするシステムのマシン体系。
- **SOURCE_OS_NAME:** ソース・エージェント・マシンのオペレーティング・システム。
- **SOURCE_OS_VERSION:** ソース・エージェント・マシンのオペレーティング・システムのバージョン。
- **SOURCE_BRIDGE_URL:** ソース・エージェントがプロトコル・ブリッジ・エージェントの場合、ブリッジを形成する対象のデータ・ソースの URL。
- **SOURCE_CD_NODE_ID:** 転送のソースである Connect:Direct ノード。
- **DESTINATION_AGENT:** ファイルの転送先のエージェントの名前。
- **DESTINATION_AGENT_TYPE:** ファイルの転送先のエージェントのタイプ。有効な値は、1 = STANDARD、2 = BRIDGE、3 = WEB_GATEWAY、4 = EMBEDDED、5 = CD_BRIDGE、6 = SFG です。
注: IBM MQ 9.0 から、Managed File Transfer は Web ゲートウェイや Web エージェントをサポートしなくなりました。
- **DESTINATION_QM:** 宛先エージェントによって使用されるキュー・マネージャー。
- **DESTINATION_BRIDGE_URL:** 宛先エージェントがブリッジ・エージェントの場合、ブリッジを形成する対象のデータ・ソースの URL。
- **DESTINATION_CD_NODE_ID:** 転送の宛先である Connect:Direct ノード。
- **ORIGINATOR_HOST:** 転送要求の送信元のマシンのホスト名。
- **ORIGINATOR_USER:** `fteCreateTransfer` コマンドによって報告される、転送要求を送信したユーザーの名前。
- **ORIGINATOR_MQ_USER:** 要求の IBM MQ メッセージ記述子に含まれている、転送要求を送信したユーザーの名前。
- **TRANSFERSET_TIME:** 転送セットが作成された時刻。
- **TRANSFERSET_SIZE:** 転送中の項目の数。
- **TRIGGER_LOG:** トリガーを含む転送定義の場合、転送が行われなかったトリガー評価をログに記録するかどうか。

TRANSFER_EXIT

ファイル転送の一環として実行された転送出口が各行に表示されます。

- **ID:** 行 ID

- **EXIT_NAME:** 出口の名前。
- **TRANSFER_ID:** この出口の適用対象の転送の ID。転送が完了した場合もあれば、取り消された場合もあります。
- **TYPE:** 出口のタイプ。 *SourceStart*、 *SourceEnd*、 *DestinationStart*、 *DestinationEnd* のいずれかの値になります。
- **STATUS:** 出口から返された値。 *cancel* または *proceed* のいずれかになります。
- **SUPPLEMENT:** 出口の状況を説明するオプション・メッセージ。

TRANSFER_ITEM

各行は、転送の一部として送信されるファイルを表します。

- **DESTINATION_CHECKSUM_METHOD:** 宛先ファイルのチェックサムを計算するために使用されるアルゴリズム。転送が正常に完了しなかったためにチェックサムが計算されない場合、ヌルになることがあります。
- **DESTINATION_CHECKSUM_VALUE:** 宛先ファイルのチェックサムの値。チェックサムが使用不可になっている場合、値はヌルになることがあります。
- **DESTINATION_ENCODING:** 宛先ファイルがテキストとして転送される場合、宛先ファイル上で使用される文字エンコード。
- **DESTINATION_EXISTS_ACTION:** ファイルが宛先に存在する場合に実行するアクション。
- **DESTINATION_FILE_SIZE:** 宛先で使用するファイル名 `z/OS` またはデータ・セット名のサイズ。
- **DESTINATION_FILENAME:** 宛先で使用するファイル名 `z/OS` またはデータ・セット名。
- **DESTINATION_LINEEND:** 宛先ファイルがテキストとして転送される場合、宛先ファイルで使用される行終了形式。
- **DESTINATION_MESSAGE_QUEUE_NAME:** ファイルからメッセージへの転送の実行時にソース・ファイルから生成されるメッセージの宛先キュー。
- **DESTINATION_MESSAGE_GROUP_ID:** 複数のメッセージが生成された場合、ファイルからメッセージへの転送の実行時にソース・ファイルから生成されるメッセージで使用されるグループ ID。
- **DESTINATION_MESSAGE_MESSAGE_ID:** 1つのメッセージのみが生成された場合、ファイルからメッセージへの転送の実行時にソース・ファイルから生成されるメッセージのメッセージ ID。
- **DESTINATION_MESSAGE_COUNT:** ファイルからメッセージへの転送の実行時にソース・ファイルが各メッセージに分割されたときのメッセージの数。
- **DESTINATION_MESSAGE_LENGTH:** ファイルからメッセージへの転送の実行時にソース・ファイルから生成されるメッセージの長さ(バイト単位)。この値が設定されるのは、`fteCreateTransfer` コマンドの `-qs` オプションを使用するなどして、出力メッセージの長さを指定した場合に限られます。`-qs 20K` を指定した場合に、ソース・ファイルのサイズが 50 KB であれば、3つのメッセージが生成され、それぞれのサイズは、20 KB、20 KB、10 KB になります。その場合、`DESTINATION_MESSAGE_LENGTH` の値は、20480 に設定されます。
- **DESTINATION_CORRELATOR_ID:** 宛先の相関関係子情報の ID。
- **FILE_MODE:** ファイル転送モード(例えば、テキストまたはバイナリー)。
- **ID:** 行 ID
- **RESULTCODE:** この項目の転送の結果を示す数字コード。この列に指定できる値が次のトピックにリストされています: 344 ページの『ファイル転送の戻りコード』。これらのコードは転送内の個々の項目に適用されます。転送全体の結果については、『`TRANSFER.RESULTCODE`』を参照してください。
- **RESULT_TEXT:** 転送の結果のテキスト説明。通常、転送が成功した場合はヌルです。
- **SOURCE_CHECKSUM_METHOD:** ソース・ファイルのチェックサムを計算するために使用されるアルゴリズム。
- **SOURCE_CHECKSUM_VALUE:** ソース・ファイルのチェックサムの値。チェックサムが使用不可になっている場合、値はヌルになることがあります。
- **SOURCE_DISPOSITION:** 転送の完了時にソース・ファイルで実行するアクション。

- **SOURCE_ENCODING:** ソース・ファイルがテキストとして転送される場合、ソース・ファイル上で使用される文字エンコード。
- **SOURCE_FILE_SIZE:** ソースで使用するファイル名 **z/OS** またはデータ・セット名のサイズ。
- **SOURCE_FILENAME:** ソース・ファイル名 **z/OS** またはデータ・セット名。
- **SOURCE_LINEEND:** ソース・ファイルがテキストとして転送される場合、ソース・ファイルで使用される行終了形式。
- **SOURCE_MESSAGE_QUEUE_NAME:** メッセージからファイルへの転送の実行時に宛先ファイルに組み込まれるメッセージのソース・キュー。
- **SOURCE_MESSAGE_GROUP_ID:** メッセージからファイルへの転送の実行時に宛先ファイルに組み込まれるメッセージのグループ ID。
- **SOURCE_MESSAGE_COUNT:** メッセージからファイルへの転送の実行時に宛先ファイルに組み込まれるメッセージの数。
- **SOURCE_CORRELATOR_ID:** ソースの相関関係子情報の ID。
- **TRANSFER_ID:** この項目が構成の一部となっている転送。
- **TRUNCATE_RECORDS:** 長さ超過データ・セット・レコードを切り捨てるか、折り返すかを示します。

TRANSFER_STATS

転送の終了時に生成される一連の統計。

- **ID:** 行 ID
- **TRANSFER_ID:** 統計の参照先の転送。
- **START_TIME:** 転送の開始時刻。 ビジー状態のシステムまたは接続が断続的になっているシステムでは、Started メッセージで報告される時刻よりも遅い時刻になる場合があります。そのメッセージで報告されるのは、成功したデータ転送が始まった時点ではなく初期の処理が始まった時点での時刻です。
- **RETRY_COUNT:** ロードまたは可用性の問題が原因で転送が再試行された回数。
- **FILE_FAILURES:** 転送に失敗したファイルの数。
- **FILE_WARNINGS:** 転送時に警告が報告されたファイルの数。

TRIGGER_CONDITION

基本的な Managed File Transfer 条件付き転送の 1 つの条件。例えば、"file example.file exists" のようになります。

- **ID:** 行 ID
- **TRANSFER_EVENT_ID:** トリガーが関連している転送イベント。
- **CONDITION_TYPE:** トリガーで使用される検査のタイプ。例えば、ファイルの存在またはファイルのサイズ。
- **COMPARISON:** 実行する特定の比較。例えば、「より大か等しい」。
- **VALUE:** 比較対象の値。
- **FILENAME:** 調べるファイル名。

関連概念

106 ページの『MFT ロガーの構成』

関連資料

530 ページの『fteStartLogger』

fteStartLogger コマンドは、Managed File Transfer ロギング・アプリケーションを開始します。

492 ページの『fteModifyLogger (Windows サービスとしての MFT ロガーの実行)』

Managed File Transfer ロガーを Windows サービスとして実行できるように変更するには、

fteModifyLogger コマンドを使用します。このコマンドは Windows プラットフォームでのみ使用でき

ます。IBM MQ 管理者であり、mqm グループのメンバーであるユーザーが実行する必要があります。まず、**fteStopLogger** コマンドを使用してロガーを停止する必要があります。

533 ページの『fteStopLogger』

fteStopLogger コマンドは Managed File Transfer ロガーを停止します。

MFT ロガーの権限

ロガーを実行するオペレーティング・システム・ユーザーには、ロガー・キューと SYSTEM.FTE トピックに対する特定の IBM MQ 権限が必要です。

ロガーを実行するオペレーティング・システム・ユーザーには、以下の IBM MQ 権限が必要です。

- 調整キュー・マネージャーに対する CONNECT および INQUIRE 権限。
- SYSTEM.FTE トピックに対する SUBSCRIBE 権限。
- SYSTEM.FTE.LOG.RJCT.*logger_name* キューに対する PUT 権限。
- SYSTEM.FTE.LOG.CMD.*logger_name* キューに対する GET 権限。

関連資料

353 ページの『MFT 固有リソースのグループ権限の制限』

関連する可能性のあるさまざまなオブジェクトのすべてについて、個々のユーザーに権限を付与する代わりに、Managed File Transfer のアクセス制御を管理する目的で、2つのセキュリティー・グループ FTEUSER と FTEAGENT を構成します。これらのグループの作成と埋め込みは、IBM MQ 管理者が行う責任があります。管理者は、ここで説明される構成の提案を拡大または変更する選択ができます。

359 ページの『MFT エージェント・アクションのユーザー権限の制限』

グループを使用してリソースに対するアクセスを管理することに加えて、追加レベルのセキュリティーを使用可能にして、ユーザーが実行できる Managed File Transfer エージェント・アクションを制限することができます。エージェントの権限キューに対する権限をユーザーに付与して、ユーザーが特定のエージェント・アクションを実行できるようにします。

宛先ファイルのファイル・アクセス許可

Managed File Transfer 宛先エージェントによって書き込まれる宛先ファイルのファイル・アクセス許可は、エージェントが実行されているプラットフォームによって決定されます。

z/OS、UNIX、および Linux のプラットフォームでの宛先エージェント



システムの **umask** の値を変更する必要があります。

例えば、ご使用の z/OS システムのユーザー ID に対するデフォルトの **umask** 値が 0022 であるとして。

MFT エージェントがこのユーザーとして実行され、宛先ファイルを書き込む場合、そのファイルには以下の権限があります。

```
-IW-I--I--
```

umask 値を、例えばコマンドを実行してからエージェントを再始動することにより 0006 に変更した場合、

```
umask 0006
```

エージェントが書き込むすべての宛先ファイルには以下の権限があります。

```
-IW-IW----
```

umask コマンドの実行後にエージェントが新しい値を取得するためには、エージェントを再始動する必要があります。ことに注意してください。

ここでは例として z/OS を使用しましたが、同じ情報は UNIX および Linux のプラットフォームにも適用されます。

Windows 上の宛先エージェント

Windows

デフォルトでは、権限はルート・フォルダーからその下のファイルおよびサブフォルダーに継承されます。ただし、この継承はオフにすることができます。

Windows 管理者またはドメイン管理者は、権限を確認および管理し、必要に応じて変更する必要があります。これらの管理者は `icalcs` コマンドを使用して、権限を表示、追加、更新、および削除することができます。

関連情報

[MFT 固有リソースのグループ権限の制限](#)

[MFT エージェント・アクションのユーザー権限の制限](#)

MFT が宛先キューに書き込むメッセージで設定する MQ メッセージ・プロパティ

ファイルからメッセージへの転送を実行する場合、Managed File Transfer では、宛先キューに書き込まれる最初のメッセージで、IBM MQ メッセージ・プロパティを設定できます。その他の IBM MQ メッセージ・プロパティは、ファイルからメッセージへの転送が失敗したときに設定されます。

IBM MQ メッセージ・プロパティを使用すれば、アプリケーションで MQ メッセージ記述子 (MQMD) または MQRFH2 ヘッダーにアクセスしなくても、処理対象のメッセージを選択したり、メッセージに関する情報を取得したりすることが可能になります。 [メッセージ・プロパティ](#) を参照してください。

このトピックでは、宛先キューに書き込まれる最初のメッセージにメッセージ・プロパティを追加する必要があることを示すために、`fteCreateTransfer` コマンドおよび `fteCreateTemplate` コマンドで使用されるパラメーターについて説明します。`fte:filespec` パラメーターの `dstmsgprop` 値を使用して、宛先キューに書き込まれる最初のメッセージにメッセージ・プロパティを追加するように指定することもできます。

標準プロパティ

`fteCreateTransfer` コマンドまたは `fteCreateTemplate` コマンドで `-qmp` パラメーターを使用して、転送によって宛先キューに書き込まれる最初のメッセージに IBM MQ メッセージ・プロパティを設定するかどうかを指定できます。このパラメーターの使用例については、203 ページの『例: ファイルからメッセージへの転送に関する IBM MQ メッセージ・プロパティの設定』のトピックを参照してください。

IBM MQ メッセージ・プロパティに転送のメタデータが格納されます。メッセージ・プロパティ名には、`usr.WMQFTE` という接頭部が付きます。`usr.` という接頭部が付いているメッセージ・プロパティは、JMS アプリケーションでの利用が可能です。

`usr.WMQFTETransferId`

固有の 16 進数転送 ID。

`usr.WMQFTETransferMode`

ファイル転送のタイプ (バイナリー・モードまたはテキスト・モード)。

`usr.WMQFTESourceAgent`

ソース・エージェントの名前。

`usr.WMQFTEDestinationAgent`

宛先エージェントの名前。

`usr.WMQFTEFileName`

ソース・ファイルの名前。

`usr.WMQFTEFileSize`

ソース・ファイルのサイズ (バイト単位)。

`usr.WMQFTEFileLastModified`

ソース・ファイルの最終変更日時。1970 年 1 月 1 日の 00:00:00 UTC を基点にしたミリ秒単位の値です。

usr.WMQFTEFileIndex

転送中のファイルのリストの現行ファイルの索引。リストの最初のファイルの索引は 0 です。

usr.WMQFTEmqmdUser

転送要求を実行依頼したユーザーの MQMD ユーザー ID。

障害プロパティ

ファイルからメッセージへの転送で、宛先エージェントが少なくとも 1 つのメッセージを宛先キューに書き込んだ後に障害が発生すると、Managed File Transfer は宛先キューにブランク・メッセージを書き込みます。 **-qmp** パラメーターが true に設定されていると、このブランク・メッセージで 2 つの IBM MQ メッセージ・プロパティが設定されます。ファイルからメッセージへの転送で発生する障害の例については、205 ページの『[ファイルからメッセージへの転送の失敗](#)』を参照してください。

ファイルからメッセージへの転送が完全に失敗する場合、Managed File Transfer は宛先キューにブランクのメッセージを書き込みます。 **-qmp** パラメーターが true に設定されていて、メッセージ・データの長さが `maxInputOutputMessageLength` の値を超えている場合には、以下のエラー・メッセージがコマンド行に表示されます。

```
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 メッセージ・プロパティには、障害に関する情報が格納されます。標準メッセージ・プロパティの場合と同じく、メッセージ・プロパティ名には、**usr.WMQFTE** という接頭部が付くので、JMS アプリケーションでの利用が可能になります。

usr.WMQFTEReturnCode

転送の戻りコード。この戻りコードの有効な値のリストについては、[338 ページの『MFT の戻りコード』](#)のトピックを参照してください。

usr.WMQFTESupplement

転送が失敗した理由をさらに詳しく記述した補足メッセージ。

ユーザー定義のプロパティ

fteCreateTransfer コマンドで **-md** パラメーターを使用して指定されるメタデータは、IBM MQ メッセージのプロパティとして設定できます。 **-qmp** パラメーターが true に設定されていると、ユーザーが指定するすべてのメタデータは最初のメッセージのメッセージ・ヘッダーに追加されます。

メタデータ名には、**usr.** という接頭部が付きます。例えば、メタデータが `department=accounts` である場合、IBM MQ メッセージ・ヘッダーは `usr.department=accounts` に設定されます。

メタデータを使用して、`usr.WMQFTE` または `usr.com.ibm.wmqfte` で始まるヘッダーを指定することはできません。 `WMQFTE` または `com.ibm.wmqfte` で始まる名前を持つメタデータを指定した場合、このメタデータはメッセージ・プロパティで使用されずに無視されます。

関連概念

[198 ページの『ファイルからメッセージへのデータ転送』](#)

Managed File Transfer のファイルからメッセージへの転送機能を使用すれば、1 つのファイルにあるデータを IBM MQ のキューにある 1 つまたは複数のメッセージに転送できます。

関連タスク

[203 ページの『例: ファイルからメッセージへの転送に関する IBM MQ メッセージ・プロパティの設定』](#)

fteCreateTransfer コマンドの **-qmp** パラメーターを使用して、転送によって宛先キューに書き込まれる最初のメッセージに IBM MQ メッセージ・プロパティを設定するかどうかを指定できます。IBM MQ メッセージ・プロパティを使用すれば、アプリケーションで IBM MQ メッセージ記述子 (MQMD) または MQRFH2 ヘッダーにアクセスしなくても、処理対象のメッセージを選択したり、メッセージに関する情報を取得したりすることが可能になります。

関連資料

679 ページの『MFT がソース・キューのメッセージから読み取る IBM MQ メッセージ・プロパティー』
メッセージからファイルへの転送でソース・キューからメッセージを読み取っているエージェントは、メ
ッセージから IBM MQ メッセージ・プロパティーを読み取ります。これらのプロパティーの値を使用して、
転送の動作を決定することができます。

338 ページの『MFT の戻りコード』

Managed File Transfer コマンド、Ant タスク、およびログ・メッセージは、機能が正常に実行されたかど
うかを示す戻りコードを提供します。

205 ページの『ファイルからメッセージへの転送の失敗』

ファイルからメッセージへの転送で、エージェントがファイル・データを宛先キューに書き込み始めた後
に障害が発生すると、エージェントは、メッセージをコンシュームするアプリケーションに障害の発生を
通知するためのメッセージをキューに書き込みます。

437 ページの『fteCreateTransfer: 新規ファイル転送の開始』

fteCreateTransfer コマンドは、コマンド行から新規のファイル転送を作成して開始します。このコマ
ンドは、ファイル転送の即時開始、将来の日時におけるファイル転送のスケジュール、ファイル転送の 1
回以上の繰り返し、および特定の条件に基づくファイル転送の起動を実行できます。

862 ページの『fte:filespec Ant のネストされたエレメント』

fte:filespec パラメーターは、その他のタスクでネストされたエレメントとして使用されます。

fte:filespec を使用すると、1 つ以上のソース・ファイル、ディレクトリー  またはデータ・
セット、および宛先間のマッピングを記述できます。一般的には、このエレメントは、移動またはコピー
する一連のファイル、ディレクトリー 、またはデータ・セットを表す場合に使用されます。

MFT がソース・キューのメッセージから読み取る IBM MQ メッセージ・プロパ ティー

メッセージからファイルへの転送でソース・キューからメッセージを読み取っているエージェントは、メ
ッセージから IBM MQ メッセージ・プロパティーを読み取ります。これらのプロパティーの値を使用して、
転送の動作を決定することができます。

メッセージからファイルへの転送を取り消すために使用するヘッダー

グループのメッセージからファイルへの転送を取り消すには、そのグループの最後のメッセージに次の
IBM MQ メッセージ・プロパティーを設定します。

usr.UserReturnCode

必須。転送の戻りコード。転送を取り消すことを示すために、このヘッダーをゼロ以外の値として設
定します。

usr.UserSupplement

オプション。転送が取り消された理由を記述するテキスト。

メッセージからファイルへの転送のソース・エージェントが、ソース・キューから **usr.UserReturnCode**
メッセージ・プロパティーをゼロ以外の値に設定したメッセージを読み取ると、キューからのメッセ
ージの読み取りを停止して、転送が失敗したことを転送ログ XML で報告します。転送ログ XML には、戻りコ
ード、およびメッセージ・ヘッダーで設定された補足テキストが含まれています。宛先エージェントが既
にデータを一時ファイルに書き込んでいる場合、このファイルは宛先から削除されます。

変数置換によって使用されるヘッダー

モニター対象キューから読み取られた最初のメッセージに含まれている IBM MQ メッセージ・プロパ
ティーの値は、タスク XML 定義に代入できます。ユーザー定義メッセージ・プロパティーには、接頭部 **usr.**
が付きますが、変数名にはこの接頭部を含めません。変数名は、中括弧 ({}) で囲んで、その前にドル記号
(\$) 文字を付加する必要があります。例えば、**#{destFileName}** は、ソース・キューから読み取られる最初
のメッセージの **usr.destFileName** メッセージ・プロパティーの値に置き換えられます。

例えば、モニターされるキューにメッセージを書き込むユーザーまたはプログラムは、グループの最初の
メッセージに IBM MQ メッセージ・プロパティーを設定して、ファイル転送の宛先として使用されるエ
ージェント、およびデータを転送するファイル名を指定することができます。

詳しくは、186 ページの『キューのモニターおよび変数置換の使用』を参照してください。

メッセージ・サイズに関連する MQ 属性および MFT プロパティを設定する際のガイダンス

さまざまなサイズのメッセージを読み書きする際の Managed File Transfer の動作に影響を与えるように、IBM MQ 属性および Managed File Transfer プロパティを変更することができます。

ソース・キューから読み取るメッセージまたは宛先キューに書き込むメッセージのサイズが 1048576 バイト (1 MB) を超す場合は、Managed File Transfer Agent プロパティ **maxInputOutputMessageLength** の値を、読み書きする最大メッセージ・サイズ以上の値に増やす必要があります。

ソース・キューのメッセージが 1048576 バイトより大きい場合は、ソース・エージェントに **maxInputOutputMessageLength** プロパティを設定する必要があります。宛先キューのメッセージが 1048576 バイトより大きい場合は、宛先エージェントに **maxInputOutputMessageLength** プロパティを設定する必要があります。 **maxInputOutputMessageLength** プロパティについては、[拡張エージェント・プロパティ](#)を参照してください。

- エージェントが読み書きするキューがエージェント・キュー・マネージャーに対してローカルである場合、IBM MQ キュー・マネージャー、キュー、およびチャンネルの **MAXMSGL** 属性の変更が必要な場合があります。

ソース・キューまたは宛先キューの最大メッセージ・サイズの値が、**maxInputOutputMessageLength** エージェント・プロパティの値以上であることを確認します。

以下の IBM MQ 属性の各値 (バイト単位) について、

- エージェント・キュー・マネージャーの最大メッセージ・サイズ
- `SYSTEM.FTE.STATE.agent_name` キューの最大メッセージ・サイズ
- クライアント・チャンネルの最大メッセージ・サイズ (使用中のエージェントがクライアント・モードでキュー・マネージャーに接続している場合)

次の計算の結果以上であることを確認します。

ファイルからメッセージへの転送 (サポートされるファイル・サイズの最大値は **100 MB**) の場合:
maxInputOutputMessageLength の値

メッセージからファイルへの転送の場合:

$3 * (\text{maxInputOutputMessageLength}) + 1048576$ の値

(この計算は、1 つの状態メッセージに 3 つのチェックポイントを保管でき、各チェックポイントはデータのメッセージ量の最大サイズまでバッファーに入れる必要があるかもしれない、という事実から導き出されたものです。)

- エージェントの書き込み先のキューがリモート・キューである場合、IBM MQ キュー・マネージャー、キュー、およびチャンネルの **MAXMSGL** 属性の変更が必要な場合があります。

以下の各 IBM MQ 属性の値が、**maxInputOutputMessageLength** エージェント・プロパティの値以上であることを確認します。

- エージェント・キュー・マネージャーのリモート・キュー・マネージャー伝送キューの最大メッセージ・サイズ
- エージェント・キュー・マネージャーからリモート・キュー・マネージャーへのチャンネルの最大メッセージ・サイズ
- リモート・キュー・マネージャーの宛先キューの最大メッセージ・サイズ
- リモート・キュー・マネージャーの最大メッセージ・サイズ

以下の IBM MQ 属性の各値 (バイト単位) について、

- エージェント・キュー・マネージャーの最大メッセージ・サイズ

- SYSTEM.FTE.STATE.agent_name キューの最大メッセージ・サイズ
- クライアント・チャンネルの最大メッセージ・サイズ (使用中のエージェントがクライアント・モードでキュー・マネージャーに接続している場合)

次の計算の結果以上であることを確認します。

ファイルからメッセージへの転送 (サポートされるファイル・サイズの最大値は 100 MB) の場合:
 $\text{maxInputOutputMessageLength}$ の値

メッセージからファイルへの転送の場合:

$3 * (\text{maxInputOutputMessageLength}) + 1048576$ の値

(この計算は、1つの状態メッセージに3つのチェックポイントを保管でき、各チェックポイントはデータのメッセージ量の最大サイズまでバッファーに入れる必要があるかもしれない、という事実から導き出されたものです。)

上記のプロパティのいずれかの値を超えた場合、エージェントは停止し、次のエラーがエージェント・イベント・ログに記録されます。

```
BFGUT0002E: An internal error has occurred. Product failure data was captured in file
"FFDC.FTE.20100928170828514.8172766022149157013.log".
BFGSS0025E: An internal error has occurred. The exception is: cc=2 rc=2010 op=put - MQPUT to
SYSTEM.FTE.STATE.agent_name
BFGAG0061E: The agent ended abnormally
```

以下の IBM MQ 理由コードがエージェント・イベント・ログのこのメッセージに含まれている場合があります。

- rc=2010 この理由コードは MQRC_DATA_LENGTH_ERROR に対応し、クライアント・チャンネルの最大メッセージ・サイズの値を超えたことを示します。この問題を解決するには、エージェント・キュー・マネージャーのクライアント・チャンネルの最大メッセージ・サイズが次の計算の結果以上であることを確認します。

$3 * (\text{maxInputOutputMessageLength}) + 1048576$

- rc=2030 この理由コードは MQRC_MSG_TOO_BIG_FOR_Q に対応し、SYSTEM.FTE.STATE.agent_name キューの最大メッセージ・サイズの値を超えたことを示します。この問題を解決するには、SYSTEM.FTE.STATE.agent_name キューの最大メッセージ・サイズが次の計算の結果以上であることを確認します。

$3 * (\text{maxInputOutputMessageLength}) + 1048576$

- rc=2031 この理由コードは MQRC_MSG_TOO_BIG_FOR_Q_MGR に対応し、エージェント・キュー・マネージャーの最大メッセージ・サイズの値を超えたことを示します。この問題を解決するには、エージェント・キュー・マネージャーの最大メッセージ・サイズが次の計算の結果以上であることを確認します。

$3 * (\text{maxInputOutputMessageLength}) + 1048576$

小さいメッセージを多数転送する場合

エージェントがキューとの間で読み書きするメッセージの平均サイズが 1310 バイト未満であり、エージェントが 10000 を超えるメッセージを読み書きする場合、キュー・マネージャーの非コミット・メッセージの最大数属性を大きくするか、チェックポイント間隔のデータ量を削減する必要があります。

エージェントがキューとの間でメッセージを読み書きする場合、対応する **GET** または **PUT** は、まとめてトランザクションにグループ化されます。トランザクション内の **GET** または **PUT** の数は、チェックポイント

間隔内のすべてのデータを処理するために必要な数によって決定されます。チェックポイント間隔のデータの概算の量は、次の計算を使用して、エージェント・プロパティから決定されます。

```
Checkpoint interval data size (in bytes) = agentCheckpointInterval * agentFrameSize *
agentWindowSize * agentChunkSize.
```

デフォルトのチェックポイント・データ・サイズは、 $1 * 5 * 10 * 262144$ バイト = 13107200 バイト (12.5 MB) です。キュー・マネージャーがサポートする、トランザクションの非コミット・メッセージの最大数は、**MaxUncommittedMsgs** キュー・マネージャー属性によって制御されます。この属性のデフォルト値は 10000 メッセージです。平均メッセージ・サイズが約 1310 バイトより小さい場合、10000 を超すメッセージが書き込まれると、非コミット・メッセージのデフォルトの最大数を超えます。

MaxUncommittedMsgs の制限を超えた場合、エージェントは停止し、次のエラーがエージェント・イベント・ログに記録されます。

```
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.
```

理由コード 2024 は MQRC_SYNCPOINT_LIMIT_REACHED に対応します。

この問題を解決するには、以下のいずれかの処置を行います。

- キューとの間で読み書きしているエージェントが接続するキュー・マネージャーの **MaxUncommittedMsgs** キュー・マネージャー属性の値を増やします。 [MaxUncommittedMsgs \(MQLONG\)](#) を参照してください。
- チェックポイント間隔のデータ量を削減します。削減のためには、以下の 1 つ以上のエージェント・プロパティの値を減らします。
 - agentCheckpointInterval
 - agentFrameSize
 - agentWindowSize
 - agentChunkSize

これらのエージェント・プロパティについて詳しくは、[拡張エージェント・プロパティ](#)を参照してください。

メッセージを永続的にキューに書き込む場合

メッセージをキューに転送し、永続的にキューに書き込む場合は、チェックポイント間隔のすべてのデータをログに記録できるように、キュー・マネージャーのログ・ファイル・スペースのサイズを増やさなければならない場合があります。

キュー・マネージャーのログ・ファイル・スペースを超えた場合、エージェントは停止し、次のエラーがエージェント・イベント・ログに記録されます。

```
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
```

理由コード '2102' は MQRC_RESOURCE_PROBLEM に対応します。

この問題を解決するには、宛先エージェント・キュー・マネージャーのログ・ファイル・スペースのサイズを増やします。

メッセージからファイルへの転送の待機時間を指定する際のガイダンス

メッセージからファイルへの転送を指定するときに、オプションで **-sqwt** パラメーターを使用して、転送の待機時間を指定できます。**-sqwt** の値は、メッセージがソース・キューに表示される (ソース・キューが空であるか、空になる場合) か、またはグループ全体がソース・キューに表示される (**-sqgi** 属性が指定されている場合) のをソース・エージェントが待つ時間です。

このトピックでは、待機時間を指定するために **fteCreateTransfer** コマンドで使用されるパラメーターについて説明します。 **fte:filespec** パラメーターの srcqueuetimeout 値を使用して、待機時間を指定することもできます。

-sqwt パラメーターの値が、ソース・エージェントによる転送が完了するのを宛先エージェントが待つ時間以上である場合、転送は完了しません。転送が完了するのを宛先エージェントが待つ時間は、次の計算で得られます。

```
transferAckTimeout * transferAckTimeoutRetries
```

プロパティ `transferAckTimeout` および `transferAckTimeoutRetries` は宛先エージェントの `agent.properties` ファイルに設定されます。これらのエージェント・プロパティについて詳しくは、550 ページの『MFT agent.properties ファイル』を参照してください。

転送の完了が失敗しないようにするには、次のいずれかの手順を実行する必要があります。

- **-sqwt** パラメーターの値を、宛先エージェントの `transferAckTimeout` プロパティの値よりも小さくなるように指定する。

注: `transferAckTimeout` プロパティのデフォルト値は 60,000 ミリ秒です。 **-sqwt** パラメーターの値は秒で指定するので、この値を 59 以下に設定します。

- 宛先エージェントの `transferAckTimeout` プロパティの値を、 **-sqwt** パラメーターの値よりも大きくなるように指定する。

注: `transferAckTimeout` プロパティの値はミリ秒で指定します。 **-sqwt** パラメーターの値は秒で指定します。

関連資料

437 ページの『[fteCreateTransfer: 新規ファイル転送の開始](#)』

fteCreateTransfer コマンドは、コマンド行から新規のファイル転送を作成して開始します。このコマンドは、ファイル転送の即時開始、将来の日時におけるファイル転送のスケジュール、ファイル転送の 1 回以上の繰り返し、および特定の条件に基づくファイル転送の起動を実行できます。

550 ページの『[MFT agent.properties ファイル](#)』

各 Managed File Transfer Agent には、独自のプロパティ・ファイル `agent.properties` があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。 `agent.properties` ファイルには、エージェントの動作を変更するプロパティを含めることもできます。

862 ページの『[fte:filespec Ant のネストされたエレメント](#)』

fte:filespec パラメーターは、その他のタスクでネストされたエレメントとして使用されます。

fte:filespec を使用すると、1 つ以上のソース・ファイル、ディレクトリー  またはデータ・セット、および宛先間のマッピングを記述できます。一般的には、このエレメントは、移動またはコピーする一連のファイル、ディレクトリー 、またはデータ・セットを表す場合に使用されます。

MFT で使用できるコード・ページ

この参照トピックでは、Managed File Transfer がサポートする各種のプラットフォームでのテキスト・ファイル変換に使用可能な、すべての文字エンコード・フォーマットをリストしています。

共通のエンコード

以下の文字エンコード・フォーマットは、サポートされるすべてのプラットフォームで使用可能です。ソース・ファイルがこの表にあるフォーマットのいずれかを使用してエンコードされており、宛先ファイルに書き込むために、この表にある別のフォーマットを使用することを望む場合には、プラットフォームについて考慮することなく、そのようにすることができます。正規名または、いずれかの別名を使用して、エンコード・フォーマットを指定することができます。

正規の名称	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
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

正規の名称	別名
IBM-863	Cp863, ibm863
IBM-864	Cp864, ibm864
IBM-865	ibm865, Cp865
windows-1257	Cp1257, ibm-1257
windows-1258	Cp1258, ibm-1129, ibm-1258
windows-31j	ms_kanji, cswindows31j, MS932、 windows-932
windows-874	MS874
windows-936	MS936、 x-mswin-936、 936
windows-949	MS949, Cp1361, ibm-1361, ibm1361, ms1361, ksc5601-1992, x-windows-949
windows-950	MS950, x-windows-950
IBM-857	ibm857, Cp857, csibm857
IBM-856	Cp856, ibm856
IBM-855	Cp855, ibm855
IBM-852	cspcp852, ibm852, Cp852
IBM-850	Cp850, ibm850, cspc850multilingual
IBM-838	Cp838, ibm838
IBM-834	Cp834, ibm834
IBM-775	ibm775, Cp775
IBM-737	Cp737, ibm737
IBM-500	Cp500, ibm500
IBM-437	ibm437, Cp437, cspc8codepage437
IBM-424	ibm424, Cp424
IBM-1123	Cp1123, ibm1123
IBM-1124	Cp1124, ibm1124
IBM-1381	Cp1381, ibm1381
IBM-866	Cp866, ibm866
IBM-868	Cp868, ibm868
IBM-869	ibm869, Cp869
IBM-870	Cp870, ibm870
IBM-871	ibm871, Cp871
IBM-874	ibm874, Cp874
IBM-875	Cp875, ibm875
IBM-921	Cp921, ibm921
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

正規の名前	別名
IBM-949	ibm949, Cp949
IBM-950	ibm950, Cp950
ISCII91	iscii
ISO-2022-CN	iso2022-cn-cns, iso2022cn-cns, iso-2022-cn-cns, iso2022cn, iso2022-cn
ISO-2022-CN-GB	iso2022-cn-gb, iso2022cn-gb
ISO-2022-JP	iso2022jp, jis, iso2022-jp, iso-2022-jp2, csiso2022jp2, csjisencoding, jis-encoding
ISO-2022-KR	csiso2022kr, iso2022-kr, iso2022kr
ISO-8859-1	iso8859_1, iso8859-1, ibm819, l1, csisolatin1, Cp819, iso-ir-100, iso-8859-1:1987, ibm-819, latin1, 8859-1
ISO-8859-13	iso8859-13, 8859-13, iso8859_13
ISO-8859-15	csisolatin9, iso8859-15, ibm923, latin9, ibm-923, l9, iso8859_15, iso8859_15_fdis, Cp923, latin0
ISO-8859-2	Cp912, ibm912, iso8859-2, iso-8859-2:1987, l2, iso8859_2, csisolatin2, latin2, ibm-912, 8859-2, iso-ir-101
ISO-8859-3	iso8859-3, Cp913, l3, iso8859_3, iso-ir-109, iso-8859-3:1988, latin3, ibm-913, 8859-3, csisolatin3
ISO-8859-4	Cp914, latin4, iso8859_4, l4, iso-8859-4:1988, ibm-914, iso8859-4, 8859-4, csisolatin4, iso-ir-110
ISO-8859-5	csisolatincyrillic, iso-ir-144, cyrillic, iso8859_5, iso-8859-5:1988, ibm-915, 8859-5, Cp915, ibm915, iso8859-5
ISO-8859-6	csisolatinarabic, Cp1089, iso-8859-6:1987, ecma-114, iso-ir-127, asmo-708, iso8859_6, 8859-6, ibm1089, arabic, iso8859-6, ibm-1089
ISO-8859-7	ecma-118, ibm813, csisolatingreek, elot-928, iso-ir-126, Cp813, 8859-7, iso-8859-7:1987, iso8859_7, greek, greek8, ibm-813, iso8859-7
ISO-8859-8	iso-ir-138, iso-8859-8:1988, csisolatinhebrew, hebrew, iso8859-8, 8859-8, ibm-916, iso8859_8, Cp916, ibm916
ISO-8859-9	ibm-920, ibm920, latin5, 8859-9, Cp920, l5, iso8859-9, iso8859_9, csisolatin5, iso-ir-148
JIS0212	
KOI8-R	koi8, ibm-878, cskoi8r, koi8_r
MacArabic	
MacCentralEurope	ibm-1282
MacCroatian	ibm-1284
MacCyrillic	ibm-1283
MacGreek	ibm-1280
MacIceland	ibm-1286
MacRoman	ibm-1275
MacRomania	ibm-1285
MacSymbol	Adobe-Symbol-Encoding, ibm-1038
MacTurkish	ibm-1281

ソース・プラットフォームのデフォルト・エンコード

ソース・ファイル用のエンコード、または宛先ファイル用のエンコードを指定しない場合には、当該プラットフォームのデフォルト・エンコードが使用されることになります。変換は宛先エージェントによって実行されます。変換が行われるためには、ソース・エンコードと宛先エンコードの両方が、宛先エージェントのプラットフォームでサポートされている必要があります。宛先のデフォルト・エンコードは宛先エ

ージェントで常にサポートされるので、それを指定しないでおけば常に安全です。しかし、デフォルト・ソース・エンコードの使用は常に安全であるとは限りません。それは、宛先エージェントがソースのデフォルトをサポートしない可能性があるからです。

デフォルトのソース・エンコードを使用する場合には、このトピックの各表を用いて、組み合わせがサポートされるかどうかを確認してください。

プラットフォーム	デフォルト・エンコード
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

プラットフォーム固有のエンコード

注: 以下の2つの表の内容は同じ情報です。プラットフォームとエンコード方式のいずれかで検索しても正しい情報を見つけられるように、2つの異なる方法で編成されています。

プラットフォーム別のエンコード

正規名が太字でリストされ、続いて別名が括弧内に示されています。

「共通のエンコード」表に既にリストされたエンコードだけをサポートするプラットフォームは、ここにはリストされていません。

プラットフォーム	サポートされるエンコード (共通エンコード方式の表にないもの)
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>

プラットフォーム	サポートされるエンコード (共通エンコード方式の表にないもの)
SUSE Linux Enterprise Server on System x	<p> windows-1256S (Cp1256s, ibm-1256s) UTF-8J (UTF8J) UTF-32LE (UTF_32LE, X-UTF-32LE, UTF32LE) UTF-32BE (UTF_32BE, X-UTF-32BE, UTF32BE) UTF-32 (UCS-4, UTF32, ISO-10646-UCS-4) PTCP154 (PT154, IBM-1169, Cyrillic-Asian, csPTCP154) KOI8-RU (ibm-1168, koi8_ru) ISO-8859-16 (8859-16, iso8859_16, iso8859-16) ISO-8859-14 (ISO_8859-14:1998, 8859-14, latin8, iso-ir-199, iso8859-14, l8, isoceltic, iso8859_14) IBM01141 (cp1141, ccsid01141, cp01141, 1141) IBM01142 (cp01142, cp1142, 1142, ccsid01142) IBM01143 (cp01143, 1143, ccsid01143, cp1143) IBM01144 (cp01144, cp1144, ccsid01144, 1144) IBM01145 (cp1145, cp01145, ccsid01145, 1145) IBM01146 (ccsid01146, cp01146, cp1146, 1146) IBM01147 (ccsid01147, cp1147, 1147, cp01147) IBM01148 (cp1148, ccsid01148, 1148, cp01148) IBM01149 (cp1149, cp01149, ccsid01149, 1149) IBM1047 (cp1047, 1047, ibm-1047) IBM918 (cp918, ebcdic-cp-ar2, ibm-918, 918) ISO-2022-JP-2 (csISO2022JP2, iso2022jp2) x-Big5-Solaris (Big5_Solaris) x-eucJP-Open (EUC_JP_Solaris, eucJP-open) x-IBM33722 (ibm33722, 33722, ibm-33722_vascii_vpua, ibm-5050, ibm-33722, cp33722) x-IBM930 (cp930, ibm930, ibm-930, 930) x-IBM939 (ibm-939, ibm939, cp939, 939) x-IBM964 (964, cp964, ibm-964, ibm964) x-ISO-2022-CN-CNS (ISO-2022-CN-CNS, ISO2022CN_CNS) x-iso-8859-11 (iso-8859-11, iso8859_11) x-JISAutoDetect (JISAutoDetect) x-MS932_0213 () x-MS950-HKSCS (MS950_HKSCS) x-PCK (pck) x-IBM1363C (ibm1363c, cp1363c, ibm-1363c) x-IBM420S (420s, ibm-420s, csibm420s, ibm420s, cp420s) x-IBM864S (csibm864s, ibm864s, cp864s, 864s, ibm-864s) x-IBM943C (cp943c, 943c, ibm-943c, ibm943c) x-IBM949C (ibm949c, cp949c, 949c, ibm-949c) x-IBM954C (cp954c, 954c, ibm-954c, ibm954c) x-ISO-8859-6S (8859_6s, iso8859-6s, iso8859_6s, iso-8859-6s) x-JIS0208 (jis_c6226-1983, jis_x0208-1983, csiso87jisx0208, x0208, iso-ir-87, jis0208) x-KSC5601 (ksc5601) x-MacDingbat (macdingbat) x-MacHebrew (machebrew) x-MacThai (macthai) x-MacUkraine (macukraine) x-IBM1046S (ibm-1046s, 1046s, cp1046s, ibm1046s) x-IBM-udcJP (IBM-udcJP) JIS_X0201 (jis_x0201, x0201, cshalfwidthkatakana, jis0201) IBM-939A (Cp939A, ibm939A) IBM-930A (ibm930A, Cp930A) IBM-33722A (Cp33722A, ibm33722A) x-windows-iso2022jp (windows-iso2022jp) x-windows-50221 (ms50221, cp50221) x-windows-50220 (cp50220, ms50220) X-UTF-32LE-BOM (UTF_32LE_BOM, UTF-32LE-BOM) X-UTF-32BE-BOM (UTF_32BE_BOM, UTF-32BE-BOM) x-SJIS_0213 () IBM01140 (ccsid01140, cp01140, 1140, cp1140) IBM00858 (cp858, ccsid00858, 858, cp00858) X-UnicodeLittle (UnicodeLittle) X-UnicodeBig (UnicodeBig) IBM-859 (Cp859, ibm859) </p>

プラットフォーム	サポートされるエンコード (共通エンコード方式の表にないもの)
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>

プラットフォーム	サポートされるエンコード (共通エンコード方式の表にないもの)
<p>IBM i IBM i IBM i</p>	<p> windows-1256S (Cp1256s, ibm-1256s) UTF-8J (UTF8J) IBM-1146 (Cp1146, ibm1146) IBM-1145 (Cp1145, ibm1145) IBM-1144 (ibm1144, Cp1144) IBM-1143 (Cp1143, ibm1143) IBM-1142 (Cp1142, ibm1142) IBM-1141 (Cp1141, ibm1141) IBM-1140 (ibm1140, Cp1140) IBM-1115 (Cp1115, ibm1115) IBM-1114 (Cp1114, ibm1114) hp-roman8 (roman8, ibm-1051, r8, Cp1051) GBK (GBK) GB2312 (gb2312-1980, gb2312-80) COMPOUND_TEXT (x-compound-text, x11-compound-text) CESU-8 (CESU8) IBM-1027 (Cp1027, ibm1027) IBM-1041 (Cp1041, ibm1041) IBM-1043 (Cp1043, ibm1043) IBM-1046S (ibm1046S, Cp1046S) IBM-1047 (Cp1047, ibm1047) IBM-1088 (Cp1088, ibm1088) IBM-1382 (ibm1382, Cp1382) IBM-1385 (Cp1385, ibm1385) IBM-1386 (ibm1386, Cp1386) IBM-1388 (Cp1388, ibm1388) IBM-836 (ibm836, Cp836) IBM-837 (ibm837, Cp837) IBM-858 (Cp858, ibm858) IBM-859 (Cp859, ibm859) IBM-864S (ibm864S, Cp864S) X-UnicodeBig (UnicodeBig) X-UnicodeLittle (UnicodeLittle) IBM-1047_LF (Cp1047_LF, ibm1047_LF) IBM-1141_LF (Cp1141_LF, ibm1141_LF) IBM-33722A (Cp33722A, ibm33722A) IBM-924_LF (Cp924_LF, ibm924_LF) IBM-930A (ibm930A, Cp930A) IBM-939A (Cp939A, ibm939A) IBM-835 (ibm835, Cp835) IBM-833 (ibm833, Cp833) IBM-808 (Cp808, ibm808) IBM-720 (Cp720, ibm720) IBM-420S (Cp420S, ibm420S) IBM-33722C (ibm-eucjp, Cp33722c) IBM-33722 (5050, Cp5050) IBM-301 (Cp301, ibm301) IBM-300 (Cp300, ibm300) IBM-290 (ibm290, Cp290) IBM-1399 (ibm1399, Cp1399) IBM-1390 (Cp1390, ibm1390) IBM-1147 (Cp1147, ibm1147) IBM-1148 (ibm1148, Cp1148) IBM-1149 (Cp1149, ibm1149) IBM-1351 (Cp1351, ibm1351) IBM-1362 (Cp1362, ibm1362) IBM-1363 (ibm1363, Cp1363) IBM-1363C (ibm1363C, Cp1363C) IBM-1364 (Cp1364, ibm1364) IBM-1370 (Cp1370, ibm1370) IBM-1371 (Cp1371, ibm1371) IBM-1380 (Cp1380, ibm1380) IBM-867 (Cp867, ibm867) IBM-897 (Cp897, ibm897) </p>

プラットフォーム	サポートされるエンコード (共通エンコード方式の表にないもの)
  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>

プラットフォーム	サポートされるエンコード (共通エンコード方式の表にないもの)
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>

プラットフォーム	サポートされるエンコード (共通エンコード方式の表にないもの)
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>

プラットフォーム	サポートされるエンコード (共通エンコード方式の表にないもの)
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>

プラットフォーム	サポートされるエンコード (共通エンコード方式の表にないもの)
AIX	<p> windows-1256S (Cp1256s, ibm-1256s) UTF-8J (UTF8J) UTF-32LE (UTF_32LE, X-UTF-32LE, UTF32LE) UTF-32BE (UTF_32BE, X-UTF-32BE, UTF32BE) UTF-32 (UCS-4, UTF32, ISO-10646-UCS-4) UTF-16 (UTF16, Unicode, UTF_16, UCS-2) Shift_JIS () PTCP154 (PT154, IBM-1169, Cyrillic-Asian, csPTCP154) MacUkraine () MacThai () MacHebrew () MacDingbat () KSC5601 () KOI8-U (koi8_u, ibm-1167) KOI8-RU (ibm-1168, koi8_ru) Johab (x-johab) JISO208 () JISO201 () ISO-8859-6S (iso8859-6S, iso8859_6S) ISO-8859-16 (8859-16, iso8859_16, iso8859-16) ISO-8859-14 (ISO_8859-14:1998, 8859-14, latin8, iso-ir-199, iso8859-14, l8, isoceltic, iso8859_14) ISO-8859-10 (latin6, 8859-10, ISO_8859-10:1992, iso8859_10, iso-ir-157, ibm-919, iso8859-10, l6, csisolatin6) IBM-971 (Cp971, ibm971) IBM-964 (ibm-euctw, Cp964) IBM-954C (Cp954c) IBM-954 (ibm954, Cp954) IBM-951 (Cp951, ibm951) IBM-949C (Cp949C, ibm949C) IBM-947 (Cp947, ibm947) IBM-943C (ibm943C, Cp943C) IBM-942C (Cp942C, ibm942C) IBM-939 (Cp5035, 5035) IBM-932 (ibm932, Cp932) IBM-930 (Cp5026, 5026) IBM-927 (ibm927, Cp927) IBM-924 (Cp924, ibm924) IBM-918 (ibm918, Cp918) IBM-897 (Cp897, ibm897) IBM-867 (Cp867, ibm867) IBM-1380 (Cp1380, ibm1380) IBM-1371 (Cp1371, ibm1371) IBM-1370 (Cp1370, ibm1370) IBM-1364 (Cp1364, ibm1364) IBM-1363C (ibm1363C, Cp1363C) IBM-1047 (Cp1047, ibm1047) IBM-1088 (Cp1088, ibm1088) IBM-1382 (ibm1382, Cp1382) IBM-1385 (Cp1385, ibm1385) IBM-1386 (ibm1386, Cp1386) IBM-1388 (Cp1388, ibm1388) IBM-1390 (Cp1390, ibm1390) IBM-1399 (ibm1399, Cp1399) IBM-290 (ibm290, Cp290) IBM-300 (Cp300, ibm300) IBM-301 (Cp301, ibm301) IBM-33722 (5050, Cp5050) X-UnicodeLittle (UnicodeLittle) X-UnicodeBig (UnicodeBig) IBM-864S (ibm864S, Cp864S) IBM-859 (Cp859, ibm859) IBM-858 (Cp858, ibm858) </p>

プラットフォーム	サポートされるエンコード (共通エンコード方式の表にないもの)
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>

プラットフォーム	サポートされるエンコード (共通エンコード方式の表にないもの)
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>

プラットフォーム	サポートされるエンコード (共通エンコード方式の表にないもの)
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>

プラットフォーム	サポートされるエンコード (共通エンコード方式の表にないもの)
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>

プラットフォーム	サポートされるエンコード (共通エンコード方式の表にないもの)
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>

プラットフォーム	サポートされるエンコード (共通エンコード方式の表にないもの)
<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>

プラットフォーム	サポートされるエンコード (共通エンコード方式の表にないもの)
  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>

プラットフォーム	サポートされるエンコード (共通エンコード方式の表にないもの)
Linux on POWER Systems - Big Endian	<p> windows-1256S (Cp1256s, ibm-1256s) UTF-8J (UTF8J) UTF-32LE (UTF_32LE, X-UTF-32LE, UTF32LE) UTF-32BE (UTF_32BE, X-UTF-32BE, UTF32BE) UTF-32 (UCS-4, UTF32, ISO-10646-UCS-4) UTF-16 (UTF16, Unicode, UTF_16, UCS-2) Shift_JIS () PTCP154 (PT154, IBM-1169, Cyrillic-Asian, csPTCP154) MacUkraine () MacThai () MacHebrew () MacDingbat () KSC5601 () KOI8-U (koi8_u, ibm-1167) KOI8-RU (ibm-1168, koi8_ru) Johab (x-johab) JISO208 () JISO201 () ISO-8859-6S (iso8859-6S, iso8859_6S) ISO-8859-16 (8859-16, iso8859_16, iso8859-16) ISO-8859-14 (ISO_8859-14:1998, 8859-14, latin8, iso-ir-199, iso8859-14, l8, isoceltic, iso8859_14) ISO-8859-10 (latin6, 8859-10, ISO_8859-10:1992, iso8859_10, iso-ir-157, ibm-919, iso8859-10, l6, csisolatin6) IBM-971 (Cp971, ibm971) IBM-964 (ibm-euctw, Cp964) IBM-954C (Cp954c) IBM-954 (ibm954, Cp954) IBM-951 (Cp951, ibm951) IBM-949C (Cp949C, ibm949C) IBM-947 (Cp947, ibm947) IBM-943C (ibm943C, Cp943C) IBM-942C (Cp942C, ibm942C) IBM-939 (Cp5035, 5035) IBM-932 (ibm932, Cp932) IBM-930 (Cp5026, 5026) IBM-927 (ibm927, Cp927) IBM-924 (Cp924, ibm924) IBM-918 (ibm918, Cp918) IBM-897 (Cp897, ibm897) IBM-867 (Cp867, ibm867) IBM-1380 (Cp1380, ibm1380) IBM-1371 (Cp1371, ibm1371) IBM-1370 (Cp1370, ibm1370) IBM-1364 (Cp1364, ibm1364) IBM-1363C (ibm1363C, Cp1363C) IBM-1047 (Cp1047, ibm1047) IBM-1088 (Cp1088, ibm1088) IBM-1382 (ibm1382, Cp1382) IBM-1385 (Cp1385, ibm1385) IBM-1386 (ibm1386, Cp1386) IBM-1388 (Cp1388, ibm1388) IBM-1390 (Cp1390, ibm1390) IBM-1399 (ibm1399, Cp1399) IBM-290 (ibm290, Cp290) IBM-300 (Cp300, ibm300) IBM-301 (Cp301, ibm301) IBM-33722 (5050, Cp5050) X-UnicodeLittle (UnicodeLittle) X-UnicodeBig (UnicodeBig) IBM-864S (ibm864S, Cp864S) IBM-859 (Cp859, ibm859) IBM-858 (Cp858, ibm858) </p>

プラットフォーム	サポートされるエンコード (共通エンコード方式の表にないもの)
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>

プラットフォーム	サポートされるエンコード (共通エンコード方式の表にないもの)
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>

エンコード別のプラットフォーム

Encoding	別名	このエンコードがサポートされるプラットフォーム
x-MacUkraine	macukraine	Solaris、 SUSE Linux Enterprise Server on System x、 Linux for IBM Z Windows、 Red Hat Enterprise Linux on System x

Encoding	別名	このエンコードがサポートされるプラットフォーム
x-MacThai	macthai	Solaris, SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
x-MacHebrew	machebrew	Solaris, SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
x-MacDingbat	macdingbat	Solaris, SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
x-KSC5601	ksc5601	Solaris, SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
x-JIS0208	jis_c6226-1983, jis_x0208-1983, csiso87jisx0208, x0208, iso-ir-87, jis0208	Solaris, SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
x-ISO-8859-6S	8859_6s, iso8859-6s, iso8859_6s, iso-8859-6s	SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
x-IBM954C	cp954c, 954c, ibm-954c, ibm954c	SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
x-IBM949C	ibm949c, cp949c, 949c, ibm-949c	Solaris, SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x

Encoding	別名	このエンコードがサポートされるプラットフォーム
x-IBM943C	cp943c, 943c, ibm-943c, ibm943c	Solaris, SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
x-IBM864S	csibm864s, ibm864s, cp864s, 864s, ibm-864s	SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
x-IBM420S	420s, ibm-420s, csibm420s, ibm420s, cp420s	SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
x-IBM1363C	ibm1363c, cp1363c, ibm-1363c	SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
x-IBM1046S	ibm-1046s, 1046s, cp1046s, ibm1046s	SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
x-IBM-udcJP	IBM-udcJP	SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
JIS_X0201	jis_x0201、x0201、cshalfwidthkatakana、 jis0201	Solaris, SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
IBM-939A	Cp939A, ibm939A	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
IBM-930A	ibm930A, Cp930A	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x,  z/OS

Encoding	別名	このエンコードがサポートされるプラットフォーム
▶ IBM i ▶ IBM i IBM-924_LF	Cp924_LF, ibm924_LF	IBM i
IBM-33722A	Cp33722A, ibm33722A	SUSE Linux Enterprise Server on System x, ▶ IBM i IBM i, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x
▶ IBM i ▶ IBM i IBM-1141_LF	Cp1141_LF, ibm1141_LF	IBM i
▶ IBM i ▶ 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)

Encoding	別名	このエンコードがサポートされるプラットフォーム
X-UTF-32BE-BOM	UTF_32BE_BOM, UTF-32BE-BOM	Solaris, SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
x-SJIS_0213		Solaris, SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
x-PCK	pck	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
x-MS950-HKSCS	MS950_HKSCS	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
x-MS932_0213		Solaris, SUSE Linux Enterprise Server on System x, Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
x-JISAutoDetect	JISAutoDetect	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)

Encoding	別名	このエンコードがサポートされるプラットフォーム
x-iso-8859-11	iso-8859-11, iso8859_11	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
x-ISO-2022-CN-CNS	ISO-2022-CN-CNS, ISO2022CN_CNS	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
x-IBM964	964, cp964, ibm-964, ibm964	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
x-IBM939	ibm-939, ibm939, cp939, 939	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
x-IBM930	cp930, ibm930, ibm-930, 930	Solaris, SUSE Linux Enterprise Server on System x, HP-UX (Itanium), Linux for IBM Z Windows, Red Hat Enterprise Linux on System x, HP (PA-RISC)
x-IBM33722	ibm33722, 33722, ibm-33722_vascij_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)

Encoding	別名	このエンコードがサポートされるプラットフォーム
x-eucJP-Open	EUC_JP_Solaris、 eucJP-open	Solaris、 SUSE Linux Enterprise Server on System x、 HP-UX (Itanium)、 Linux for IBM Z Windows、 Red Hat Enterprise Linux on System x、 HP (PA-RISC)
x-Big5-Solaris	Big5_Solaris	Solaris、 SUSE Linux Enterprise Server on System x、 HP-UX (Itanium)、 Linux for IBM Z Windows、 Red Hat Enterprise Linux on System x、 HP (PA-RISC)
Roman9	Roman9	HP (PA-RISC)
ISO-2022-JP-2	csISO2022JP2、 iso2022jp2	Solaris、 SUSE Linux Enterprise Server on System x、 Linux for IBM Z Windows、 Red Hat Enterprise Linux on System x、 HP (PA-RISC)
IBM918	cp918、 ebcdic-cp-ar2、 ibm-918、 918	Solaris、 SUSE Linux Enterprise Server on System x、 HP-UX (Itanium)、 Linux for IBM Z Windows、 Red Hat Enterprise Linux on System x、 HP (PA-RISC)
IBM1047	cp1047、 1047、 ibm-1047	Solaris、 SUSE Linux Enterprise Server on System x、 HP-UX (Itanium)、 Linux for IBM Z Windows、 Red Hat Enterprise Linux on System x、 HP (PA-RISC)
IBM01149	cp1149、 cp01149、 ccsid01149、 1149	Solaris、 SUSE Linux Enterprise Server on System x、 HP-UX (Itanium)、 Linux for IBM Z Windows、 Red Hat Enterprise Linux on System x、 HP (PA-RISC)

Encoding	別名	このエンコードがサポートされるプラットフォーム
IBM01148	cp1148、ccsid01148、1148、cp01148	Solaris、 SUSE Linux Enterprise Server on System x、 HP-UX (Itanium)、 Linux for IBM Z Windows、 Red Hat Enterprise Linux on System x、 HP (PA-RISC)
IBM01147	ccsid01147、cp1147、1147、cp01147	Solaris、 SUSE Linux Enterprise Server on System x、 HP-UX (Itanium)、 Linux for IBM Z Windows、 Red Hat Enterprise Linux on System x、 HP (PA-RISC)
IBM01146	ccsid01146、cp01146、cp1146、1146	Solaris、 SUSE Linux Enterprise Server on System x、 HP-UX (Itanium)、 Linux for IBM Z Windows、 Red Hat Enterprise Linux on System x、 HP (PA-RISC)
IBM01145	cp1145、cp01145、ccsid01145、1145	Solaris、 SUSE Linux Enterprise Server on System x、 HP-UX (Itanium)、 Linux for IBM Z Windows、 Red Hat Enterprise Linux on System x、 HP (PA-RISC)
IBM01144	cp01144、cp1144、ccsid01144、1144	Solaris、 SUSE Linux Enterprise Server on System x、 HP-UX (Itanium)、 Linux for IBM Z Windows、 Red Hat Enterprise Linux on System x、 HP (PA-RISC)
IBM01143	cp01143、1143、ccsid01143、cp1143	Solaris、 SUSE Linux Enterprise Server on System x、 HP-UX (Itanium)、 Linux for IBM Z Windows、 Red Hat Enterprise Linux on System x、 HP (PA-RISC)

Encoding	別名	このエンコードがサポートされるプラットフォーム
IBM01142	cp01142、 cp1142、 1142、 ccsid01142	Solaris、 SUSE Linux Enterprise Server on System x、 HP-UX (Itanium)、 Linux for IBM Z Windows、 Red Hat Enterprise Linux on System x、 HP (PA-RISC)
IBM01141	cp1141、 ccsid01141、 cp01141、 1141	Solaris、 SUSE Linux Enterprise Server on System x、 HP-UX (Itanium)、 Linux for IBM Z Windows、 Red Hat Enterprise Linux on System x、 HP (PA-RISC)
IBM01140	ccsid01140、 cp01140、 1140、 cp1140	Solaris、 SUSE Linux Enterprise Server on System x、 HP-UX (Itanium)、 Linux for IBM Z Windows、 Red Hat Enterprise Linux on System x、 HP (PA-RISC)
IBM00858	cp858、 ccsid00858、 858、 cp00858	Solaris、 SUSE Linux Enterprise Server on System x、 HP-UX (Itanium)、 Linux for IBM Z Windows、 Red Hat Enterprise Linux on System x、 HP (PA-RISC)
X-UnicodeLittle	UnicodeLittle	Solaris、 SUSE Linux Enterprise Server on System x、  IBM i、 Linux for IBM Z  AIX、 Windows、 Red Hat Enterprise Linux on System x、  z/OS、 Linux on POWER Systems - Big Endian、 HP (PA-RISC)

Encoding	別名	このエンコードがサポートされるプラットフォーム
X-UnicodeBig	UnicodeBig	Solaris, SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
IBM-864S	ibm864S, Cp864S	 IBM i,  AIX,  z/OS, Linux on POWER Systems - Big Endian
IBM-859	Cp859, ibm859	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-858	Cp858, ibm858	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian
IBM-837	ibm837, Cp837	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian

Encoding	別名	このエンコードがサポートされるプラットフォーム
IBM-836	ibm836, Cp836	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-835	ibm835, Cp835	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-833	ibm833, Cp833	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-808	Cp808, ibm808	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-720	Cp720, ibm720	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian

Encoding	別名	このエンコードがサポートされるプラットフォーム
IBM-420S	Cp420S, ibm420S	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian
IBM-33722C	ibm-eucjp, Cp33722c	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-33722	5050, Cp5050	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian
IBM-301	Cp301, ibm301	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-300	Cp300, ibm300	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian

Encoding	別名	このエンコードがサポートされるプラットフォーム
IBM-290	ibm290, Cp290	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-1399	ibm1399, Cp1399	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-1390	Cp1390, ibm1390	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-1388	Cp1388, ibm1388	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-1386	ibm1386, Cp1386	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian

Encoding	別名	このエンコードがサポートされるプラットフォーム
IBM-1385	Cp1385, ibm1385	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-1382	ibm1382, Cp1382	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-1088	Cp1088, ibm1088	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-1047	Cp1047, ibm1047	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian
IBM-1046S	ibm1046S, Cp1046S	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian

Encoding	別名	このエンコードがサポートされるプラットフォーム
IBM-1043	Cp1043, ibm1043	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-1041	Cp1041, ibm1041	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-1027	Cp1027, ibm1027	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
CESU-8	CESU8	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian

Encoding	別名	このエンコードがサポートされるプラットフォーム
COMPOUND_TEXT	x-compound-text, x11-compound-text	Solaris, SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
GB2312	gb2312-1980, gb2312-80	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
GBK	GBK	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
hp-roman8	roman8, ibm-1051, r8, Cp1051	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)

Encoding	別名	このエンコードがサポートされるプラットフォーム
IBM-1114	Cp1114, ibm1114	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-1115	Cp1115, ibm1115	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-1140	ibm1140, Cp1140	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian
IBM-1141	Cp1141, ibm1141	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian
IBM-1142	Cp1142, ibm1142	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian
IBM-1143	Cp1143, ibm1143	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian
IBM-1144	ibm1144, Cp1144	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian

Encoding	別名	このエンコードがサポートされるプラットフォーム
IBM-1145	Cp1145, ibm1145	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian
IBM-1146	Cp1146, ibm1146	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian
IBM-1147	Cp1147, ibm1147	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian
IBM-1148	ibm1148, Cp1148	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian
IBM-1149	Cp1149, ibm1149	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian
IBM-1351	Cp1351, ibm1351	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-1362	Cp1362, ibm1362	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian

Encoding	別名	このエンコードがサポートされるプラットフォーム
IBM-1363	ibm1363, Cp1363	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-1363C	ibm1363C, Cp1363C	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian
IBM-1364	Cp1364, ibm1364	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
IBM-1370	Cp1370, ibm1370	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-1371	Cp1371, ibm1371	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian

Encoding	別名	このエンコードがサポートされるプラットフォーム
IBM-1380	Cp1380, ibm1380	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-867	Cp867, ibm867	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-897	Cp897, ibm897	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-918	ibm918, Cp918	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian
IBM-924	Cp924, ibm924	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian

Encoding	別名	このエンコードがサポートされるプラットフォーム
IBM-927	ibm927, Cp927	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-930	Cp5026, 5026	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian
IBM-932	ibm932, Cp932	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-939	Cp5035, 5035	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian
IBM-942C	Cp942C, ibm942C	Solaris,  IBM i, HP-UX (Itanium), AIX,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
IBM-943C	ibm943C, Cp943C	 IBM i, HP-UX (Itanium), AIX,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)

Encoding	別名	このエンコードがサポートされるプラットフォーム
IBM-947	Cp947, ibm947	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-949C	Cp949C, ibm949C	 IBM i, HP-UX (Itanium), AIX,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
IBM-951	Cp951, ibm951	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-954	ibm954, Cp954	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
IBM-954C	Cp954c	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian
IBM-964	ibm-euctw, Cp964	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian

Encoding	別名	このエンコードがサポートされるプラットフォーム
IBM-971	Cp971, ibm971	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
ISO-8859-10	latin6, 8859-10, ISO_8859-10:1992、 iso8859_10、iso-ir-157、ibm-919、 iso8859-10、l6、csisolatin6	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
ISO-8859-14	ISO_8859-14:1998、8859-14、latin8、iso- ir-199、iso8859-14、l8、isoceltic、 iso8859_14	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
ISO-8859-16	8859-16、iso8859_16、iso8859-16	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
ISO-8859-6S	iso8859-6S, iso8859_6S	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian

Encoding	別名	このエンコードがサポートされるプラットフォーム
JIS0201		 IBM i, HP-UX (Itanium), AIX,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
JIS0208		 IBM i, HP-UX (Itanium), AIX,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
Johab	x-johab	 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian
KOI8-RU	ibm-1168, koi8_ru	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
KOI8-U	koi8_u, ibm-1167	Solaris,  IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
KSC5601		 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian

Encoding	別名	このエンコードがサポートされるプラットフォーム
MacDingbat		 IBM i, HP-UX (Itanium), AIX,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
MacHebrew		 IBM i, HP-UX (Itanium), AIX,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
MacThai		 IBM i, HP-UX (Itanium), AIX,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
MacUkraine		 IBM i, HP-UX (Itanium), AIX,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
PTCP154	PT154、IBM-1169、Cyrillic-Asian、csPTCP154	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
Shift_JIS		 IBM i, AIX,  z/OS, Linux on POWER Systems - Big Endian

Encoding	別名	このエンコードがサポートされるプラットフォーム
UTF-16	UTF16、Unicode、UTF_16、UCS-2	 IBM i, HP-UX (Itanium), AIX,  z/OS, Linux on POWER Systems - Big Endian
UTF-32	UCS-4, UTF32, ISO-10646-UCS-4	Solaris, SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
UTF-32BE	UTF_32BE, X-UTF-32BE, UTF32BE	Solaris, SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)
UTF-32LE	UTF_32LE, X-UTF-32LE, UTF32LE	Solaris, SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian, HP (PA-RISC)

Encoding	別名	このエンコードがサポートされるプラットフォーム
UTF-8J	UTF8J	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian
windows-1256S	Cp1256s, ibm-1256s	SUSE Linux Enterprise Server on System x,  IBM i, Linux for IBM Z AIX, Windows, Red Hat Enterprise Linux on System x,  z/OS, Linux on POWER Systems - Big Endian

関連概念

161 ページの『転送定義ファイルの使用』

ファイル転送を作成するために使用できる転送定義ファイルを指定できます。転送定義ファイルは、転送を作成するために必要な情報の一部またはすべてを定義した XML ファイルです。

関連資料

643 ページの『MFT でのテキスト・ファイルの転送』

テキスト・ファイルの転送には、あるコード・ページから別のコード・ページへの、ファイルのコード・ページの変換が伴います。テキスト・ファイルの転送には、システム間の CRLF (復帰改行) 文字の変換も伴います。このトピックでは Managed File Transfer のテキスト・ファイル転送動作をまとめています。

437 ページの『fteCreateTransfer: 新規ファイル転送の開始』

fteCreateTransfer コマンドは、コマンド行から新規のファイル転送を作成して開始します。このコマンドは、ファイル転送の即時開始、将来の日時におけるファイル転送のスケジュール、ファイル転送の 1 回以上の繰り返し、および特定の条件に基づくファイル転送の起動を実行できます。

MFT エージェントが Java ヒープ・メモリーとネイティブ・ヒープ・メモリーを使用する方法

IBM MQ Managed File Transfer エージェントは、Java プロセスとして稼働します。そのため、このエージェントは、Java 仮想マシン (JVM) の仮想化環境で稼働します。

JVM 自体はネイティブ・プロセスであり、ハードウェアとオペレーティング・システムの制約を受けます。JVM は 2 つのメモリー領域を保持します。

- Java ヒープ

このヒープには Java オブジェクトのインスタンスが入ります。ガーベッジ・コレクション処理による管理の対象になります。Java ヒープの最大サイズが、JVM の始動時に **-Xmx** JVM オプションで割り振られます。

- ネイティブ・ヒープ

ネイティブ・ヒープには JVM 自体のリソースが入ります。例えば、JIT (Just-In-Time) コンパイラー、クラス、クラス・ローダーなどです。

エージェントは主に Java ヒープを使用します。エージェントは管理対象転送を実行する時に、Java ヒープを使用して、転送に必要な Java オブジェクトを作成します。エージェントによってバッファーに読み込まれるファイル・データも、Java ヒープ・メモリーに保管されます。

エージェント自体には、ネイティブ・ヒープを使用するコードが含まれていません。ただし、エージェントがエージェント・キュー・マネージャーとの通信で使用するネイティブ・コードが、Java メッセージ・キュー・インターフェース (JMQUI) 内にあります。

このネイティブ・コードは、エージェントが BINDINGS トランスポートを使用してエージェント・キュー・マネージャーに接続する時に使用されます。これは、エージェントが CLIENT トランスポートで接続する場合に使用する TCP/IP 接続ではなく、ローカル共有メモリー接続です (プロセス間通信 (IPC) ともいいます)。エージェントが BINDINGS トランスポートを使用するように構成されていると、エージェントとエージェント・キュー・マネージャーとの間のメッセージやコマンドの受け渡しにネイティブ・ヒープが使用されます。

つまり、BINDINGS トランスポートでエージェント・キュー・マネージャーに接続する高負荷のエージェントは、CLIENT トランスポートで接続する同等のエージェントよりも、ネイティブ・ヒープをかなり多く使用します。

エージェントの Java ヒープは転送対象の最大ファイルのサイズ以上でなければならないというのは、よくある誤解です。この考えは正しくありません。ファイル・データは段階的にメモリーに読み込まれていくからです。

1 回の転送でファイル・データの保管に使用される Java ヒープの最大量は、大まかに言って、以下のよう
に計算できます。

```
Memory allocated for a transfer = agentCheckpointInterval *  
agentFrameSize * agentWindowSize * agentChunkSize
```

Java ヒープとネイティブ・ヒープの使用がエージェントに与える影響

java.lang.OutOfMemoryError が発生すると、**-Xmx** Java システム・プロパティーを使用して、アプリケーションで使用できる Java ヒープの量を増やす方がよい場合もあります。例えば、最大 Java ヒープ・サイズとして 2GB を割り振る場合は、以下のプロパティー設定を使用します。

```
-Xmx2048M
```

ただし、アプリケーションの Java ヒープの割り振りが多すぎると、逆にネイティブ・ヒープが不足して、java.lang.OutOfMemoryError が発生する可能性もあります。Java ヒープ・スペースが大きくなると、それに応じてネイティブ・ヒープが縮小するからです。

ネイティブ・ヒープの不足が原因で発生する java.lang.OutOfMemoryErrors の防止方法については、[ネイティブ・メモリーの消耗が原因で、MFT エージェントが java.lang.OutOfMemoryError で異常終了した場合の対策を参照してください。](#)

MFT で使用する XML メッセージ形式

Managed File Transfer では、エージェントへの指示、モニターに関する情報の記録、スケジューリング、転送、および構成に使用される情報の定義など、さまざまな目的のために XML 形式のメッセージを使用します。これらの目的のために使用される XML 形式の論理構造は XML スキーマにより記述されます。

Managed File Transfer の各バージョンは、XML で作成されたメッセージを妥当性検査するために XML スキーマを使用します。エージェントは XML スキーマのバージョンを取り出し、そのスキーマがサポートされているかどうかを判別します。

Managed File Transfer のインストール後、Managed File Transfer メッセージ・スキーマ・ファイルは `MQ_INSTALLATION_PATH/mqft/samples/schema` ディレクトリーにあります。以下のスキーマが含まれています。

エージェント・コマンド・キューに配置できる XML メッセージのスキーマ

- FileTransfer.xsd
- Internal.xsd

Monitor.xsd

PingAgent.xsd

エージェント・コマンド・キューへの XML メッセージの配置についての詳細は [270 ページの『エージェント・コマンド・キューにメッセージを PUT することによる MFT の制御』](#) を参照してください。

SYSTEM.FTE トピックにパブリッシュされる XML メッセージのスキーマ

MonitorList.xsd

MonitorLog.xsd

ScheduleList.xsd

ScheduleLog.xsd

TransferLog.xsd

TransferStatus.xsd

SYSTEM.FTE トピックにパブリッシュされる XML メッセージおよび SYSTEM.FTE トピックの構造についての詳細は、[619 ページの『SYSTEM.FTE トピック』](#) を参照してください。

Managed File Transfer によって使用されるその他のスキーマ

fteutils.xsd。このスキーマには共通の元素定義が含まれており、他のいくつかのスキーマによって組み込まれます。

Notification.xsd

ProtocolBridgeCredentials.xsd

ProtocolBridgeProperties.xsd

ConnectDirectCredentials.xsd

ConnectDirectNodeProperties.xsd

ConnectDirectProcessDefinitions.xsd

Reply.xsd

UserSandboxes.xsd

MFT エージェント状況メッセージ・フォーマット

Managed File Transfer Agent を作成したり開始したりすると、エージェントは自身の詳細情報を調整キュー・マネージャーの SYSTEM.FTE トピック (SYSTEM.FTE/Agents/*agent name* のトピック) にパブリッシュします。

以下の情報が含まれています。

- エージェント名
- エージェントが動作しているプラットフォーム
- エージェントの説明 (提供されている場合)
- エージェントのキュー・マネージャー
- エージェントが動作しているタイム・ゾーン
- エージェントのバージョン
- エージェント転送制限
- エージェントのそれぞれの現行転送状態。各状態については、[『エージェント転送状態』](#) を参照
- エージェントのタイプ

エージェントがプロトコル・ブリッジ・エージェントの場合は、以下の情報も組み込まれます。

- プロトコル・ブリッジ・エージェントのタイプ
- プロトコル・ブリッジ・サーバーのホスト名または IP アドレス

エージェントの転送状態が変わるときには、エージェント状況はリパブリッシュされますが、デフォルトでは最大で 30 秒に 1 回です。このデフォルト設定は `agentStatusPublishRateLimit` エージェント・プロパティを使用して変更できます。このことについては、[『拡張エージェント・プロパティ』](#) で説明されています。

以下の出力例は、エージェント状況の各データ・エレメントで使用されているキーを示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <entry key="agentOsName">Windows 7</entry>
  <entry key="agentDescription"/>
  <entry key="queueManager">QM1</entry>
  <entry key="agentTimeZone">Europe/London</entry>
  <entry key="agentVersion">1.00</entry>
  <entry key="agentName">FTEAGENT</entry>
  <entry key="maxDestinationTransfers">25</entry>
  <entry key="maxSourceTransfers">25</entry>
  <entry key="maxQueuedTransfers">100</entry>
  <entry
key="DestinationTransferStates">414d51204d554e474f202020202020d857374a60a72622=RunningTransfer
414d51204d554e474f202020202020d857374a69a72622=RunningTransfer
414d51204d554e474f202020202020d857374a75a72622=RunningTransfer
  </entry>
  <entry
key="SourceTransferStates">414d51204d554e474f202020202020d857374a93a72622=NegotiatingTransfer
414d51204d554e474f202020202020d857374a78a72622=RunningTransfer
414d51204d554e474f202020202020d857374aaba72622=NewSenderTransfer
414d51204d554e474f202020202020d857374a63a72622=RunningTransfer
  </entry>
</properties>
```

以下の出力例は、プロトコル・ブリッジ・エージェントのエージェント状況の各データ・エレメントで使用されているキーを示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <entry key="agentOsName">Windows 7</entry>
  <entry key="agentDescription"/>
  <entry key="queueManager">QM1</entry>
  <entry key="agentTimeZone">Europe/London</entry>
  <entry key="agentVersion">1.00</entry>
  <entry key="agentName">BRIDGE</entry>
  <entry key="protocolBridgeType">ftp</entry>
  <entry key="protocolBridgeServerHost">ftpserver.example.org</entry>
  <entry key="maxDestinationTransfers">25</entry>
  <entry key="maxSourceTransfers">25</entry>
  <entry key="maxQueuedTransfers">100</entry>
  <entry key="DestinationTransferStates">414d51204d554e474f202020202020d857374a60a72622=RunningTransfer
  </entry>
  <entry key="SourceTransferStates">414d51204d554e474f202020202020d857374a93a72622=NegotiatingTransfer
  </entry>
</properties>
```

MFT エージェント 転送状態

開始された Managed File Transfer Agent は、その詳細を調整キュー・マネージャーの SYSTEM.FTE トピックにパブリッシュします。詳細には、そのエージェントに関係する現在の各転送の状態が含まれます。

転送状態	説明
NewSenderTransfer	ネゴシエーションが開始されていないソース・エージェントからの新規の転送。
NewReceiverTransfer	ネゴシエーションの一部として宛先エージェントで新規の転送が作成されましたが、転送はまだ実行されていません。
NegotiatingTransfer	ソース・エージェントが宛先エージェントと転送実行前のネゴシエーション中です。
RunningTransfer	通常の実行状態のソース・エージェントまたは宛先エージェントからの転送。

転送状態	説明
RecoveringTransfer	ソース・エージェントまたは宛先エージェントがリカバリー・プロセスを開始すると、実行状態の転送は転送状態に移行します。再同期メッセージがピア・エージェントに送信されると、転送はこの状態から ReSynchronisingTransfer 状態に移行します。 例えば、宛先エージェントが実行中の転送のリカバリー・プロセスを開始した場合、再同期メッセージがソース・エージェントに送信されると、その転送は ReSynchronisingTransfer 状態に移行します。
ReSynchronisingTransfer	転送のソース・エージェントまたは宛先エージェントが問題を検出し、再同期メッセージをそれぞれ対応する宛先エージェントまたはソース・エージェントに送信しました。
CompletedTransfer	宛先エージェントが転送を完了し、完了メッセージをソース・エージェントに送信しました。宛先エージェントは、ソース・エージェントからの確認応答メッセージを待っています。
CompleteReceivedTransfer	ソース・エージェントが宛先エージェントから完了メッセージを受信し、完了メッセージを確認するメッセージを宛先エージェントに送信しました。
CancelledNewTransfer	ソース・エージェントが新規の転送の取り消しメッセージを受信しました。
CancelledInProgressTransfer	ソース・エージェントが進行中の転送の取り消しメッセージを受信しました。
ResumingTransfer	ソース・エージェントが再同期応答メッセージを受信し、転送の再開を現在スケジュールしています。
RestartingTransfer	ソース・エージェントまたは宛先エージェントが再同期要求メッセージを受信し、対応する宛先エージェントまたはソース・エージェントの再始動を待機しています。
WaitingForDestinationCapacity	ソース・エージェントが宛先エージェントから DESTINATION_CAPACITY_EXCEEDED エラーを受信しました。転送は、一定期間後の再試行のために現在待ち状態にあります。
FailedTransferEnding	転送が失敗しましたが、完了ログ・メッセージは未公開で、転送は状態ストアから削除されていません。例えば、宛先エージェントからの障害応答を受け取ってから、後続の処理が完了するまでの間にエージェント・プロセスが停止した場合、この状態が発生します。

MFT モニター・リスト・メッセージ・フォーマット

トピック・ストリング SYSTEM.FTE/monitors/agent_name/monitor_name への保存パブリケーションとしてパブリッシュされる XML メッセージは、MonitorList.xsd スキーマに準拠します。各 XML メッセージは、そのエージェントに属するアクティブ・モニターをリストします。この情報は **ftelistmonitors** コマンドおよび IBM MQ Explorer・プラグインで、ユーザーに対してモニターのリストを表示するために使用されます。MonitorList.xsd スキーマ文書は MQ_INSTALLATION_PATH/mqft/samples/schema ディレクトリーにあります。MonitorList.xsd スキーマは、同じディレクトリーにある Monitor.xsd をインポートします。

スキーマ

次のスキーマは、モニター・リスト XML メッセージで有効なエレメントについて示します。

```
<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>

```

モニター・リスト・メッセージについて

モニター・リスト・メッセージで使用するエレメントと属性を以下のリストにまとめます。

<monitorList>

エージェントに対して定義されたモニターを記述するエレメントを含むグループ・エレメント。

属性	説明
エージェント	必須。 リソース・モニターが定義されているエージェントの名前。
monitor	必須。 モニターの名前。 このエージェントに固有です。
バージョン	必須。 モニター・リスト・メッセージ・フォーマットのバージョン。

<status>

モニターの状況。

属性	説明
状態	モニターの状態。

<configuration>

モニターの構成を記述するエレメントを含むグループ・エレメント。

<description>

モニターの説明。(現在は使用されていません。)

<resources>

モニターされるリソース。

<directory>

モニター対象のディレクトリー。

属性	説明
recursionLevel	モニター対象の最上位レベルから数えたディレクトリー・レベルの数。
ID	リソースの ID。

<queue>

モニター対象のキュー。

属性	説明
ID	リソースの ID。

<triggerMatch>

<conditions> エレメントを含むエレメント。

<conditions>

リソース・モニターのモニター対象である条件が含まれているエレメント。このエレメントは、<allOf>、<anyOf>、または <condition> のいずれか 1 つのエレメントのみを含むことができます。

<allOf>

リソース・モニターのモニター対象である条件が含まれているエレメント。このエレメントは 1 つまたは多数の <condition> エレメントを含むことができます。リソース・モニターが起動されるためには、このエレメント内のすべての条件が満たされる必要があります。

<anyOf>

リソース・モニターのモニター対象である条件が含まれているエレメント。このエレメントは 1 つまたは多数の <condition> エレメントを含むことができます。リソース・モニターが起動されるためには、このエレメント内の条件が 1 つだけ満たされる必要があります。

<condition>

リソース・モニターのモニター対象である条件が含まれているエレメント。このエレメントは、<fileMatch>、<fileNoMatch>、<fileSize>、<queueNotEmpty>、<completeGroups>、または <fileSizeSame> のいずれか 1 つのエレメントのみを含むことができます。また、<name> エレメントと <resource> エレメントを含むことができます。

モニター対象のリソースがディレクトリーである場合は、以下の 3 つのエレメントのいずれかを条件で指定する必要があります。

- fileMatch
- fileNoMatch
- fileSize

モニター対象のリソースがキューである場合は、以下の 2 つのエレメントのいずれかを条件で指定する必要があります。

- queueNotEmpty
- completeGroups

<fileMatch>

ファイル名マッチング条件のグループ・エレメント。

<pattern>

ファイル名マッチング・パターンを指定します。条件を満たすには、リソース上のファイルがこのパターンと一致している必要があります。デフォルト・パターンは、* (任意のファイルが一致) です。

<fileNoMatch>

ファイル名逆マッチング条件のグループ・エレメント

<pattern>

ファイル名逆マッチング・パターンを指定します。モニター対象リソースに一致するファイルがない場合に、この条件が満たされます。デフォルト・パターンは、*(ファイルがまったくない場合に一致)です。

<fileSize>

ファイル・サイズ比較のグループ・エレメント。

<compare>

ファイル・サイズ比較を指定します。値は負以外の整数でなければなりません。

属性	説明
operator	使用する比較演算子。「>=」のみがサポートされています。
単位	ファイル・サイズ単位を指定します。以下のいずれかです。 <ul style="list-style-type: none"> • B - バイト • KB - キロバイト • MB - メガバイト • GB - ギガバイト 単位の値では、大/小文字は区別されません。したがって、「mb」は「MB」としても機能します。

<pattern>

一致するファイル名パターン。デフォルトは、*(任意のファイルが一致)です。

<queueNotEmpty>

リソースがキューの場合にのみ指定できます。モニターを起動するためにはキューにメッセージが入っていないなければならない、という動作を指定します。

<completeGroups>

リソースがキューの場合にのみ指定できます。モニターを起動するためには完全に揃ったメッセージ・グループがキューに存在していなければならない、という動作を指定します。キューに存在する完全に揃ったグループごとに1つの転送タスクが実行されます。

<名前>

条件の名前。

<resource>

条件を比較する際の基準となるリソース定義を指定します。

属性	説明
ID	リソースの固有 ID。

<tasks>

モニター・トリガー条件が満たされた場合に呼び出すタスクを指定するエレメントが含まれているグループ・エレメント。

<task>

トリガー条件が満たされた場合にモニターが呼び出す個々のタスクを定義するグループ・エレメント。現時点で、指定できるのは1つのタスクのみです。

<名前>

タスクの名前。任意の英数字を受け入れます。

<description>

タスクの説明。任意のテキスト値を使用できます。

<taskXML>

モニターが実行するタスクを記述する XML メッセージ。このエレメントの内容はエスケープ XML フォーマットで記述されます。

<pollInterval>

トリガー条件に照らし合わせるリソースの各検査間の時間間隔。

属性	説明
単位	ポーリング間隔の時間単位を指定します。有効な値は次のとおりです。 <ul style="list-style-type: none">• seconds• minutes• hours• 日• weeks• months• years

<batch>

1回のバッチに含まれる、一致するトリガーの最大数。

属性	説明
maxSize	1回のバッチに含まれる、一致するトリガーの最大数

次のXMLは、MONITORTWO と呼ばれるモニターが AGENT_JUPITER に作成されるときに、トピック・ストリング SYSTEM.FTE/monitors/agent_name/MONITORTWO にパブリッシュされる保存パブリケーションの例を示しています。 <taskXML> エレメント内のエスケープXMLは、モニター条件が満たされたときにサブミットされるタスクを記述します。

```
<?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 スケジュール・リスト・メッセージ・フォーマット

保存パブリケーションからトピック・ストリング SYSTEM.FTE/Scheduler/agent_name にパブリッシュされる XML メッセージは、ScheduleList.xsd スキーマに準拠します。この XML メッセージは、そのエージェントに属するすべてのアクティブ・スケジュールをリストします。この情報は、**fteListScheduledTransfers** コマンドおよび IBM MQ Explorer で、ユーザーに対してスケジュールのリストを表示するために使用されます。ScheduleList.xsd スキーマ文書は MQ_INSTALLATION_PATH/mqft/samples/schema ディレクトリーにあります。ScheduleList.xsd スキーマは、同じディレクトリーにある FileTransfer.xsd をインポートします。

スキーマ

次のスキーマは、モニター・リスト XML メッセージで有効なエレメントについて示します。

```
<?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>
```

スケジュール・リスト・メッセージについて

スケジュール・リスト・メッセージで使用するエレメントと属性を以下のリストにまとめます。

<schedules>

1つのエージェント上で定義されるすべてのスケジュールに関する情報を含むグループ・エレメント。

属性	説明
エージェント	必須。スケジュールが定義されているソース・エージェントの名前。
size	必須。このエージェントに定義されているスケジュールの数。
バージョン	必須。スケジュール・メッセージ・フォーマットのバージョン。

<managedTransfer>

単一のスケジュールに関する情報を含むグループ・エレメント。

属性	説明
ID	必須。スケジュール要求メッセージの 16 進数ストリングの ID。

<originator>

スケジュール要求の送信元。

<hostName>

スケジュール要求の送信元マシンのホスト名。

<userID>

スケジュール要求を送信したユーザーのユーザー ID。

<mqmdUserID>

スケジュール要求を送信したユーザーの MQMD ユーザー ID。

<schedule>

スケジュール済み転送がいつ発生するかを記述するエレメントを含むエレメント。

<submit>

スケジュール済み転送の開始日時を示します。

属性	説明
timebase	使用するタイム・ゾーンを示します。この属性の値は以下のいずれかの値になります。 <ul style="list-style-type: none"> • source - ソース・エージェントのタイム・ゾーンを使用します。 • admin - コマンドを発行する管理者のタイム・ゾーンを使用します。 • UTC - 協定世界時を使用します。
timezone	時間基準値に従ったタイム・ゾーンの記述

<repeat>

スケジュール済み転送の繰り返し頻度、スケジュール済み転送の繰り返し回数、およびスケジュール済み転送の繰り返しの停止日時の詳細を含むグループ・エレメント。

属性	説明
interval	以下のいずれかで示される間隔の単位。 <ul style="list-style-type: none"> • minutes • hours • 日 • weeks • months • years

<frequency>

転送を繰り返す時間間隔。

属性	説明
interval	以下のいずれかで示される間隔の単位。 <ul style="list-style-type: none"> • minutes • hours • 日 • weeks • months • years

<expireTime>

スケジュール済み転送の繰り返しを停止する日時を指定するオプションのエレメント。このエレメントと <expireCount> エレメントは相互に排他的です。

<expireCount>

スケジュール済みファイル転送を終了するまでの実行回数を指定するオプションのエレメント。
このエレメントと <expireTime> エレメントは相互に排他的です。

<next>

次のスケジュール済み転送を開始する日時を指定します。

<sourceAgent>

ソース・ファイルがあるシステム上のエージェントの名前を示します。

属性	説明
エージェント	エージェントの名前を示します。
キュー・マネージャー	エージェントのキュー・マネージャーの名前。

<destinationAgent>

ファイルの転送先にするシステム上のエージェントの名前を示します。

属性	説明
エージェント	エージェントの名前を示します。
キュー・マネージャー	エージェントのキュー・マネージャーの名前。

<trigger>

ファイル転送の実行のために満たす必要がある条件を指定するオプションのエレメント。

属性	説明
log	トリガー障害がログに記録されるかどうかを示すフラグ。有効な値は次のとおりです。 <ul style="list-style-type: none"> • yes - 失敗したトリガー転送についてのログ項目が作成されます。 • no - 失敗したトリガー転送についてのログ項目が作成されません。

<reply>

同期ファイル転送用に生成された一時応答キューの名前を示します (コマンド行で **-w** パラメーターにより指定します)。キューの名前は、`command.properties` 構成ファイル内のキー **dynamicQueuePrefix**、またはデフォルトの `WMQFTE.*` によって定義されます。指定されていない場合、

属性	説明
QMGR	応答を受け取るために一時動的キューが生成されるコマンド・キュー・マネージャーの名前。

<transferSet>

まとめてスケジュール済み転送を実行するファイル転送のグループを示します。送信中は、<transferSet> は <item> エレメントが含まれるグループ・エレメントです。

属性	説明
priority	転送の優先順位。priority は 0 から 9 までの範囲の値で、0 が最低の優先順位です。デフォルトの優先順位は 0 で、デフォルトではソース・エージェントの優先順位が転送で使用されます。

<job>

転送仕様全体のジョブ情報を含むオプションのグループ・エレメント。<job> は、転送が開始されたときに、ログ・メッセージに追加されるユーザー定義ジョブ名 ID です。この<job> エレメントは、転送要求メッセージに含まれる <job> エレメントと同じです。748 ページの『[ファイル転送ログ・メッセージ・フォーマット](#)』のトピックを参照してください。

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 テンプレート XML メッセージの例

テンプレートが作成される時に、メッセージは SYSTEM.FTE トピックに `Templates/template_ID` というトピック・ストリングでパブリッシュされます。この XML 例では、Managed File Transfer のネットワークで定義された 1 つのテンプレートを説明しています。

```
<?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>

```

関連タスク

197 ページの『IBM MQ Explorer を使用したファイル転送テンプレートの作成』

ファイル転送テンプレートを IBM MQ Explorer またはコマンド行から作成することができます。その後そのテンプレートを使用してそのテンプレート詳細を使用する新規ファイル転送を作成したり、そのテンプレートを送信してファイル転送を開始したりできます。

関連資料

421 ページの『fteCreateTemplate: 新規ファイル転送テンプレートの作成』

fteCreateTemplate コマンドは、将来の使用のために保持できるファイル転送テンプレートを作成します。必須パラメーターは **-tn** (*template_name*) パラメーターのみです。その他のパラメーターはすべてオプションです。ただし、ソース・ファイル仕様を指定する場合には、宛先ファイルも提供する必要があります。同様に、宛先ファイルを指定する場合には、ソース・ファイル仕様も指定する必要があります。

ファイル転送状況メッセージ・フォーマット

メッセージは、転送セット内の各ファイルの転送状況を示すために、調整キュー・マネージャーにパブリッシュされます。エージェントがファイル転送の要求を処理するごとに、TransferStatus.xsd XML スキーマに準拠する調整キュー・マネージャー (SYSTEM.FTE/Transfers/agent_name/transfer ID トピック上) にトランザクション・メッセージがパブリッシュされます。TransferStatus.xsd ファイルは、WMQMFT インストール済み環境の MQ_INSTALLATION_PATH/mqft/samples/schema ディレクトリーにあります。

スキーマ

次のスキーマは、転送状況 XML メッセージで有効なエレメントについて示します。

```

<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"
          minOccurs="1"/>
        <xsd:element name="destinationAgent" type="agentType"
          minOccurs="1"/>
        <xsd:element name="transferSet" type="transferSetType"
          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"
        minOccurs="1" maxOccurs="1"/>
      <xsd:element name="current" type="currentType"
        minOccurs="0" maxOccurs="1"/>
    </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"
        minOccurs="1" maxOccurs="1"/>
      <xsd:element name="destination" type="fileDestinationType"
        minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="transferred" type="xsd:nonNegativeInteger"
      use="required"/>
  </xsd:complexType>

```

```

        <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>

```

転送状況メッセージについて

転送状況メッセージで使用するエレメントと属性を以下のリストにまとめます。

<transaction>

ファイル転送のすべてのエレメントが含まれるグループ・エレメント。

属性	説明
バージョン	Managed File Transfer によって指定された、このエレメントのバージョンを示します。
ID	ファイル転送のための固有 ID。

<sourceAgent>

ソース・ファイルがあるシステム上のエージェントの名前を示します。

属性	説明
エージェント	エージェントの名前。
キュー・マネージャー	エージェントのキュー・マネージャーの名前。

<destinationAgent>

ファイルの転送先にするシステム上のエージェントの名前を示します。

属性	説明
エージェント	エージェントの名前。
キュー・マネージャー	エージェントのキュー・マネージャーの名前。

<transferSet>

一緒に行われているファイル転送のグループを示します。転送に組み込むファイルはすべて、同じソース・エージェントを起点とし、同じ宛先エージェントを終点としなければなりません。

属性	説明
time	日付と時間を示します (日時形式)。

<stats>

必須。一定秒数の転送に関するメトリック (その時点までにコピーされたバイト数を含む) を定義します。また、<transferSet> 内の合計項目数中の現在の項目数も提供します。

属性	説明
バイト	これまでコピーされたバイト数。
seconds	それらのバイト数の転送にかかった秒数。
currentItem	現在転送中の項目の索引。

属性	説明
totalItems	転送中の項目の合計数。

<current>

オプションの要素。現在進行中のファイル転送を示す要素が含まれるグループ・要素。<current>要素は、現在の項目に関してその時点までに転送されたデータのバイト数と、予想される合計バイト数を示します。

<source>

ソース・ファイル名を示す要素が含まれるグループ・要素。

<file>

転送中のファイルのソース・パスを指定します。このパスは、転送で指定されているとおりです。このパスは、転送ログの一部として出力される絶対パス形式のパスとは異なる場合があります。

<destination>

宛先ファイル名または仕様を示す要素が含まれるグループ・要素。

<file>

転送中のファイルの宛先パスを示します。このパスは、転送で指定されているとおりです。このパスは、転送ログの一部として出力される絶対パス形式のパスとは異なる場合があります。

属性	説明
別名	宛先ファイルの別名を指定します。この別名は、転送で指定されているディレクトリー・パスを除外したソース・ファイルの名前です。
filesystem	宛先ファイルの書き込み先のファイル・スペースの名前を指定します。

<queue>

<destination>要素と一緒に使用する場合には、転送先にするキューの名前を指定します。この名前のフォーマットは QUEUE または QUEUE@QUEUE_MANAGER です。

ファイル転送進行メッセージの例

転送の進行中、メッセージは SYSTEM.FTE トピックに Transfers/agent_name/transfer_ID というトピック・ストリングでパブリッシュされます。以下の XML の例は、単一ファイル転送および複数ファイル転送の進行メッセージを示しています。

単一ファイル転送

以下の例は、進行中の単一ファイル転送の詳細を示します。

```
<?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>
```

複数ファイル転送

転送セット内にさらにファイルがあった場合、処理中のファイルとそれまでに転送されたバイト数を示す転送状況メッセージが表示されます。

```
<?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>
```

ファイル転送ログ・メッセージ・フォーマット

ファイル転送ログ・メッセージは、SYSTEM.FTE トピックに、Log/agent_name/transfer_ID というトピック・ストリングでパブリッシュされます。これらのメッセージは、Managed File Transfer インストール済み環境の MQ_INSTALLATION_PATH/mqft/samples/schema ディレクトリーにある TransferLog.xsd スキーマに準拠しています。

ファイル転送をモニターしたり、それに関するデータを収集したりする場合は、対象となる転送に合わせて調整されたワイルドカード・トピックのサブスクリプションをセットアップします。以下に例を示します。

```
Log/#
```

または、

```
Log/FTEAGENT/#
```

このサブスクリプションは、永続または非永続のどちらにもできます。永続サブスクリプションは、キュー・マネージャーへのサブスクライブ・アプリケーションの接続が閉じられても存続します。非永続サブスクリプションは、サブスクライブ・アプリケーションのキュー・マネージャーへの接続が開いている間だけ存在します。

スキーマ

次のスキーマは、転送ログ XML メッセージ内で有効なエレメントについて示します。

```
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
  <xsd:include schemaLocation="fteutils.xsd"/>
  <xsd:element name="transaction">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="action" type="actionType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="sourceAgent" type="agentExitStatusType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="sourceWebGateway" type="webGatewayType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="sourceWebUser" type="webUserType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="destinationAgent" type="agentExitStatusType"/>
```

```

        maxOccurs="1"                minOccurs="0" />
<xsd:element name="destinationWebGateway" type="webGatewayType"
maxOccurs="1"                minOccurs="0" />
<xsd:element name="destinationWebUser" type="webUserType"
maxOccurs="1"                minOccurs="0" />
<xsd:element name="agent" type="agentExitStatusType"
maxOccurs="1"                minOccurs="0" />
<xsd:element name="originator" type="origRequestType"
maxOccurs="1"                minOccurs="1" />
<xsd:element name="status" type="statusType"
maxOccurs="1"                minOccurs="0" />
<xsd:element name="trigger" type="triggerType"
maxOccurs="1"                minOccurs="0" />
<xsd:element name="transferSet" type="transferSetType"
maxOccurs="1"                minOccurs="1" />
<xsd:element name="job" type="jobType"
maxOccurs="1"                minOccurs="0" />
<xsd:element name="scheduleLog" type="scheduleLogType"
maxOccurs="1"                minOccurs="0" />
<xsd:element name="statistics" type="statisticsType"
maxOccurs="1"                minOccurs="0" />
</xsd:sequence>
<xsd:attribute name="version" type="versionType" use="required" />
<xsd:attribute name="ID" type="IDType" use="required" />
<xsd:attribute name="relatedID" type="IDType" use="optional" />
<xsd:attribute name="agentRole" type="agentRoleType" use="optional" />
</xsd:complexType>
</xsd:element>

<xsd:complexType name="agentExitStatusType">
  <xsd:complexContent>
    <xsd:extension base="agentType">
      <xsd:sequence>
        <xsd:element name="startExits" type="exitGroupType" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="endExits" type="exitGroupType" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="systemInfo" type="systemInfoType" minOccurs="0"
maxOccurs="1" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="transferSetType">
  <xsd:sequence>
    <xsd:element name="metaDataSet" type="metaDataSetType"
maxOccurs="1"                minOccurs="0" />
    <xsd:element name="call" type="callGroupType"
maxOccurs="1"                minOccurs="0" />
    <xsd:element name="preSourceCall" type="callGroupType"
maxOccurs="1"                minOccurs="0" />
    <xsd:element name="postSourceCall" type="callGroupType"
maxOccurs="1"                minOccurs="0" />
    <xsd:element name="preDestinationCall" type="callGroupType"
maxOccurs="1"                minOccurs="0" />
    <xsd:element name="postDestinationCall" type="callGroupType"
maxOccurs="1"                minOccurs="0" />
    <xsd:element name="item" type="itemType"
maxOccurs="unbounded"        minOccurs="0" />
  </xsd:sequence>
  <xsd:attribute name="index" type="xsd:nonNegativeInteger" use="optional" />
  <xsd:attribute name="size" type="xsd:nonNegativeInteger" use="optional" />
  <xsd:attribute name="startTime" type="xsd:dateTime" use="required" />
  <xsd:attribute name="total" type="xsd:nonNegativeInteger" use="required" />
  <xsd:attribute name="bytesSent" type="xsd:nonNegativeInteger" use="required" />
</xsd:complexType>

<xsd:complexType name="itemType">
  <xsd:sequence>
    <xsd:element name="source" type="fileSourceChecksumType"
maxOccurs="1"                minOccurs="1" />
    <xsd:element name="destination" type="fileDestinationChecksumType"
maxOccurs="1"                minOccurs="1" />
    <xsd:element name="status" type="statusType"
maxOccurs="1"                minOccurs="1" />
  </xsd:sequence>
  <xsd:attribute name="mode" type="modeType" use="required" />
</xsd:complexType>

<xsd:complexType name="fileSourceChecksumType">
  <xsd:complexContent>

```

```

        <xsd:extension base="fileSourceType">
          <xsd:sequence>
            <xsd:element name="checksum" type="checksumType" minOccurs="0"
maxOccurs="1"/>
          </xsd:sequence>
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="fileDestinationChecksumType">
      <xsd:complexContent>
        <xsd:extension base="fileDestinationType">
          <xsd:sequence>
            <xsd:element name="checksum" type="checksumType"
minOccurs="0" maxOccurs="1"/>
          </xsd:sequence>
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="actionType">
      <xsd:simpleContent>
        <xsd:extension base="actionEnumType">
          <xsd:attribute name="time" type="xsd:dateTime" use="required"/>
        </xsd:extension>
      </xsd:simpleContent>
    </xsd:complexType>

    <xsd:simpleType name="actionEnumType">
      <xsd:restriction base="xsd:token">
        <xsd:enumeration value="cancelled"/>
        <xsd:enumeration value="started"/>
        <xsd:enumeration value="progress"/>
        <xsd:enumeration value="completed"/>
        <xsd:enumeration value="malformed"/>
        <xsd:enumeration value="notAuthorized"/>
        <xsd:enumeration value="deleted"/>
      </xsd:restriction>
    </xsd:simpleType>

    <xsd:complexType name="systemInfoType">
      <xsd:attribute name="architecture" type="xsd:string" use="required"/>
      <xsd:attribute name="name" type="xsd:string" use="required"/>
      <xsd:attribute name="version" type="xsd:string" use="required"/>
    </xsd:complexType>

    <xsd:element name="malformed">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="action" type="actionType"
maxOccurs="1" minOccurs="1"/>
          <xsd:element name="agent" type="agentExitStatusType"
maxOccurs="1" minOccurs="0"/>
          <xsd:element name="status" type="statusType"
maxOccurs="1" minOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="version" type="versionType" use="required"/>
        <xsd:attribute name="ID" type="IDType" use="required"/>
        <xsd:attribute name="agentRole" type="agentRoleType" use="required"/>
      </xsd:complexType>
    </xsd:element>

    <xsd:element name="notAuthorized">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="action" type="actionType"
maxOccurs="1" minOccurs="1"/>
          <xsd:element name="originator" type="origRequestType"
maxOccurs="1" minOccurs="1"/>
          <xsd:element name="authority" type="xsd:string"
minOccurs="1" maxOccurs="1"/>
          <xsd:element name="status" type="statusType"
maxOccurs="1" minOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="version" type="versionType" use="required"/>
        <xsd:attribute name="ID" type="IDType" use="required"/>
        <xsd:attribute name="agentRole" type="agentRoleType" use="required"/>
      </xsd:complexType>
    </xsd:element>

    <xsd:complexType name="statisticsType">
      <xsd:sequence>

```

```

<xsd:element name="actualStartTime" type="xsd:dateTime"
  maxOccurs="1" minOccurs="0"/>
<xsd:element name="retryCount" type="xsd:nonNegativeInteger"
  maxOccurs="1" minOccurs="1"/>
<xsd:element name="numFileFailures" type="xsd:nonNegativeInteger"
  maxOccurs="1" minOccurs="1"/>
<xsd:element name="numFileWarnings" type="xsd:nonNegativeInteger"
  maxOccurs="1" minOccurs="1"/>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="webGatewayType">
  <xsd:attribute name="webGatewayName" type="xsd:string" use="optional"/>
  <xsd:attribute name="webGatewayAgentName" type="xsd:string" use="optional"/>
  <xsd:attribute name="webGatewayAgentQMgr" type="xsd:string" use="optional"/>
</xsd:complexType>

<xsd:complexType name="webUserType">
  <xsd:attribute name="webGatewayName" type="xsd:string" use="required"/>
  <xsd:attribute name="webGatewayAgentName" type="xsd:string" use="optional"/>
  <xsd:attribute name="webGatewayAgentQMgr" type="xsd:string" use="optional"/>
</xsd:complexType>

</xsd:schema>

```

注：IBM MQ 9.0 から、Managed File Transfer は Web ゲートウェイや Web エージェントをサポートしなくなりました。

転送ログ・メッセージについて

<transaction>

まとめて実行する転送のグループを示すグループ・エレメント。

属性	説明
バージョン	Managed File Transfer で示されるこのエレメントのバージョンを示します。
ID	固有トランザクション ID を指定します。ID には、最大で 48 文字の英数字を指定できます。
relatedID	オプション。トランザクションがファイル・スペースからのファイルの削除またはダウンロードである場合に、 relatedID はファイル・スペースにファイルをアップロードした転送のトランザクション ID を指定します。
agentRole	オプション。関係するエージェントがソース・システムにあるか宛先システムにあるかを示します。
xmlns:xsi	ネーム・スペースの宣言。このスキーマで使用されるエレメントおよびデータ・タイプが「 https://www.w3.org/2001/XMLSchema-instance 」ネーム・スペースから派生することを示します。
xsi:noNamespaceSchemaLocation	ネーム・スペース宣言が存在しない場合にこのメッセージの妥当性検査を行うための XML スキーマ文書の名前と位置を指定します。この属性に指定する値は、Managed File Transfer TransferLog.xsd 文書を参照する必要があります。

<action>

time 属性によりログに記録された時刻におけるファイル転送の状況を示します。状況は以下のいずれかの値になります。

- 開始済み
- 進行
- 完了

- 取り消し済み
- malformed (ファイル転送の要求メッセージの内容を解釈できなかったことを示す。)
- notAuthorized
- deleted

属性	説明
time	転送状況が収集された時刻を、UTC 形式で表記します。

<sourceAgent>

ソース・ファイルがあるシステム上のエージェントの名前を示します。 <sourceAgent> または <sourceWebUser> だけを指定できます。

<startExits>

1つ以上のユーザー出口エレメントを含むグループ・エレメント。このエレメントは、一度だけ使用できます。

<endExits>

1つ以上のユーザー出口エレメントを含むグループ・エレメント。このエレメントは、一度だけ使用できます。

<systemInfo>

システム体系、名前、およびバージョンを説明します。このエレメントは、一度だけ使用できません。

属性	説明
エージェント	ソース・システム上のエージェントの名前。
キュー・マネージャー	ソース・システム上のキュー・マネージャーの名前。
agentType	エージェントのタイプ。有効な値は次のとおりです。 <ul style="list-style-type: none"> • STANDARD - 通常のエージェント • BRIDGE - プロトコル・ブリッジ・エージェント • CD_BRIDGE - Connect:Direct ブリッジ・エージェント • EMBEDDED - 組み込みエージェント • SFG - Sterling File Gateway 組み込みエージェント
bridgeURL	オプション。エージェントがプロトコル・ブリッジ・エージェントの場合、プロトコル・サーバーをホストするシステムのホスト名。
pnode	オプション。Connect:Direct ブリッジ・エージェントの場合は、転送にかかわっている Connect:Direct 1 次ノードの名前。
snode	オプション。Connect:Direct ブリッジ・エージェントの場合は、転送にかかわっている Connect:Direct 2 次ノードの名前。
bridgeNode	オプション。Connect:Direct ブリッジ・エージェントの場合は、Connect:Direct ブリッジの一部になっている Connect:Direct ノードの名前。1 次ノードまたは 2 次ノードのいずれかと同じノードになります。

<destinationAgent>

ファイルが転送されたシステム上のエージェントの名前を示します。 <destinationAgent> または <destinationWebUser> だけを指定できます。

属性	説明
エージェント	宛先システム上のエージェントの名前。

属性	説明
キュー・マネージャー	宛先システム上のキュー・マネージャーの名前。
agentType	エージェントのタイプ。有効な値は次のとおりです。 <ul style="list-style-type: none"> • STANDARD - 通常のエージェント • BRIDGE - プロトコル・ブリッジ・エージェント • CD_BRIDGE - Connect:Direct ブリッジ・エージェント • EMBEDDED - 組み込みエージェント • SFG - Sterling File Gateway 組み込みエージェント
bridgeURL	オプション。エージェントがプロトコル・ブリッジ・エージェントの場合、プロトコル・サーバーをホストするシステムのホスト名。
pnode	オプション。Connect:Direct ブリッジ・エージェントの場合は、転送にかかわっている Connect:Direct 1 次ノードの名前。
snode	オプション。Connect:Direct ブリッジ・エージェントの場合は、転送にかかわっている Connect:Direct 2 次ノードの名前。
bridgeNode	オプション。Connect:Direct ブリッジ・エージェントの場合は、Connect:Direct ブリッジの一部になっている Connect:Direct ノードの名前。1 次ノードまたは 2 次ノードのいずれかと同じノードになります。

<startExits>

1 つ以上のユーザー出口エレメントを含むグループ・エレメント。このエレメントは、一度だけ使用できます。

<endExits>

1 つ以上のユーザー出口エレメントを含むグループ・エレメント。このエレメントは、一度だけ使用できます。

<systemInfo>

システム体系、名前、およびバージョンを説明します。このエレメントは、一度だけ使用できます。

<originator>

要求の発信元を示すエレメントが含まれるグループ・エレメント。

<hostName>

ソース・ファイルが置かれているシステムのホスト名。

<userID>

ファイル転送を開始したユーザーの ID。

<mqmdUserID>

メッセージ記述子 (MQMD) に指定された IBM MQ ユーザー ID

<webUserID>

オプション。転送要求を実行依頼する、Web ブラウザーに指定されたユーザー ID。

<webBrowser>

オプション。転送要求の実行依頼元の Web ブラウザー。

<status>

結果コードと補足メッセージ。

<trigger>

元の転送要求で定義されているトリガー・エレメントを含むグループ・エレメント。このエレメントは、次のうちのいずれかあるいはその両方になります。

<fileExist>

ファイルが存在するかどうかに基づくトリガー条件

<fileSize>

指定されたサイズをファイルが満たしているかまたは超えているかに基づくトリガー条件

<transferSet>

まとめて実行するファイル転送のグループを示します。送信中は、<transferSet> は <item> エレメントが含まれるグループ・エレメントです。

属性	説明
startTime	一連の転送が開始された時刻を、UTC 形式の表記で記録します。
total	この一連の転送の合計項目数を示します。
索引 (index)	オプションの属性。転送セットの進行における最初の項目の位置を示します。index 属性は、ゼロからインクリメントします。例えば、索引が 1 に設定されている場合、この進行メッセージは 2 つある項目のうちの 2 番目です。
size	オプションの属性。進行レポート内の項目の数を示します。
priority	オプションの属性。転送の優先順位。priority は 0 から 9 までの範囲の値で、0 が最低の優先順位です。デフォルトの優先順位は 0 で、デフォルトではソース・エージェントの優先順位が転送で使用されます。

<metaDataSet>

以下の属性の 1 つ以上を含むグループ・エレメント:

<metaData>

属性	説明
key	メタデータのキーと値の対の片方のキー。<metaData> エレメント・コンテンツに、対の片方の値が含まれます。例えば、<metaData key="testkey1">testvalue1</metaData> のようになります。

<job>

ジョブの詳細を指定するエレメントを含むグループ・エレメント。<job> は、転送が開始されたときに、ログ・メッセージに追加されるユーザー定義ジョブ名 ID です。この<job> エレメントは、転送要求メッセージに含まれる <job> エレメントと同じです。[780 ページの『ファイル転送要求メッセージ・フォーマット』](#) のトピックを参照してください。

<名前>

名前の値は、任意のストリングです。

<scheduleLog>

ソースおよび宛先のファイル名と位置を示すエレメントが含まれるグループ・エレメント。

属性	説明
ID	転送がスケジュールした転送である場合、この ID はスケジュール ID と一致します。

<item>

ソースおよび宛先のファイル名と位置を示すエレメントが含まれるグループ・エレメント。

<source>

ソース・システムのファイルに関する <file> エレメントまたは <queue> エレメントと <checksum> エレメントが含まれるグループ・エレメント。

属性	説明
recursive	<source> エレメントがディレクトリーであるかそこにワイルドカード文字が含まれる場合にファイルが再帰的にサブディレクトリーに転送されることを指定します。

属性	説明
disposition	<p><source> がその宛先に正常に転送されたときに <source> エlementに対して取るアクションを指定します。有効なオプションは以下のとおりです。</p> <ul style="list-style-type: none"> • leave - ソース・ファイルは変更されません。 • delete - ソース・ファイルは、それが正常に転送された後にソース・システムから削除されます。
correlationBoolean	ブール相関値。ソースが Connect:Direct ブリッジの場合に、Connect:Direct プロセスがユーザー定義かどうかを指定します。
correlationString1	文字列相関値。ソースが Connect:Direct ブリッジの場合に、転送の宛先で発生する Connect:Direct プロセスの名前を指定します。
correlationNum1	数値相関値。ソースが Connect:Direct ブリッジの場合に、転送の宛先で発生する Connect:Direct プロセスの ID 番号を指定します。

<queue>

<source> エlementと一緒に使用する場合には、転送メッセージの読み取り元のキュー (ソース・エージェントのキュー・マネージャーに存在するキュー) の名前を指定します。

属性	説明
messageCount	キューから読み取られたメッセージの数。
groupId	キューから読み取られたメッセージの IBM MQ グループ ID。

<destination>

宛先に関する <file> Elementまたは <queue> Elementと <checksum> Elementが含まれるグループ・Element。

<file> と <queue> のいずれかを宛先の子Elementとして記述します。

属性	説明
タイプ	<p>宛先のタイプ。有効なオプションは以下のとおりです。</p> <ul style="list-style-type: none"> • queue - 宛先として IBM MQ キューを指定します • file - ファイルを宛先として指定します • directory - ディレクトリーを宛先として指定します •  dataset - z/OS データ・セットを宛先として指定します •  pds - z/OS 区分データ・セットを宛先として指定します <p>queue オプションを記述できるのは、<destination> Elementに子Element <queue> がある場合に限られます。その他のオプションを記述できるのは、<destination> Elementに子Element <file> がある場合に限られます。</p>
exist	<p>宛先システムに宛先ファイルが存在する場合に取る処置を示します。有効なオプションは以下のとおりです。</p> <ul style="list-style-type: none"> • error - エラーを報告し、ファイルは転送されません。 • overwrite - 既存の宛先ファイルを上書きします。 <p><destination> Elementに子Element <queue> がある場合は、この属性を記述できません。</p>
correlationBoolean	ブール相関値。宛先が Connect:Direct ブリッジの場合に、Connect:Direct プロセスがユーザー定義かどうかを指定します。

属性	説明
correlationString1	ストリング相関値。宛先が Connect:Direct ブリッジの場合に、転送の宛先で発生する Connect:Direct プロセスの名前を指定します。
correlationNum1	数値相関値。宛先が Connect:Direct ブリッジの場合に、転送の宛先で発生する Connect:Direct プロセスの ID 番号を指定します。

<file>

(ソースと宛先の両方で) 転送されたファイルの絶対パスを示します。この完全修飾パスは、ご使用のオペレーティング・システムと整合するフォーマットです (例えば C:/from/here.txt)。ファイル URI は使用されません。

<queue>

<destination> エlementと一緒に使用する場合には、転送先のキュー (宛先エージェント・キュー・マネージャーに接続しているいずれかのキュー・マネージャーに存在するキュー) の名前を指定します。

属性	説明
messageCount	キューに書き込まれたメッセージの数。
messageLength	キューに書き込まれたメッセージの長さ。
groupId	転送要求がファイルを複数のメッセージに分割するように指定した場合、この属性の値は、キューに書き込まれるメッセージの IBM MQ グループ ID です。
messageId	転送要求がファイルを複数のメッセージに分割するように指定しない場合、この属性の値は、キューに書き込まれるメッセージの IBM MQ メッセージ ID です。

<checksum>

オプションのエlement。

デジタル署名を作成するためにメッセージ・ダイジェストを生成したハッシュ・アルゴリズムのタイプを示します。現在のところ Managed File Transfer は、メッセージ・ダイジェスト・アルゴリズム 5 (MD5) だけをサポートします。転送されたファイルの完全性が損なわれていないことを確認する方法として、チェックサムが備えられています。

<malformed>

誤った形式のメッセージのグループ・element。

属性	説明
バージョン	
ID	
agentRole	ソース・エージェントまたは宛先エージェントのいずれか

<statistics>

転送の統計情報に関するグループ・element (使用可能な場合)。

<actualStartTime>

エージェントが転送の実行を開始した実際の時刻。通常、時刻は、転送について記録された開始時刻と同じ (またはそれに非常に近い) です。しかし、エージェントがビジー状態の場合、エージェントが転送を実行できるようになるまで、実行依頼された転送はキューに入れられることがあります。

<retryCount>

転送がリカバリー状態に入り、エージェントによって再試行された回数。ソースと宛先のエージェントが通信を失ったために、転送がリカバリー状態に入った可能性があります。その理由は、IBM MQ ネットワーク・エラーか、または両方のエージェントが一定の期間、データまたは確認応答メッセージを受信していないことのいずれかです。この期間は、エージェント・プロパティ transferAckTimeout および transferAckTimeoutRetries によって決定されます。

<numFileFailures>

transferSet に含まれているファイルのうち、正常に転送できなかったファイルの数。

<numFileWarnings>

transferSet に含まれているファイルのうち、転送時に警告が生成されたが、それ以外の点では正常に転送されたファイルの数。

例

このスキーマに準拠する XML メッセージの例が、以下の各転送タイプ用に提供されています。

- [単一ファイルの転送](#)
- [複数のファイルを含む転送](#)
- [失敗したファイル転送](#)
- [トリガーを使用して定義された転送](#)
- [スケジュールで開始された転送](#)
- [ユーザー出口を呼び出す転送](#)
- [Connect:Direct ブリッジ・ノード経由の転送](#)

単一転送ログ・メッセージの例

転送が起こるときに、メッセージは SYSTEM.FTE トピックに Log/agent_name/transfer_ID というトピック・ストリングでパブリッシュされます。次の XML の例では単一ファイル転送の開始、進行中、および完了のログ・メッセージが示されています。

単一ファイル転送 - 開始

```
<?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>
```

単一ファイル転送の成功 - 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>

```

単一ファイル転送の成功 - 完了

```

<?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>

```

複数ファイル転送ログ・メッセージの例

複数のファイルが含まれている転送が発生したときに、SYSTEM.FTE トピックに `Log/agent_name/transfer_ID` というトピック・ストリングでパブリッシュされるメッセージの例。

複数ファイル転送 - 開始

```
<?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>
```

複数ファイル転送 - 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>

```

複数ファイル転送 - 完了

```

<?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">414d51205553322e42494e44494e47538b0f404d036c0020</
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>

```

ファイル転送失敗のログ・メッセージの例

転送が起こるときに、メッセージは SYSTEM.FTE トピックに `Log/agent_name/transfer_ID` というトピック・ストリングでパブリッシュされます。次の XML の例では転送の開始時、進行中、および完了時にファイル転送が失敗した場合のログ・メッセージが示されています。

ファイル転送の失敗 - 開始

```

<?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>

```

ファイル転送の失敗 - 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>

```

ファイル転送の失敗 - 完了

```

<?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>

```



```

        <endExits>
          <exit name="class testExits.SourceExit1">
            <status>
              <supplement>Source End</supplement>
            </status>
          </exit>
        </endExits>
        <systemInfo architecture="x86" name="Windows 7"
          version="6.1 build 7601 Service Pack 1"/>
      </sourceAgent>
      <destinationAgent agent="FTEAGENT" QMgr="QM1">
        <startExits>
          <exit name="class testExits.DestinationExitProceed">
            <status resultCode="proceed">
              <supplement>Destination start, with proceed</supplement>
            </status>
          </exit>
        </startExits>
      </endExits>
      <exit name="class testExits.DestinationExitProceed">
        <status>
          <supplement>destination end</supplement>
        </status>
      </exit>
    </endExits>
    <systemInfo architecture="x86" name="Windows 7"
      version="6.1 build 7601 Service Pack 1"/>
  </destinationAgent>
  <originator>
    <hostName>reportserver.com</hostName>
    <userID>USER1</userID>
    <mqmdUserID>USER1      </mqmdUserID>
  </originator>
  <transferSet startTime="2008-11-02T22:36:13.046Z" total="1">
    <metaDataSet>
      <metaData key="newkey2">newvalue2</metaData>
      <metaData key="newkey1">newvalue1</metaData>
      <metaData key="newkey4">newvalue4</metaData>
      <metaData key="newkey3">newvalue3</metaData>
      <metaData key="newkey5">newvalue5</metaData>
      <metaData key="testkey1">testvalue1</metaData>
      <metaData key="testkey2">testvalue2</metaData>
    </metaDataSet>
  </transferSet>
</transaction>

<!--
  In this example the source transfer start exit has modified the
  metadata as follows:

  Added keys and values for:
  newkey1, newvalue1
  newkey2, newvalue2
  newkey3, newvalue3
  newkey4, newvalue4
  newkey5, newvalue5

  Replaced values for:
  key1 to modifiedValue1

  Deleted keys and values for:
  key2
-->

```

出口単一ファイル転送の取り消し - 取り消し

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
  ID="414d5120514d3120202020202020202020202020207e970d492000c702" 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>
  </sourceAgent>
</transaction>

```

```

        </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ブリッジの転送ログ・メッセージの例

宛先エージェントまたはソース・エージェントが Connect:Direct ブリッジ・エージェントの場合は、destinationAgent エlement または sourceAgent エlement に追加の属性が含まれます。開始ログ・メッセージには、Connect:Direct 転送に関する情報のサブセットだけが含まれます。進行中と完了のログ・メッセージには、Connect:Direct 転送に関する情報全体が含まれます。

ソース・エージェントが Connect:Direct ブリッジ・エージェントの場合

開始:

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
    ID="414d5120514d5f696b6b796f20202020a704654d20092507"
    agentRole="sourceAgent"
    version="4.00"
    xsi:noNamespaceSchemaLocation="TransferLog.xsd"
    xmlns="">
    <action time="2011-03-07T13:05:01.838Z">started</action>
    <sourceAgent QMgr="QM_KUIPER" agent="VARUNA" agentType="CD_BRIDGE" bridgeNode="CDNODE_VARUNA">
        <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
    </sourceAgent>
    <destinationAgent QMgr="QM_KUIPER" agent="IXION"/>
    <originator>
        <hostName>kuiper.example.com.</hostName>
        <userID>sol</userID>
        <mqmdUserID>sol</mqmdUserID>
    </originator>
    <transferSet bytesSent="0" startTime="2011-03-07T13:05:01.838Z" total="1">
        <metaDataSet>
            <metaData key="com.ibm.wmqfte.SourceAgent">VARUNA</metaData>
            <metaData key="com.ibm.wmqfte.DestinationAgent">IXION</metaData>
            <metaData key="com.ibm.wmqfte.MqmdUser">sol</metaData>
            <metaData key="com.ibm.wmqfte.OriginatingUser">sol</metaData>
            <metaData key="com.ibm.wmqfte.OriginatingHost">kuiper.example.com.</metaData>
            <metaData key="com.ibm.wmqfte.TransferId">414d5120514d5f696b6b796f20202020a704654d20092507</
metaData>
            <metaData key="com.ibm.wmqfte.Priority">0</metaData>
        </metaDataSet>
    </transferSet>
</transaction>

```

```
</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>
```

完了:

```
<?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>

```

宛先エージェントが Connect:Direct ブリッジ・エージェントの場合 開始:

```

<?xml version="1.0" encoding="UTF-8"?>
<transaction xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
    ID="414d5120514d5f696b6b796f20202020a704654d2008e102"
    agentRole="sourceAgent"
    version="4.00"
    xsi:noNamespaceSchemaLocation="TransferLog.xsd"
    xmlns="">
  <action time="2011-03-07T10:29:44.854Z">started</action>
  <sourceAgent QMgr="QM_asteroid" agent="PALLAS" agentType="STANDARD">
    <systemInfo architecture="x86" name="Windows 7" version="6.1 build 7601 Service Pack 1"/>
  </sourceAgent>
  <destinationAgent QMgr="QM_asteroid" agent="VESTA"/>
  <originator>
    <hostName>belt.example.com.</hostName>
    <userID>sol</userID>
    <mqmdUserID>sol</mqmdUserID>
  </originator>
  <transferSet bytesSent="0" startTime="2011-03-07T10:29:44.854Z" total="1">
    <metaDataSet>
      <metaData key="com.ibm.wmqfte.SourceAgent">PALLAS</metaData>
      <metaData key="com.ibm.wmqfte.DestinationAgent">VESTA</metaData>
      <metaData key="com.ibm.wmqfte.MqmdUser">sol</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingUser">sol</metaData>
      <metaData key="com.ibm.wmqfte.OriginatingHost">belt.example.com.</metaData>
      <metaData key="com.ibm.wmqfte.TransferId">414d5120514d5f696b6b796f20202020a704654d2008e102</
metaData>
      <metaData key="com.ibm.wmqfte.Priority">0</metaData>
    </metaDataSet>
  </transferSet>
</transaction>

```

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>

```

完了:

```
<?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>
```

スケジュール済みファイル転送ログ・メッセージ・フォーマット

エージェントがスケジュール済み転送の要求を処理するごとに、スケジュール・ログ・メッセージが調整キュー・マネージャー (SYSTEM.FTE/Log/agent name/schedule ID トピック上) にパブリッシュされます。このメッセージは、ScheduleLog.xsd XML スキーマに準拠しています。

スキーマ

次のスキーマは、スケジュール・ログ XML メッセージで有効なエレメントについて示します。

```
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
  <xsd:include schemaLocation="fteutils.xsd"/>

  <xsd:element name="schedulelog">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="originator" type="hostUserIDType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="action" type="actionType"
          maxOccurs="1" minOccurs="1"/>
        <xsd:element name="schedule" type="scheduleType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="sourceAgent" type="agentType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="destinationAgent" type="agentClientType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="status" type="statusType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="transferSet" type="transferSetType"
          maxOccurs="1" minOccurs="0"/>
        <xsd:element name="job" type="jobType"/>
```

```

                maxOccurs="1"                minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="version" type="versionType" use="required"/>
        <xsd:attribute name="ID" type="xsd:string" use="required"/>
    </xsd:complexType>
</xsd:element>

<xsd:complexType name="actionType">
    <xsd:simpleContent>
        <xsd:extension base="actionEnumType">
            <xsd:attribute name="time" type="xsd:dateTime" use="required"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:simpleType name="actionEnumType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="submit"/>
        <xsd:enumeration value="delete"/>
        <xsd:enumeration value="expire"/>
        <xsd:enumeration value="skipped"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="transferSetType">
    <xsd:sequence>
        <xsd:element name="item" type="itemType"
            maxOccurs="unbounded" minOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="size" type="xsd:int" use="required"/>
    <xsd:attribute name="priority" type="priorityType" use="optional"/>
</xsd:complexType>

<xsd:complexType name="itemType">
    <xsd:sequence>
        <xsd:element name="source" type="fileSourceType"
            maxOccurs="1" minOccurs="1"/>
        <xsd:element name="destination" type="fileDestinationType"
            maxOccurs="1" minOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="mode" type="modeType" use="required"/>
    <xsd:attribute name="checksumMethod" type="checkSumMethod" use="required"/>
</xsd:complexType>

</xsd:schema>

```

スケジュール・ログ・メッセージについて

スケジュール・ログ・メッセージで使用されるエレメントと属性について説明します。

<schedulelog>

単一の送信されたスケジュール済みファイル転送について説明するグループ・エレメント。

属性	説明
バージョン	Managed File Transfer で示されるこのエレメントのバージョンを示します。
ID	送信されたスケジュール・ファイル転送の固有 ID。

<originator>

要求の発信元を示すエレメントが含まれるグループ・エレメント。

<hostName>

ソース・ファイルが置かれているシステムのホスト名。

<userID>

ファイル転送を開始したユーザーの ID。

<mqmdUserID>

メッセージ記述子 (MQMD) に指定された MQ ユーザー ID

<action>

<schedulelog> エレメントの ID 属性に一致するスケジュール済み転送で実行するアクションを指定します。このエレメントは以下のいずれかの値でなければなりません。

- submit - 新しいスケジュール済み転送
- delete - スケジュール転送の取り消し
- expire - 処理しようとしているスケジュール転送項目
- skipped - エージェントがオフラインのため、スケジュールされていた転送は開始できません。エージェントが使用可能になると、転送がスキップされたことを示すためにこのメッセージはログに記録されます。

属性	説明
time	ログ項目がパブリッシュされた日時を示します (日時形式)。

<sourceAgent>

ソース・ファイルがあるシステム上のエージェントの名前を示します。

属性	説明
エージェント	エージェントの名前を示します。
キュー・マネージャー	エージェントのキュー・マネージャーの名前。

<destinationAgent>

ファイルの転送先にするシステム上のエージェントの名前を示します。

属性	説明
エージェント	エージェントの名前を示します。
キュー・マネージャー	エージェントのキュー・マネージャーの名前。

<status>

結果コードと補足メッセージ。

<transferSet>

まとめて実行するファイル転送のグループを示します。送信中は、<transferSet> は <item> エレメントが含まれるグループ・エレメントです。

属性	説明
size	転送項目の数を示します。
priority	転送の優先順位。priority は 0 から 9 までの範囲の値で、0 が最低の優先順位です。デフォルトの優先順位は 0 で、デフォルトではソース・エージェントの優先順位が転送で使用されます。

<item>

ソースおよび宛先のファイル名と位置を示すエレメントが含まれるグループ・エレメント。

属性	説明
mode	転送モードがバイナリーであるかテキストであるかを示します。
checksumMethod	デジタル署名を作成するためにメッセージ・ダイジェストを生成するハッシュ・アルゴリズムのタイプを示します。許可値は MD5 または none です。

<source>

ソース・システム上のファイルの <file> および <checksum> エレメントが含まれるグループ・エレメント。

属性	説明
recursive	<source> エレメントがディレクトリーであるかそこにワイルドカード文字が含まれる場合にファイルが再帰的にサブディレクトリーに転送されることを指定します。
disposition	<source> がその宛先に正常に転送されたときに <source> エレメントに対して取るアクションを指定します。有効なオプションは以下のとおりです。 <ul style="list-style-type: none"> • leave - ソース・ファイルは変更されません。 • delete - ソース・ファイルは、それが正常に転送された後にソース・システムから削除されます。

<destination>

宛先システム上のファイルの <file> および <checksum> エレメントが含まれるグループ・エレメント。

属性	説明
タイプ	宛先のファイルまたはディレクトリーのタイプ。有効なオプションは以下のとおりです。 <ul style="list-style-type: none"> • file - ファイルを宛先として指定します • directory - ディレクトリーを宛先として指定します •  dataset - z/OS データ・セットを宛先として指定します •  PDS - z/OS 区分データ・セットを宛先として指定します
exist	宛先システムに宛先ファイルが存在する場合に取る処置を示します。有効なオプションは以下のとおりです。 <ul style="list-style-type: none"> • error - エラーを報告し、ファイルは転送されません。 • overwrite - 既存の宛先ファイルを上書きします。

<file>

転送するファイルの名前を指定します。ご使用のオペレーティング・システムと整合するフォーマットの完全修飾パスを使用します (例えば C:/from/here.txt)。ファイル URI は使用しないでください。

属性	説明
encoding	テキスト・ファイル転送のエンコード。
EOL	行末マーカを指定します。指定できる値は、次のとおりです。 <ul style="list-style-type: none"> • LF - 改行文字のみ • CRLF - 復帰と改行の文字シーケンス

<job>

ジョブの詳細を指定するエレメントを含むグループ・エレメント。<job> は、転送が開始されたときに、ログ・メッセージに追加されるユーザー定義ジョブ名 ID です。この <job> エレメントは、転送要求メッセージに含まれる <job> エレメントと同じです。[780 ページの『ファイル転送要求メッセージ・フォーマット』](#) のトピックを参照してください。

<名前>

名前の値は、任意のストリングです。

例

このスキーマに準拠する XML メッセージの例が、以下のスケジュール済み転送操作用に提供されています。

- [スケジュール済み転送が作成される](#)

- スケジュール済み転送が取り消される
- スケジュール済み転送が期限切れになる

スケジュールにより開始された転送は、標準的な転送と同じ方法で記録されます。スケジュールにより開始された転送のログ・メッセージの例については、[763 ページの『スケジュール済みファイル転送ログ・メッセージの例』](#)を参照してください。

スケジュール・ファイル転送ログ・メッセージの例

スケジュール済み転送操作が発生したときに、SYSTEM.FTE トピックに `Log/agent_name/schedule_ID` というトピック・ストリングでパブリッシュされるメッセージの例。

スケジュール済み転送のログ・メッセージ

エージェントがスケジュール済み転送の要求を処理するごとに、スケジュール・ログ・メッセージが調整キュー・マネージャー (SYSTEM.FTE/Log/agent name/schedule ID トピック上) にパブリッシュされます。このメッセージは、ScheduleLog.xsd XML スキーマに準拠しています。以下に例を示します。

```
<?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>
```

このメッセージは以下の情報のログです。

- 要求の発信者
- 要求の送信日時
- スケジュール済み転送の開始日時
- ソース・エージェントと宛先エージェントの詳細
- 転送仕様

`<schedulelog>` エレメントの ID 属性は、このスケジュール済み転送 (ソース・エージェント内) に対する固有の ID です。この ID はスケジュール項目を実際のファイル転送と関連付けるために使用します。

`submit` の `<action>` エレメント値により、要求が受信されたことを確認します。

スケジュール済み転送取り消しのログ・メッセージ

保留中のスケジュール済みファイル転送を取り消す要求をエージェントが受け取ると、以下のメッセージが SYSTEM.FTE/Log/agent_name トピックにパブリッシュされます。

```
<?xml version="1.0" encoding="UTF-8"?>
<schedulelog version="1.00" ID="5"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance">
```

```

xsi:noNamespaceSchemaLocation="ScheduleLog.xsd">
<originator>
  <hostName>reportserver.com</hostName>
  <userID>USER1</userID>
</originator>
<action time="2008-11-23T21:56:27Z">delete</action>
<status resultCode="0"/>
</schedulelog>

```

ID 属性値は、スケジュール・メッセージ内の保留中の転送要求の ID と対応しています。

スケジュール済み転送満了のログ・メッセージ

現在時刻が、(<next> エレメントの値によって示される) スケジュール・リスト内の最も早い保留中ファイル転送の時間と一致すると、スケジュール済み転送項目が満了したことを示すために、スケジュール・ログ・メッセージがパブリッシュされます。

```

<?xml version="1.0" encoding="UTF-8"?>
<schedulelog xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00" ID="3"
  xsi:noNamespaceSchemaLocation="ScheduleLog.xsd">
  <originator>
    <hostName>example.com.</hostName>
    <userID>mqm</userID>
    <mqmdUserID>mqm</mqmdUserID>
  </originator>
  <action time="2011-01-26T13:03:26Z">expire</action>
  <sourceAgent agent="AGENT_JUPITER" QMgr="QM_JUPITER"/>
  <status resultCode="0"/>
</schedulelog>

```

「expire」の <action> エレメント値により、スケジュール項目がスケジュール・リストから除去されて処理中であることを確認します。エージェントに対するスケジュール・メッセージは、すでに存在しない満了項目でパブリッシュされます。

関連資料

769 ページの『[スケジュール済みファイル転送ログ・メッセージ・フォーマット](#)』

エージェントがスケジュール済み転送の要求を処理するごとに、スケジュール・ログ・メッセージが調整キュー・マネージャー (SYSTEM.FTE/Log/agent name/schedule ID トピック上) にパブリッシュされます。このメッセージは、ScheduleLog.xsd XML スキーマに準拠しています。

763 ページの『[スケジュール済みファイル転送ログ・メッセージの例](#)』

転送の進行中、メッセージは SYSTEM.FTE トピックに、Log/agent_name/transfer_ID というトピック・ストリングでパブリッシュされます。次の XML の例では、スケジュールの結果としてファイル転送が発生したときに作成されたログ・メッセージを示しています。

MFT モニター・ログ・メッセージ・フォーマット

モニター・ログ・メッセージは、SYSTEM.FTE トピックに、Log/agent_name/monitors/monitor_name/monitor_ID というトピック・ストリングでパブリッシュされます。

データを収集したりモニター操作を表示する場合は、対象となるモニターに合わせて調整されたワイルドカード・トピックのサブスクリプションをセットアップします。以下に例を示します。

```
Log/#
```

または、

```
Log/agent_name/#
```

このサブスクリプションは、永続または非永続のどちらにもできます。永続サブスクリプションは、キュー・マネージャーへのサブスクリプション・アプリケーションの接続が閉じられても存続します。非永続サブスクリプションは、サブスクリプション・アプリケーションのキュー・マネージャーへの接続が開いている間だけ存在します。

MonitorLog.xsd スキーマ文書は *MQ_INSTALLATION_PATH/mqft/samples/schema* ディレクトリーにあります。MonitorLog.xsd スキーマは、同じディレクトリーにある *fteutils.xsd* をインポートします。

スキーマ

次のスキーマは、モニター・ログXMLメッセージで有効なエレメントについて示します。

```
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
  <xsd:include schemaLocation="fteutils.xsd"/>
  <xsd:element name="monitorLog">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="originator" type="hostUserIDType" maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="references" type="referencesType" maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="action" type="monitorActionType" maxOccurs="1"
minOccurs="1"/>
        <xsd:element name="monitorAgent" type="agentType" maxOccurs="1"
minOccurs="1"/>
        <xsd:element name="status" type="statusType" maxOccurs="1"
minOccurs="1"/>
        <xsd:element name="monitorMetaData" type="monitorMetaDataType" maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="monitorExits" type="exitGroupType" maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="jobDetails" type="jobType" maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="taskXMLRequest" type="taskXMLRequestType" maxOccurs="1"
minOccurs="0"/>
        <xsd:element name="monitorXMLRequest" type="monitorXMLRequestType"
maxOccurs="1" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="version" type="versionType" use="required"/>
      <xsd:attribute name="monitorName" type="xsd:string" use="required"/>
      <xsd:attribute name="referenceId" type="xsd:string" use="optional"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="monitorActionType">
    <xsd:simpleContent>
      <xsd:extension base="monitorActionEnumType">
        <xsd:attribute name="time" type="xsd:dateTime" use="required"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:simpleType name="monitorActionEnumType">
    <xsd:restriction base="xsd:token">
      <xsd:enumeration value="create"/>
      <xsd:enumeration value="delete"/>
      <xsd:enumeration value="start"/>
      <xsd:enumeration value="stop"/>
      <xsd:enumeration value="triggerSatisfied"/>
      <xsd:enumeration value="triggerNotSatisfied"/>
      <xsd:enumeration value="triggerFail"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:complexType name="monitorMetaDataType">
    <xsd:sequence>
      <xsd:element name="originalMetaData" type="metaDataSetType" maxOccurs="1" minOccurs="0"/>
      <xsd:element name="updatedMetaData" type="metaDataSetType" maxOccurs="unbounded"
minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="taskXMLRequestType">
    <xsd:sequence>
      <xsd:element name="originalRequest" type="xsd:string" maxOccurs="1" minOccurs="1"/>
      <xsd:element name="updatedRequest" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="taskId" type="xsd:string" use="required"/>
  </xsd:complexType>
</xsd:schema>
```

```

<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>

```

モニター・ログ・メッセージについて

モニター・ログ・メッセージで使用する要素と属性を以下のリストにまとめます。

<monitorLog>

モニターにより実行された操作を記述する要素を含むグループ・要素。

属性	説明
バージョン	必須。モニター・リスト・メッセージ・フォーマットのバージョン。
monitorName	必須。モニターの名前。モニターが定義されているエージェントに固有です。
referenceId	モニター操作の ID。

<originator>

要求の発信元を示す要素が含まれるグループ・要素。

<hostName>

ソース・ファイルが置かれているシステムのホスト名。

<userID>

ファイル転送を開始したユーザーの ID。

<mqmdUserID>

オプション。メッセージ記述子 (MQMD) に指定された IBM MQ ユーザー ID。

<references>

このモニター操作に関連付けられている他のメッセージの ID への参照。

<createRequest>

モニターを作成するために使用された XML 要求メッセージのメッセージ ID。

<taskRequest>

このアクションの結果としてモニターがサブミットする XML 要求メッセージのメッセージ ID。

<action>

このログ・メッセージが関連づけられている、発生した操作。要素内の値は、create、delete、start、stop、triggerSatisfied、triggerNotSatisfied、または triggerFail のいずれかであることが可能です。

<monitorAgent>

リソースをモニターしているエージェント。

属性	説明
エージェント	必須。エージェントの名前。
キュー・マネージャー	オプション。エージェントの接続先のキュー・マネージャーの名前。

属性	説明
bridgeURL	オプション。エージェントがプロトコル・ブリッジ・エージェントの場合の、プロトコル・サーバーの URL。

<status>

ログ対象のリソース・モニター操作の状況。

属性	説明
resultCode	必須。操作の結果コード (整数)。

<supplement>

ログ対象のリソース・モニター操作の状況に関する追加情報。

<monitorMetaData>

<originalMetaData> エlementおよび <updatedMetaData> エlementを含むグループ・Element。

<originalMetaData>

操作が発生する前のモニターのメタデータを記述する 1つ以上の <metadata> Elementを含む Element。

<updatedMetaData>

操作が発生した後のモニターのメタデータを記述する 1つ以上の <metadata> Elementを含む Element。

<メタデータ>

メタデータのキー/値ペアを定義します。キーはElementの属性で値はElementの内容です。

属性	説明
key	メタデータのキー。

<monitorExits>

1つ以上の <exit> Elementを含むグループ・Element。

<exits>

リソース・モニターにより実行される出口を記述するElement。

属性	説明
名前	必須。リソース・モニター出口の名前。

<status>

ログ対象のリソース・モニター出口の状況。

属性	説明
resultCode	必須。出口の結果コード (整数)。

<supplement>

ログ対象のリソース・モニター出口の状況に関する追加情報。

<jobDetails>

単一の <name> Elementを含むElement。

<名前>

ジョブの名前。

<taskXMLRequest>

<originalRequest> Elementおよび <updatedRequest> Elementを含むグループ・Element。

属性	説明
taskId	タスク要求メッセージの ID。

<originalRequest>

モニターが実行するタスクのエスケープ XML 要求メッセージを含むエレメント。

<updatedRequest>

モニターが実行するタスクの、更新済みのエスケープ XML 要求メッセージを含むエレメント。

<monitorXMLRequest>

モニター XML 要求。

属性	説明
タイプ	必須。 <monitorXMLRequest> エレメント内のモニター XML 要求データのフォーマット。有効な値は escapedXML のみです。

例

このスキーマに準拠する XML メッセージの例が、以下のモニター操作用に提供されています。

- [モニターが作成される](#)
- [モニターがリソースをポーリングするときに、モニターの条件が満たされる](#)
- [モニターがリソースをポーリングするときに、モニターの条件が満たされない](#)
- [モニターが削除される](#)

関連資料

[778 ページの『MFT モニター・ログ・メッセージの例』](#)

モニター操作が発生したときに、SYSTEM.FTE トピックに `Log/agent_name/monitor_ID` というトピック・ストリングでパブリッシュされるメッセージの例。

MFT モニター・ログ・メッセージの例

モニター操作が発生したときに、SYSTEM.FTE トピックに `Log/agent_name/monitor_ID` というトピック・ストリングでパブリッシュされるメッセージの例。

モニター作成ログ・メッセージ

```
<?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>
```

モニター条件を満たしたログ・メッセージ

```
<?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>414d51205553322e42494e444494e47538b0f404d09430020</createRequest>
</references>
<action time="2011-01-26T12:56:46Z">triggerSatisfied</action>
<monitorAgent agent="US2.BINDINGS.FILE" QMgr="US2.BINDINGS"/>
<status resultCode="0"/>
<monitorMetaData>
  <originalMetaData>
    <metaData key="AGENTNAME">AGENT_JUPITER</metaData>
    <metaData key="LASTMODIFIEDDATEUTC">2011-01-26</metaData>
    <metaData key="CURRENTTIMESTAMPUTC">20110126125646793</metaData>
    <metaData key="CURRENTTIMESTAMP">20110126125646793</metaData>
    <metaData key="LASTMODIFIEDDATE">2011-01-26</metaData>
    <metaData key="FILENAME">new.completed</metaData>
    <metaData key="LASTMODIFIEDTIMEUTC">12.56</metaData>
    <metaData key="LASTMODIFIEDTIME">12.56</metaData>
    <metaData key="FILESIZE">0</metaData>
    <metaData key="FILEPATH">/srv/nfs/incoming/new.completed</metaData>
  </originalMetaData>
  <updatedMetaData>
    <metaData key="AGENTNAME">AGENT_JUPITER</metaData>
    <metaData key="LASTMODIFIEDDATEUTC">2011-01-26</metaData>
    <metaData key="CURRENTTIMESTAMPUTC">20110126125646793</metaData>
    <metaData key="CURRENTTIMESTAMP">20110126125646793</metaData>
    <metaData key="LASTMODIFIEDDATE">2011-01-26</metaData>
    <metaData key="FILENAME">new.completed</metaData>
    <metaData key="LASTMODIFIEDTIMEUTC">12.56</metaData>
    <metaData key="LASTMODIFIEDTIME">12.56</metaData>
    <metaData key="FILESIZE">0</metaData>
    <metaData key="FILEPATH">/srv/nfs/incoming/new.completed</metaData>
  </updatedMetaData>
</monitorMetaData>
<taskXMLRequest taskId="null">
  <originalRequest>&lt;?xml version="1.0" encoding="UTF-8"?&gt;&lt;request
    xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="4.00"
    xsi:noNamespaceSchemaLocation="FileTransfer.xsd"&gt;&lt;managedTransfer&gt;
      &lt;originator&gt;&lt;hostName&gt;example.com.&lt;/hostName&gt;
      &lt;userID&gt;mqm&lt;/userID&gt;&lt;/originator&gt;
      &lt;sourceAgent QMgr="QM_JUPITER" agent="AGENT_JUPITER"/&gt;
      &lt;destinationAgent QMgr="QM_JUPITER" agent="AGENT_SATURN"/&gt;
      &lt;transferSet&gt;&lt;item checksumMethod="MD5" mode="binary"&gt;
        &lt;source disposition="leave" recursive="false"&gt;
          &lt;file&gt;/srv/nfs/incoming/*.txt&lt;/file&gt;&lt;/source&gt;
          &lt;destination exist="error" type="directory"&gt;
            &lt;file&gt;/srv/backup&lt;/file&gt;&lt;/destination&gt;
          &lt;/item&gt;&lt;/transferSet&gt;&lt;/managedTransfer&gt;&lt;/request&gt;
        &lt;/originalRequest>
      <updatedRequest>&lt;?xml version="1.0" encoding="UTF-8"?&gt;&lt;request
        xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="4.00"
        xsi:noNamespaceSchemaLocation="FileTransfer.xsd"&gt;&lt;managedTransfer&gt;
          &lt;originator&gt;&lt;hostName&gt;example.com.&lt;/hostName&gt;
          &lt;userID&gt;mqm&lt;/userID&gt;&lt;/originator&gt;
          &lt;sourceAgent QMgr="QM_JUPITER" agent="AGENT_JUPITER"/&gt;
          &lt;destinationAgent QMgr="QM_JUPITER" agent="AGENT_SATURN"/&gt;
          &lt;transferSet&gt;&lt;item checksumMethod="MD5" mode="binary"&gt;
            &lt;source disposition="leave" recursive="false"&gt;
              &lt;file&gt;/srv/nfs/incoming/*.txt&lt;/file&gt;
              &lt;/source&gt;&lt;destination exist="error" type="directory"&gt;
                &lt;file&gt;/srv/backup&lt;/file&gt;&lt;/destination&gt;
              &lt;/item&gt;&lt;/transferSet&gt;&lt;/managedTransfer&gt;&lt;/request&gt;
            &lt;/updatedRequest>
          </taskXMLRequest>
</monitorLog>

```

モニター条件が満たされなかったログ・メッセージ

```

<?xml version="1.0" encoding="UTF-8"?>
<monitorLog xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="4.00"
  monitorName="MONITORONE"
  referenceId="414d51205553322e42494e444494e47538b0f404d09430020"
  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>

```

モニター削除ログ・メッセージ

```

<?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;&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>

```

ファイル転送要求メッセージ・フォーマット

ファイル転送は、エージェントのコマンド・キューに到着した XML メッセージによって開始されます。通常、ユーザーがファイル転送コマンドを発行したか、IBM MQ Explorer を使用した結果、XML メッセージが到着します。転送要求 XML は、FileTransfer.xsd スキーマに準拠していて、<request> エレメントをルート・エレメントとして持っている必要があります。FileTransfer.xsd スキーマ文書は MQ_INSTALLATION_PATH/mqft/samples/schema ディレクトリーにあります。FileTransfer.xsd スキーマは、同じディレクトリーにある fteutils.xsd をインポートします。

ファイル転送メッセージは、以下の 3 つのルート・エレメントのいずれかを含むことができます。

- <request> - 新しい転送要求、管理対象呼び出し要求、または保留中のスケジュール済み転送の削除のためのもの

- <cancel> - 進行中のファイル転送を取り消すためのもの
 - <transferSpecifications>- **fteCreateTransfer** コマンドで使用される複数の転送ファイル・グループを指定する場合
- <transferSpecifications> エレメントを使用して複数の転送グループを指定する方法については、[転送定義ファイルの使用](#)を参照してください。

スキーマ

次のスキーマは、転送要求 XML メッセージで有効なエレメントについて示します。

```
<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:sequence>
</xsd:complexType>

```

```

        <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>

```

転送要求メッセージについて

転送要求メッセージで使用するエレメントと属性を以下のリストにまとめます。

エレメントの説明

<request>

ファイル転送要求を指定するために必要なすべてのエレメントが含まれるグループ・エレメント。

属性	説明
バージョン	Managed File Transfer によって指定された、このエレメントのバージョンを示します。

<managedTransfer>

1つのファイル転送または1つのファイル転送グループに必要なすべてのエレメントが含まれるグループ・エレメント。

<deleteScheduledTransfer>

スケジュール転送を取り消すための発信元と ID 情報が含まれるグループ・エレメント。

<managedCall>

プログラムまたは実行可能ファイルの1つの管理対象呼び出しに必要なすべてのエレメントが含まれるグループ・エレメント。

<ID>

保留中のスケジュール済み転送のリストから削除する転送要求を指定する固有 ID。

<originator>

要求の発信元を示すエレメントが含まれるグループ・エレメント。

<hostName>

ソース・ファイルが置かれているシステムのホスト名。

<userID>

ファイル転送を開始したユーザーの ID。

<mqmdUserID>

オプション。メッセージ記述子 (MQMD) に指定された IBM MQ ユーザー ID。

<schedule>

ファイル転送の予定時間、繰り返し動作、および次の実行予定を示すグループ・エレメント。

<submit>

スケジュール済み転送の開始日時を示します。

属性	説明
timebase	使用するタイム・ゾーンを示します。この属性には、以下のいずれかの値を使用できます。 <ul style="list-style-type: none"> • source - ソース・エージェントのタイム・ゾーンを使用します。 • admin - コマンドを発行する管理者のタイム・ゾーンを使用します。 • UTC - 協定世界時を使用します。
timezone	時間基準値に従ったタイム・ゾーンの記述

<repeat>

スケジュール済み転送の繰り返し頻度、スケジュール済み転送の繰り返し回数、およびスケジュール済み転送の繰り返しの停止日時の詳細を含むグループ・エレメント。

<frequency>

転送を繰り返す時間間隔。

属性	説明
interval	以下のいずれかで示される間隔の単位。

属性	説明
	<ul style="list-style-type: none"> • minutes • hours • 日 • weeks • months • years

<expireTime>

スケジュール済み転送の繰り返しを停止する日時を指定するオプションの要素。この要素と <expireCount> 要素は相互に排他的です。

<expireCount>

スケジュール済みファイル転送を終了するまでの実行回数を指定するオプションの要素。この要素と <expireTime> 要素は相互に排他的です。

<sourceAgent>

ソース・ファイルがあるシステム上のエージェントの名前を示します。

属性	説明
エージェント	エージェントの名前を示します。
キュー・マネージャー	エージェントのキュー・マネージャーの名前。

<destinationAgent>

ファイルの転送先にするシステム上のエージェントの名前を示します。

属性	説明
エージェント	エージェントの名前を示します。
キュー・マネージャー	エージェントのキュー・マネージャーの名前。
hostName	エージェントのキュー・マネージャーのホスト名または IP アドレス。
portNumber	宛先エージェントのキュー・マネージャーとのクライアント接続で使用されるポート番号。
channel	宛先エージェントのキュー・マネージャーとの接続に使用されるチャンネル名。

<trigger>

ファイル転送の実行のために満たす必要がある条件を指定するオプションの要素。

属性	説明
log	<p>トリガー障害がログに記録されるかどうかを示すフラグ。有効な値は以下のとおりです。</p> <ul style="list-style-type: none"> • yes - 失敗したトリガー転送についてのログ項目が作成されます。 • no - 失敗したトリガー転送についてのログ項目が作成されません。

<fileExist>

ソース・エージェントと同じシステムにあるファイル名のコンマ区切りリストを指定します。この名前リスト内のファイルがトリガーの条件を満たす場合、転送が実行されます。この要素と <fileSize> 要素は相互に排他的です。

属性	説明
comparison	名前リストに対してソース・ファイル名を評価する方法を示します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • = 名前リスト内の少なくとも1つのファイル名が一致しなければなりません。 • != 名前リストの少なくとも1つのファイルが存在しません。
値	比較タイプを示します。 <ul style="list-style-type: none"> • exist: ファイルが存在しなければなりません

<fileSize>

ソース・エージェントと同じシステムにあるファイル名のコンマ区切りリストを指定します。この名前リスト内のファイルがトリガーの条件を満たす場合、転送が実行されます。このエレメントと <fileExist> エレメントは相互に排他的です。

属性	説明
comparison	名前リストに対してソース・ファイル名を評価する方法を示します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • >= 名前リスト内のファイル名のいずれかが存在し、value 属性で指定した最小サイズを有しています。
値	次のいずれかの単位を持つ整数値で指定したファイル・サイズ <ul style="list-style-type: none"> • B - バイト • KB - キロバイト • MB - メガバイト • GB - ギガバイト (単位値の大/小文字は区別されません)

<reply>

同期ファイル転送用に生成された一時応答キューの名前を示します(コマンド行で **-w** パラメーターにより指定します)。キューの名前は、`command.properties` 構成ファイル内のキー **dynamicQueuePrefix**、またはデフォルトの `WMQFTE.*` によって定義されます。指定されていない場合、

属性	説明
detailed	応答メッセージで詳細な転送結果情報が必要かどうか。転送ごとに複数の応答メッセージを生成できます。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • true - 詳細な応答情報が必要です。情報の形式は、転送ログの進行メッセージ(つまり <transferSet> エレメント)に公開されるものと同じです。詳しくは、748 ページの『ファイル転送ログ・メッセージ・フォーマット』を参照してください。詳細な応答情報は、転送ソース・エージェントの <code>enableDetailedReplyMessages</code> プロパティが true に設定されている場合にのみ存在します。 • false - 詳細な応答情報は不要です。 デフォルト値は false です。
QMGR	応答を受け取るために一時動的キューが生成されるコマンド・キュー・マネージャーの名前。
persistent	応答キューに書き込まれるメッセージが永続メッセージかどうか。有効な値は以下のとおりです。

属性	説明
	<ul style="list-style-type: none"> • true - メッセージは永続メッセージです • false - メッセージは永続メッセージではありません • qdef - メッセージの永続性を応答キューのプロパティで定義します デフォルト値は false です。

<transferSet>

まとめて実行するファイル転送のグループ、またはまとめて実行する管理対象呼び出しのグループを示します。送信中は、<transferSet> は <item> エレメントが含まれるグループ・エレメントです。

属性	説明
priority	転送の優先順位。priority は 0 から 9 までの範囲の値で、0 が最低の優先順位です。デフォルトの優先順位は 0 で、デフォルトではソース・エージェントの優先順位が転送で使用されます。

<metaDataSet>

1 つ以上のメタデータ項目を含むオプションのグループ・エレメント。

<metaData>

エージェントにより呼び出される出口点に渡される、ユーザー定義のメタデータを指定します。このエレメントは、ストリングとしてメタデータ値を含みます。

属性	説明
key	ストリングとしてのメタデータ名

<call>

呼び出すプログラムまたは実行可能ファイルを指定する <command> エレメントを含むグループ・エレメント。

<コマンド>

呼び出すプログラムまたは実行可能ファイルを指定します。このコマンドは、エージェント・コマンド・パスに存在していなければなりません。詳しくは、[552 ページの表 45](#) を参照してください。このエレメントには、オプションの <argument> エレメントを含めることができます。

属性	説明
名前	コマンドの名前。
successRC	このコマンドによって返される、コマンドの成功を示す戻りコード。デフォルトは 0 です。
retryCount	コマンドが失敗したときに、コマンドを再試行する回数。
retryWait	コマンドを次に再試行するまで待機する時間 (秒数)。
タイプ	呼び出すプログラムのタイプ。有効な値は antscript、jcl、または executable です。

<argument>

コマンドに渡す引数を指定します。

<item>

ソースおよび宛先のファイル名と位置を示すエレメントが含まれるグループ・エレメント。

属性	説明
mode	転送モードがバイナリーであるかテキストであるかを示します。

属性	説明
checksumMethod	デジタル署名を作成するためにメッセージ・ダイジェストを生成するハッシュ・アルゴリズムのタイプを示します。有効な値は MD5 または none です。

<source>

ソース・システム上のファイルと、転送完了後にそれらのファイルを除去するかどうかを指定するグループ・エレメント

属性	説明
recursive	<source> エレメントがディレクトリーであるかそこにワイルドカード文字が含まれる場合にファイルが再帰的にサブディレクトリーに転送されることを指定します。
disposition	<source> がその宛先に正常に転送されたときに <source> エレメントに対して取るアクションを指定します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • leave - ソース・ファイルは変更されません。 • delete - ソース・ファイルは、それが正常に転送された後にソース・システムから削除されます。

<file>

転送ソースを指定します。 **Multi** Multiplatforms の場合、転送ソースは、ファイルまたはディレクトリー名にすることができます。z/OS プラットフォームでは、ファイル名、ディレクトリー名、データ・セット名、または PDS 名を転送のソースにすることができます。ご使用のオペレーティング・システムと整合するフォーマットの完全修飾パスを使用します (例えば C:/from/here.txt)。ファイル URI は使用しないでください。

属性	説明
別名	ソース・ファイルの別名を指定します。この別名は、転送で指定されているディレクトリー・パスを除外したソース・ファイルの名前です。
EOL	テキスト転送の行末マーカを指定します。有効な値は次のとおりです。 <ul style="list-style-type: none"> • LF - 改行文字のみ • CRLF - 復帰と改行の文字シーケンス
encoding	テキスト・ファイル転送のソース・ファイルのエンコード。
z/OS delimiter	レコード単位ソース・ファイル (z/OS データ・セットなど) でレコードとレコードの間に組み込まれる区切り文字を指定します。区切り文字の値は、00-FF の範囲にある 2 桁の 16 進数に接頭部 x を付けた形式で指定してください。例えば、x12 または x03,x7F などです。
delimiterType	宛先ファイルで個々のメッセージ・データの後に組み込む区切り文字のタイプを指定します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • binary - 16 進数区切り文字 この属性は、7.0.4.1 機能を有効にした場合にのみ使用できます。
delimiterPosition	レコード単位ソース・ファイル・レコードを通常のファイルに書き込むときに、区切り文字を挿入する位置を指定します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • prefix - 区切り文字は、宛先ファイルの各ソース・レコード単位ファイル・レコードのデータの前に挿入されます。 • postfix - 区切り文字は、宛先ファイルの、ソース・レコード単位の各ファイル・レコードからのデータの後に挿入されます。

属性	説明
includeDelimiterInFile	レコード単位ソース・ファイルでレコードとレコードの間に区切り文字を組み込むかどうかを指定します。
 keepTrailingSpaces	<p>テキスト・モード転送の一部として固定長形式のデータ・セットから読み取られるソース・レコードの末尾スペースを保持するかどうかを指定します。デフォルトでは、末尾スペースは削除されます。有効な値は以下のとおりです。</p> <ul style="list-style-type: none"> • true - テキスト・モード転送の一部として固定長形式のデータ・セットから読み取られるソース・レコードの末尾スペースを保持します • false - テキスト・モード転送の一部として固定長形式のデータ・セットから読み取られるソース・レコードから末尾スペースを削除します

<queue>

<source> エlementと一緒に使用する場合には、転送元のキューの名前を指定します。そのキューは、ソース・エージェントのキュー・マネージャーに存在していなければなりません。 **QUEUE** というフォーマットを使用してください。キュー・マネージャー名を組み込むことはできません。ソース・エージェントのキュー・マネージャーに存在しているキューを指定する必要があります。 <queue> Elementを <destination> Element内で使用した場合、 <source> Element内では使用できません。

属性	説明
useGroups	<p>完全に揃った最初のメッセージ・グループだけをソース・キューから転送するかどうかを指定します。有効な値は以下のとおりです。</p> <ul style="list-style-type: none"> • true - 完全に揃った最初のメッセージ・グループだけを転送します • false - ソース・キューにあるすべてのメッセージを転送します
groupId	<p>ソース・キューから読み取るメッセージ・グループを指定します。この属性は、useGroups 属性の値が true の場合のみ有効です。</p>
delimiterType	<p>宛先ファイルで個々のメッセージ・データの後に組み込む区切り文字のタイプを指定します。有効な値は以下のとおりです。</p> <ul style="list-style-type: none"> • text - テキスト区切り文字または Java リテラル区切り文字 • binary - 16 進数区切り文字
delimiter	<p>宛先ファイルで個々のメッセージ・データ間に組み込む区切り文字を指定します。</p>
delimiterPosition	<p>宛先ファイルで、区切り文字を個々のメッセージ・データの前に組み込むか、後に組み込むかを指定します。有効な値は以下のとおりです。</p> <ul style="list-style-type: none"> • prefix - 区切り文字はデータの前に組み込まれます • postfix - 区切り文字はデータの後ろに組み込まれます。
encoding	<p>ソース・キューのエンコード方式を指定します。</p>
waitTime	<p>ソース・エージェントが以下のいずれかの状態の発生を待つ時間を秒単位で指定します。</p> <ul style="list-style-type: none"> • キューがもともと空だったか、後で空になった場合は、ソース・キューにメッセージが出現する、という状態 • useGroups 属性が true に設定されている場合は、完全に揃ったグループがソース・キューに出現する、という状態

属性	説明
	waitTime 値の設定については、682 ページの『メッセージからファイルへの転送の待機時間を指定する際のガイダンス』を参照してください。

<destination>

宛先と、宛先エージェントにファイルが存在する場合の動作を指定するグループ・エレメント。

<file> と <queue> のいずれかを destination の子エレメントとして指定できます。

属性	説明
タイプ	<p>宛先のタイプ。有効な値は以下のとおりです。</p> <ul style="list-style-type: none"> • file - ファイルを宛先として指定します • directory - ディレクトリーを宛先として指定します •  dataset - z/OS データ・セットを宛先として指定します •  pds - z/OS 区分データ・セットを宛先として指定します • queue - 宛先として IBM MQ キューを指定します • filespace - 宛先としてファイル・スペースを指定します <p>値 queue が有効なのは、<destination> エレメントに子エレメント <queue> がある場合に限られます。</p> <p>値 filespace が有効なのは、<destination> エレメントに子エレメント <filespace> がある場合に限られます。</p> <p>その他の値は、<destination> エレメントに子エレメント <file> がある場合に限り有効です。</p>
exist	<p>宛先システムに宛先ファイルが存在する場合に取る処置を示します。有効な値は以下のとおりです。</p> <ul style="list-style-type: none"> • error - エラーを報告し、ファイルは転送されません。 • overwrite - 既存の宛先ファイルを上書きします。 <p><destination> エレメントに子エレメント <queue> または <filespace> がある場合は、この属性が無効になります。</p>

<file>

前述の <destination> エレメントの追加設定を指定します。ご使用のオペレーティング・システムと整合するフォーマットの完全修飾パスを使用します (例えば C:/from/here.txt)。ファイル URI は使用しないでください。

属性	説明
別名	<destination> ファイルの別名を指定します。この別名は、転送で指定されているディレクトリー・パスを除外したソース・ファイルの名前です。
encoding	テキスト・ファイル転送の <destination> ファイルのエンコード。
EOL	<p>テキスト転送の行末マーカを指定します。有効な値は次のとおりです。</p> <ul style="list-style-type: none"> • LF - 改行文字のみ • CRLF - 復帰と改行の文字シーケンス
truncateRecords	オプション。LRECL データ・セット属性よりも長い <destination> レコードが切り捨てられることを指定します。

属性	説明
	<ul style="list-style-type: none"> • true - それらのレコードは切り捨てられます • false - それらのレコードは折り返されます デフォルト設定は false です。

<queue>

<destination> エlementと一緒に使用する場合には、転送先のキューの名前を指定します。そのキューは、宛先エージェントのキュー・マネージャーに接続しているいずれかのキュー・マネージャーに存在していなければなりません。 *QUEUE@QM* というフォーマットを使用してください (*QUEUE* は、メッセージの書き込み先のキューの名前、*QM* は、そのキューが存在しているキュー・マネージャーです)。
<queue> エlementを **<source>** Element内で使用した場合、 **<destination>** Element内では使用できません。

属性	説明
delimiter	ファイルを複数のメッセージに分割するための区切り文字。
delimiterType	区切り文字のタイプを指定します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • text - Java 正規表現 • binary - 16 進数バイトのシーケンス • size - バイト数、キビバイト数、メビバイト数のいずれか。例えば、1 B、1 K、または 1 M のようにします。
delimiterPosition	区切り文字が個々のメッセージ内で、データの前に組み込まれると想定されているか、データの後ろに組み込まれると想定されているかを指定します。有効なオプションは以下のとおりです。 <ul style="list-style-type: none"> • prefix - 区切り文字はデータの前にあると想定されています • postfix - 区切り文字はデータの後ろにあると想定されています
includeDelimiterInMessage	ファイルを複数のメッセージに分割するために使用した区切り文字をそれらのメッセージの末尾に組み込むかどうかを指定するブール値。
encoding	宛先キューのエンコード方式を指定します。
persistent	メッセージが永続メッセージであるかどうかを指定します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • true - メッセージは永続メッセージです • false - メッセージは永続メッセージではありません • qdef - メッセージの永続性の値を宛先キューの設定で定義します
setMqProps	ファイルの最初のメッセージで IBM MQ メッセージ・プロパティを設定するかどうか、およびエラーの発生時にキューにメッセージを書き込むかどうかを指定するブール値。
unrecognisedCodePage	宛先キュー・マネージャーがデータのコード・ページを認識できない場合に、テキスト・モードの転送が失敗するか、それとも変換が実行されるかを指定します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • fail - 転送は失敗を報告します。 • binary - データは宛先コード・ページに変換され、データの形式を記述する IBM MQ メッセージ・ヘッダーは MQFMT_NONE に設定されます。 デフォルトの動作は fail です。

<filespace>

転送先のファイル・スペースの名前を指定するグループ・エレメント。

<名前>

<filespace> エレメントと一緒に使用する場合には、ファイル・スペースの名前を値として指定します。

<preSourceCall>

転送のソースで転送の開始前に呼び出すコマンドを指定するグループ・エレメント。

<postSourceCall>

転送のソースで転送の完了後に呼び出すコマンドを指定するグループ・エレメント。

<preDestinationCall>

転送の宛先で転送の開始前に呼び出すコマンドを指定するグループ・エレメント。

<postDestinationCall>

転送の宛先で転送の完了後に呼び出すコマンドを指定するグループ・エレメント。

<コマンド>

<preSourceCall>、<postSourceCall>、<preDestinationCall>、<postDestinationCall> のいずれかのエレメントと一緒に使用する場合には、呼び出すコマンドを指定します。このコマンドは、エージェント・コマンド・パスに存在していなければなりません。詳しくは、[552 ページの表 45](#) を参照してください。

属性	説明
名前	実行するコマンドの名前。
successRC	コマンドが正常に実行された場合に返される戻りコード。

<argument>

<command> エレメントと一緒に使用する場合には、コマンドに渡す引数を指定します。<command> エレメントの中に <argument> エレメントをいくつでも記述できます。

<job>

転送仕様全体のジョブ情報を含むオプションのグループ・エレメント。<job> は、転送が開始されたときに、ログ・メッセージに追加されるユーザー定義ジョブ名 ID です。この<job> エレメントは、転送要求メッセージに含まれる <job> エレメントと同じです。[748 ページの『ファイル転送ログ・メッセージ・フォーマット』](#) のトピックを参照してください。

<名前>

<job> エレメントと一緒に使用する場合には、ジョブの名前を値として指定します。

<transferSpecifications>

複数の転送グループの <item> エレメントを含むグループ・エレメント。このエレメントの使用方法の詳細については、『[転送定義ファイルの使用](#)』を参照してください。

<cancel>

進行中のファイル転送を取り消すために必要なすべてのエレメントが含まれるグループ・エレメント。

属性	説明
バージョン	Managed File Transfer によって指定された、このエレメントのバージョンを示します。

<transfer>

<cancel> エレメントと一緒に使用する場合、このエレメントの値は取り消す転送要求 ID を示します。

<job>

ジョブ情報が入ったグループ・エレメント。

<jobName>

論理ジョブ ID を指定します。

ファイル転送取り消しメッセージ・フォーマット

ファイル転送要求は、特定のエージェントに対する転送を識別する 48 文字の ID を返します。この ID は転送を取り消すために使用されます。

転送取り消しメッセージについて

転送取り消しメッセージで使用される要素と属性について説明します。

<cancel>

進行中のファイル転送を取り消すために必要なすべての要素が含まれるグループ・要素。

属性	説明
バージョン	Managed File Transfer によって指定された、この要素のバージョンを示します。

<originator>

要求の発信元を示す要素が含まれるグループ・要素。

<hostName>

ソース・ファイルが置かれているシステムのホスト名。

<userID>

ファイル転送を開始したユーザーの ID。

<mqmdUserID>

オプション。メッセージ記述子 (MQMD) に指定された IBM MQ ユーザー ID。

<transfer>

<cancel> 要素と一緒に使用する場合は、この要素の値は取り消す転送要求 ID を示します。

<job>

オプション。ジョブ情報が入ったグループ・要素。

<jobName>

論理ジョブ ID を指定します。

例

このスキーマに準拠する XML メッセージの例が、以下の要求用に提供されています。

- [ファイル転送の作成](#)
- [非同期ファイル転送の作成要求](#)
- [ファイル転送を取り消す](#)
- [スケジュール済み転送の作成](#)
- [スケジュール済み転送の削除](#)
- [管理対象呼び出しの作成](#)
- [管理対象呼び出しを含むファイル転送の作成](#)

ファイル転送要求メッセージの例

エージェントによる転送の作成または取り消しを要求するために、エージェントのコマンド・キューに PUT できるメッセージの例。

転送作成要求

```
<?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>

```

転送作成要求 - 同期

ユーザーがブロック化同期要求を行う場合、すなわち、転送が完了するのを待って状況メッセージを受け取る場合、コマンド・キューに置かれたメッセージには、応答メッセージの送信先キューを指定した応答エレメントが含まれています。以下の例は、FTEAGENT が使用するコマンド・キューに置かれたメッセージを示しています。

```

<?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>

```

<reply> エレメントには、転送の正常終了（または異常終了）についての応答を受け取るための一時動的キューが作成されたコマンド・キュー・マネージャーの名前が指定されています。一時動的キューの名前は2つの部分で構成されています。

- `command.properties` 構成ファイル (WMQFTE) 内のキー **dynamicQueuePrefix** によって定義される接頭部。デフォルト)
- IBM MQ が生成したキューの ID

転送取り消し要求

```

<?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>414D51205553322E42494E44494E47538B0F404D032C0020</transfer>  
<reply QMGR="QM_JUPITER">WMQFTE.4D400F8B20002007</reply>  
</cancel>
```

関連資料

[780 ページの『ファイル転送要求メッセージ・フォーマット』](#)

ファイル転送は、エージェントのコマンド・キューに到着した XML メッセージによって開始されます。通常、ユーザーがファイル転送コマンドを発行したか、IBM MQ Explorer を使用した結果、XML メッセージが到着します。転送要求 XML は、FileTransfer.xsd スキーマに準拠していて、<request> エレメントをルート・エレメントとして持っている必要があります。FileTransfer.xsd スキーマ文書は MQ_INSTALLATION_PATH/mqft/samples/schema ディレクトリーにあります。FileTransfer.xsd スキーマは、同じディレクトリーにある fteutils.xsd をインポートします。

スケジュール済みファイル転送メッセージの例

エージェントによるスケジュールの作成または削除を要求するために、エージェントのコマンド・キューに PUT できるメッセージの例。

スケジュール済み転送の作成

```
<?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>
```

スケジュール済み転送の削除

```
<?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>
```

関連資料

[780 ページの『ファイル転送要求メッセージ・フォーマット』](#)

ファイル転送は、エージェントのコマンド・キューに到着した XML メッセージによって開始されます。通常、ユーザーがファイル転送コマンドを発行したか、IBM MQ Explorer を使用した結果、XML メッセージが到着します。転送要求 XML は、FileTransfer.xsd スキーマに準拠していて、<request> エレメントをルート・エレメントとして持っている必要があります。FileTransfer.xsd スキーマ文書は MQ_INSTALLATION_PATH/mqft/samples/schema ディレクトリーにあります。FileTransfer.xsd スキーマは、同じディレクトリーにある fteutils.xsd をインポートします。

MFT エージェント呼び出し要求メッセージの例

エージェント・コマンド・キューに置いて、そのエージェントが管理呼び出しを作成するか、プログラムを呼び出す転送を作成するよう要求するメッセージの例。

管理対象呼び出し要求の例

```
<?xml version="1.0" encoding="UTF-8"?>
<request xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="1.00"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedCall>
    <originator>
      <hostName>example.com.</hostName>
      <userID>mqm</userID>
    </originator>
    <agent agent="DNWE" QMgr="QM1"/>
    <transferSet>
      <call>
        <command name="echo" successRC="0">
          <argument>call</argument>
          <argument>test</argument>
        </command>
      </call>
    </transferSet>
  </job>
  <name>managedCallCalls.xml</name>
</managedCall>
</request>
```

呼び出しを使用したマネージド転送要求の例

```
<?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>

```

関連概念

250 ページの『MFT で実行するプログラムの指定』

Managed File Transfer Agent が実行されているシステムで、プログラムを実行することができます。ファイル転送要求の一部として、転送の開始前または終了後のいずれかにプログラムを実行するように指定することができます。また、管理対象呼び出し要求を実行依頼することで、ファイル転送要求に含まれないプログラムを開始することも可能です。

関連資料

780 ページの『ファイル転送要求メッセージ・フォーマット』

ファイル転送は、エージェントのコマンド・キューに到着した XML メッセージによって開始されます。通常、ユーザーがファイル転送コマンドを発行したか、IBM MQ Explorer を使用した結果、XML メッセージが到着します。転送要求 XML は、FileTransfer.xsd スキーマに準拠していて、<request> エレメントをルート・エレメントとして持っている必要があります。FileTransfer.xsd スキーマ文書は MQ_INSTALLATION_PATH/mqft/samples/schema ディレクトリーにあります。FileTransfer.xsd スキーマは、同じディレクトリーにある fteutils.xsd をインポートします。

MFT モニター要求メッセージ・フォーマット

リソース・モニターは、通常はユーザーによる fteCreateMonitor コマンドの発行または IBM MQ Explorer インターフェースの使用の結果、適切な XML メッセージがエージェントのコマンド・キューに到達すると作成されます。

モニター XML は、ルート・エレメントとして <monitor> エレメントを使用する Monitor.xsd スキーマに準拠している必要があります。

モニター・メッセージには、以下の 3 つのルート・エレメントのいずれかを含むことができます。

- <monitor> - 新規リソース・モニターの作成および開始用
- <deleteMonitor> - 既存モニターの停止および削除用

fteListMonitors コマンドは、直接 SYSTEM.FTE トピックから一致するモニター定義を取り出すため、このコマンドのコマンド・メッセージはありません。

スキーマ

次のスキーマは、モニター要求 XML メッセージで有効なエレメントについて示します。

```

<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"/>
        <xsd:element name="reply" type="replyType"
        maxOccurs="1"
        minOccurs="0"/>
        <xsd:element name="tasks" type="monitorTasksType"
        maxOccurs="1"
        minOccurs="1"/>
        <xsd:element name="originator" type="origRequestType"
        maxOccurs="1"
        minOccurs="1"/>
        <xsd:element name="job" type="jobType"
        maxOccurs="1"
        minOccurs="0"/>
        <xsd:element name="defaultVariables" type="defaultVariablesType"
        maxOccurs="1"
        minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="version" type="versionType" use="required"/>
</xsd:complexType>
</xsd:element>

<xsd:element name="deleteMonitor">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="name" type="monitorNameType"
            minOccurs="1"
            maxOccurs="1"/>
            <xsd:element name="originator" type="origRequestType"
            maxOccurs="1"
            minOccurs="1"/>
            <xsd:element name="reply" type="replyType"
            maxOccurs="1"
            minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="version" type="versionType" use="required"/>
    </xsd:complexType>
</xsd:element>

<xsd:complexType name="transferRequestType">
    <xsd:choice>
        <xsd:element name="managedTransfer" type="managedTransferType"/>
        <xsd:element name="managedCall" type="managedCallType"/>
    </xsd:choice>
    <xsd:attribute name="version" type="versionType"/>
</xsd:complexType>

<xsd:complexType name="monitorResourcesType">
    <xsd:choice>
        <xsd:sequence>
            <xsd:element name="directory" type="monitoredDirectoryType"
            minOccurs="1"
            maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:element name="queue" type="monitoredQueueType"/>
    </xsd:choice>
</xsd:complexType>

<xsd:complexType name="monitoredDirectoryType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute name="recursionLevel"
            type="xsd:nonNegativeInteger"/>
            <xsd:attribute name="id" type="resourceIdAttrType"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="monitoredQueueType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute name="id" type="resourceIdAttrType"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="triggerMatchType">
    <xsd:sequence>
        <xsd:element name="conditions" type="conditionsType"
        minOccurs="1"
        maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="conditionsType">
    <xsd:choice minOccurs="1">
        <xsd:element name="allOf" type="listPredicateType"
        minOccurs="1"
        maxOccurs="1"/>

```

```

        <xsd:element name="anyOf" type="listPredicateType"
          minOccurs="1" maxOccurs="1"/>
        <xsd:element name="condition" type="conditionType"
          minOccurs="1" maxOccurs="1"/>
      </xsd:choice>
    </xsd:complexType>

    <xsd:complexType name="listPredicateType">
      <xsd:choice>
        <xsd:element name="condition" type="conditionType"
          minOccurs="1" maxOccurs="unbounded"/>
      </xsd:choice>
    </xsd:complexType>

    <xsd:complexType name="conditionType">
      <xsd:sequence>
        <xsd:element name="name" type="conditionNameType"
          minOccurs="0" maxOccurs="1"/>
        <xsd:element name="resource" type="resourceIdType"
          minOccurs="0" maxOccurs="1"/>
        <xsd:choice minOccurs="1">
          <xsd:element name="fileMatch"
            type="fileMatchConditionType"
            minOccurs="1" maxOccurs="1"/>
          <xsd:element name="fileNoMatch"
            type="fileNoMatchConditionType"
            minOccurs="1"
            maxOccurs="1"/>
          <xsd:element name="fileSize"
            type="fileSizeConditionType"
            minOccurs="1" maxOccurs="1"/>
          <xsd:element name="queueNotEmpty"
            type="queueNotEmptyConditionType"
            minOccurs="1" maxOccurs="1"/>
          <xsd:element name="completeGroups"
            type="completeGroupsConditionType"
            minOccurs="1" maxOccurs="1"/>
          <xsd:element name="fileSizeSame"
            type="fileSizeSameType"
            minOccurs="1" maxOccurs="1"/>
        </xsd:choice>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="fileMatchConditionType">
      <xsd:sequence>
        <xsd:element name="pattern" type="conditionPatternType"
          minOccurs="0" default="*.*"/>
        <xsd:element name="exclude" type="conditionPatternType"
          minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="fileNoMatchConditionType">
      <xsd:sequence>
        <xsd:element name="pattern" type="conditionPatternType"
          minOccurs="0" default="*.*"/>
        <xsd:element name="exclude" type="conditionPatternType"
          minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="fileSizeConditionType">
      <xsd:sequence>
        <xsd:element name="compare" type="sizeCompareType"
          minOccurs="1" default="0"/>
        <xsd:element name="pattern" type="conditionPatternType"
          minOccurs="0" default="*.*"/>
        <xsd:element name="exclude" type="conditionPatternType"
          minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="sizeCompareType">
      <xsd:simpleContent>
        <xsd:extension base="xsd:int">
          <xsd:attribute name="operator" type="sizeOperatorType"
            use="required"/>
          <xsd:attribute name="units" type="fileSizeUnitsType"
            use="required"/>
        </xsd:extension>
      </xsd:simpleContent>
    </xsd:complexType>

```

```

<xsd:simpleType name="sizeOperatorType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="="/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="fileSizeUnitsType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[bB] | [kK] [bB] | [mM] [bB] | [gG] [bB]"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="conditionPatternType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="type" type="patternTypeAttributeType"
        use="optional" default="wildcard"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:simpleType name="patternTypeAttributeType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="regex"/>
    <xsd:enumeration value="wildcard"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="conditionNameType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string"/>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="queueNotEmptyConditionType"/>

<xsd:complexType name="completeGroupsConditionType"/>

<xsd:complexType name="fileSizeSameType">
  <xsd:sequence>
    <xsd:element name="pattern" type="conditionPatternType"
      minOccurs="1" maxOccurs="1"/>
    <xsd:element name="exclude" type="conditionPatternType"
      minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="polls" type="positiveIntegerType" use="required"/>
</xsd:complexType>

<xsd:complexType name="pollIntervalType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:int">
      <xsd:attribute name="units" type="timeUnitsType"
        use="optional" default="minutes"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="batchType">
  <xsd:attribute name="maxSize" type="positiveIntegerType" use="required"/>
</xsd:complexType>

<xsd:simpleType name="timeUnitsType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="seconds"/>
    <xsd:enumeration value="minutes"/>
    <xsd:enumeration value="hours"/>
    <xsd:enumeration value="days"/>
    <xsd:enumeration value="weeks"/>
    <xsd:enumeration value="months"/>
    <xsd:enumeration value="years"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="monitorTasksType">
  <xsd:sequence>
    <xsd:element name="task" type="monitorTaskType"
      minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="monitorTaskType">

```

```

    <xsd:sequence>
      <xsd:element name="name" type="monitorTaskNameType"
        minOccurs="1" maxOccurs="1"/>
      <xsd:element name="description" type="xsd:string"
        minOccurs="0" maxOccurs="1"/>
      <xsd:element name="transfer" type="transferTaskType"
        minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="transferTaskType">
    <xsd:sequence>
      <xsd:element name="request" type="transferRequestType"
        minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="resourceIdType">
    <xsd:attribute name="id" type="xsd:string" use="optional"/>
  </xsd:complexType>

  <xsd:simpleType name="resourceIdAttrType">
    <xsd:restriction base="xsd:string"></xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="monitorNameType">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="^[^\%]*"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="agentNameType">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="[^%_0-9A-Z]*"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="monitorTaskNameType">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value=".*"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:complexType name="defaultVariablesType">
    <xsd:sequence>
      <xsd:element name="variable" type="variableType"
        maxOccurs="unbounded" minOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="variableType">
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="key" type="xsd:string" use="required"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:schema>

```

モニター作成メッセージの理解

モニター作成メッセージで使用されるエレメントと属性について説明します。

エレメントの説明

<monitor>

進行中のファイル転送を取り消すために必要なすべてのエレメントが含まれるグループ・エレメント。

属性	説明
バージョン	Managed File Transfer によって指定された、このエレメントのバージョンを示します。

<名前>

モニターのエージェント内で固有のモニター名。

<description>

モニターの説明 (現在は使用されていません)。

<pollInterval>

トリガー条件に照らし合わせるリソースの各検査間の時間間隔。

属性	説明
単位	ポーリング間隔の時間単位を指定します。有効な値は次のとおりです。 <ul style="list-style-type: none"> • seconds • minutes • hours • 日 • weeks • months • years

<agent>

モニターが関連付けられているエージェントの名前。

<resources>

モニター対象リソースを指定するエレメントが含まれているグループ・エレメント。

<directory>

モニター対象であるモニターのエージェント・マシン上のディレクトリーを指定する完全修飾パス。

属性	説明
recursionLevel	指定したディレクトリーに加え、モニター対象となるサブディレクトリーの数。
ID	リソースの固有 ID。

<queue>

モニター・エージェントのキュー・マネージャーでモニターするキューを指定するキュー名。

<triggerMatch>

モニター対象リソースと比較するトリガー条件を指定するエレメントが含まれているグループ・エレメント。

<conditions>

モニター対象リソースと比較する条件のタイプを指定するエレメントが含まれているグループ・エレメント。

<allOf>

含まれているすべての条件が満たされる必要があることを指定する述部。

<anyOf>

含まれている任意の条件が満たされる必要があることを指定する述部。

<condition>

モニター・トリガー条件全体に与えられる比較条件を定義します。

<名前>

条件の名前。

<resource>

条件を比較する際の基準となるリソース定義を指定します。

属性	説明
ID	リソースの固有 ID。

モニター対象のリソースがディレクトリーである場合は、以下の3つのエレメントのいずれかを条件で指定する必要があります。

- fileMatch
- fileNoMatch
- fileSize

モニター対象のリソースがキューである場合は、以下の2つのエレメントのいずれかを条件で指定する必要があります。

- queueNotEmpty
- completeGroups

<fileMatch>

ファイル名マッチング条件のグループ・エレメント。

<pattern>

ファイル名マッチング・パターンを指定します。条件を満たすには、リソース上のファイルがこのパターンと一致している必要があります。デフォルト・パターンは、*(任意のファイルが一致)です。

<fileNoMatch>

ファイル名逆マッチング条件のグループ・エレメント

<pattern>

ファイル名逆マッチング・パターンを指定します。モニター対象リソースに一致するファイルがない場合に、この条件が満たされます。デフォルト・パターンは、*(ファイルがまったくない場合に一致)です。

<fileSize>

ファイル・サイズ比較のグループ・エレメント。

<compare>

ファイル・サイズ比較を指定します。値は負以外の整数でなければなりません。

属性	説明
operator	使用する比較演算子。「>=」のみがサポートされています。
単位	ファイル・サイズ単位を指定します。以下のいずれかです。 <ul style="list-style-type: none">• B - バイト• KB - キロバイト• MB - メガバイト• GB - ギガバイト 単位の値では、大/小文字は区別されません。したがって、「mb」は「MB」としても機能します。

<pattern>

一致するファイル名パターン。デフォルトは、*(任意のファイルが一致)です。

<queueNotEmpty>

リソースがキューの場合にのみ指定できます。モニターを起動するためにはキューにメッセージが入っていないなければならない、という動作を指定します。

<completeGroups>

リソースがキューの場合にのみ指定できます。モニターを起動するためには完全に揃ったメッセージ・グループがキューに存在していなければならない、という動作を指定します。キューに存在する完全に揃ったグループごとに1つの転送タスクが実行されます。

<reply>

非同期要求の応答キューを指定するために使用されるエレメント (オプション)。

属性	説明
QMGR	キュー・マネージャー名。

<tasks>

モニター・トリガー条件が満たされた場合に呼び出すタスクを指定するエレメントが含まれているグループ・エレメント。

<task>

トリガー条件が満たされた場合にモニターが呼び出す個々のタスクを定義するグループ・エレメント。現時点で、指定できるのは1つのタスクのみです。

<名前>

タスクの名前。任意の英数字を受け入れます。

<description>

タスクの説明。任意のテキスト値を使用できます。

<transfer>

転送タスクを定義するグループ・エレメント。

<request>

タスクのタイプを定義するグループ・エレメント。これには、FileTransfer.xsd スキーマ定義から継承される次のエレメントのいずれかが含まれていなければなりません。

- [managedTransfer](#)
- [managedCall](#)

属性	説明
バージョン	Managed File Transfer から提供される、この要求のバージョン。これは、フォーマット n.mm で表されます。ここで、n はメジャー・リリース・バージョン、mm はマイナー・バージョンです。例: 1.00。

<originator>

要求の発信元を示すエレメントが含まれるグループ・エレメント。

<hostName>

ソース・ファイルが置かれているシステムのホスト名。

<userID>

ファイル転送を開始したユーザーの ID。

<mqmdUserID>

オプション。メッセージ記述子 (MQMD) に指定された IBM MQ ユーザー ID。

<job>

ジョブ情報が入ったグループ・エレメント。

<jobName>

論理ジョブ ID を指定します。

<defaultVariables>

1つ以上の variable エレメントを含むグループ・エレメント。これらの変数は、キューをモニターする際に変数置換で使用されます。変数置換について詳しくは、180 ページの『[変数置換を使用した MFT リソース・モニター・タスクのカスタマイズ](#)』を参照してください。

<variable>

key 属性によって与えられるキーに関連付けられた値を含むエレメント。

属性	説明
key	デフォルトの変数の名前。

モニター削除メッセージの理解

モニター削除メッセージで使用されるエレメントと属性について説明します。

エレメントの説明

<deleteMonitor>

モニターの停止および削除に必要なすべてのエレメントが含まれるグループ・エレメント。

属性	説明
バージョン	Managed File Transfer によって指定された、このエレメントのバージョンを示します。

<名前>

削除するモニターの名前。

<originator>

要求の発信元を示すエレメントが含まれるグループ・エレメント。

<hostName>

ソース・ファイルが置かれているシステムのホスト名。

<userID>

ファイル転送を開始したユーザーの ID。

<mqmdUserID>

オプション。メッセージ記述子 (MQMD) に指定された IBM MQ ユーザー ID。

<reply>

要求用に生成された一時応答キューの名前を示します。キューの名前は、`command.properties` 構成ファイル内のキー `dynamicQueuePrefix` によって定義されているとおりです。これが指定されていない場合には、キュー名はデフォルト値の `WMQFTE` になります。

属性	説明
QMGR	応答を受け取るために一時動的キューが生成されるコマンド・キュー・マネージャーの名前。

例

このスキーマに準拠する XML メッセージの例が、以下のモニター要求用に提供されています。

- [モニターの作成](#)
- [モニターの削除](#)

MFT モニター要求メッセージの例

エージェントによるモニターの作成または削除を要求するために、エージェントのコマンド・キューに PUT できるメッセージの例。

モニター作成要求

```
<?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>

```

モニター削除要求

```

<?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>

```

関連資料

797 ページの『[MFT モニター要求メッセージ・フォーマット](#)』

リソース・モニターは、通常はユーザーによる `fteCreateMonitor` コマンドの発行または IBM MQ Explorer・インターフェースの使用の結果、適切な XML メッセージがエージェントのコマンド・キューに到達すると作成されます。

Ping MFT エージェント要求メッセージ・フォーマット

ftePingAgent コマンドを発行するか、XML メッセージをエージェントのコマンド・キューに PUT することにより、エージェントを ping できます。ping エージェント要求の XML は、`PingAgent.xsd` スキーマに準拠する必要があります。Managed File Transfer のインストール後には、`PingAgent.xsd` スキーマ・ファイルが `MQ_INSTALLATION_PATH/mqft/samples/schema` ディレクトリーに配置されます。`PingAgent.xsd` スキーマは、同じディレクトリーにある `fteutils.xsd` をインポートします。

エージェントがアクティブの場合に、コマンド・キュー上の ping エージェント要求メッセージを受け取ると、そのエージェントは、ping エージェント要求メッセージをコマンド行に PUT したコマンドまたはアプリケーションに XML 応答メッセージを返します。エージェントからの応答メッセージは、`Reply.xsd` で定義されたフォーマットです。このフォーマットに関して詳しくは、808 ページの『[MFT エージェント応答メッセージ・フォーマット](#)』を参照してください。

スキーマ

次のスキーマは、ping エージェント要求 XML メッセージで有効なエレメントについて示します。

```
<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>
```

ping エージェント要求メッセージについて

ping エージェント要求メッセージで使用するエレメントと属性を以下のリストにまとめます。

<pingAgent>

ping エージェント要求を指定するために必要なすべてのエレメントが含まれるグループ・エレメント。

<originator>

ping 要求の送信元を指定するために必要なすべてのエレメントが含まれるグループ・エレメント。

<hostName>

要求の送信元であるマシンのホスト名。

<userID>

要求の送信元のユーザー名。

<mqmdUserID>

要求の送信元の MQMD ユーザー名。

<agent>

ping 対象のエージェント。

属性	説明
エージェント	必須。エージェントの名前。
キュー・マネージャー	オプション。エージェントの接続先キュー・マネージャー。

<reply>

応答メッセージの送信先エージェントのキューの名前。

属性	説明
QMGR	必須。応答キューがあるキュー・マネージャーの名前。

例

この例は、エージェント AGENT_JUPITER に送信された ping エージェント・メッセージを示しています。AGENT_JUPITER がアクティブで、エージェント要求を処理できる場合、このエージェントは応答メッセージを QM_JUPITER 上にあるキュー WMQFTE.4D400F8B20003708 に送信します。

```
<?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 エージェント応答メッセージ・フォーマット

エージェントがエージェント・コマンド・キューで XML メッセージを受け取ったとき、応答が必要な場合は、オリジナル・メッセージで定義された応答キューにエージェントが XML 応答メッセージを送信します。応答 XML は Reply.xsd スキーマに準拠しています。Reply.xsd スキーマ文書は MQ_INSTALLATION_PATH/mqft/samples/schema ディレクトリーにあります。Reply.xsd スキーマは、同じディレクトリーにある fteutils.xsd をインポートします。

スキーマ

次のスキーマは、応答 XML メッセージで有効なエレメントについて示します。

```

<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>

```

応答メッセージについて

応答メッセージで使用するエレメントと属性を以下のリストにまとめます。

<reply>

応答情報を指定するエレメントを含むエレメント。

属性	説明
ID	応答の ID。
バージョン	応答メッセージ・フォーマットのバージョン。
detailedReplyMessagesDisabled	エージェントが詳細応答を無効にした (enableDetailedReplyMessages エージェント・プロパティーが false に設定されている) という通知。

<transferSet>

転送のために要求されるファイルの転送結果情報を指定します。詳しくは、748 ページの『ファイル転送ログ・メッセージ・フォーマット』を参照してください。

<status>

エージェントが実行を要求された操作の状況。

属性	説明
resultCode	エージェントが実行した操作から返された結果コード。

非許可ログ・メッセージについて

非許可メッセージで使用されるエレメントと属性について説明します。

<notAuthorized>

単一のユーザー権限検査の失敗を説明するグループ・エレメント。

属性	説明
バージョン	Managed File Transfer で示されるこのエレメントのバージョンを示します。
ID	許可されなかった要求の固有 ID。

<originator>

要求の発信元を示すエレメントが含まれるグループ・エレメント。

<authority>

要求されたアクションの実行のためにユーザーが必要とする Managed File Transfer アクセス権限のレベルを示します。

<mqmdUserID>

メッセージ記述子 (MQMD) に指定された IBM MQ ユーザー ID

<action>

<notAuthorized> エレメントの ID 属性と一致する要求の許可状況を示します。

属性	説明
time	ログ項目がパブリッシュされた日時を示します (日時形式)。

<status>

結果コードと補足メッセージ。

MFT 資格情報ファイル・フォーマット

MQMFTCredentials.xml ファイルには、ユーザー ID とパスワードに関する重要な情報が含まれています。MQMFTCredentials.xml ファイル内のエレメントは MQMFTCredentials.xsd スキーマに準拠する必要があります。資格情報ファイルのセキュリティはユーザーの責任です。

V 9.0.0.2 **V 9.0.3** IBM MQ 9.0.0 Fix Pack 2 および IBM MQ 9.0.3 以降では、関連するユーザーの MFT 資格情報ファイル MQMFTCredentials.xml に新しいパラメーター **useMQCSPAuthentication** を追加し、それを true に設定することにより、デフォルトの互換モードを無効にし、Managed File Transfer エージェントの MQCSP 認証を有効にすることができます。詳しくは、『[MQCSP 認証モードの有効化](#)』を参照してください。

z/OS **z/OS** IBM MQ 8.0 以降では、MQMFTCredentials.xsd ファイルを z/OS の PDSE メンバーにすることもできます。

IBM WebSphere MQ 7.5 より前のバージョンでは、MQMFTCredentials.xml ファイルに含まれている情報は、別のプロパティファイルに保持されていました。

MQMFTCredentials.xml ファイルは MQMFTCredentials.xsd スキーマに準拠する必要があります。MQMFTCredentials.xml スキーマ文書は、Managed File Transfer インストール済み環境の `MQ_INSTALLATION_PATH/mqft/samples/schema` ディレクトリーにあります。

スキーマ

次のスキーマは、MQMFTCredentials.xml ファイル内で有効なエレメントについて示します。

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

Licensed Materials - Property of IBM

5724-H72

Copyright IBM Corp. 2012, 2023. All Rights Reserved.

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

```
@end_non_restricted_prolog@
-->

<!--
This schema defines the format of an MQMFTCredentials file. Files of this type
store credential information for agent and logger processes. They can contain
user names and passwords either in clear text or which have been obfuscated
using the fteObfuscate command.
-->

<!-- Example mqmftCredentials.xml file:
<?xml version="1.0" encoding="UTF-8"?>
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/
MQMFTCredentials"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/MQMFTCredentials MQMFTCredentials.xsd">

  <tns:logger name="LOG1" user="user1" password="passw0rd"/>
  <tns:logger name="ORACLE" userCipher="kj2h3dfkgf" passwordCipher="1a3n67eae" />
  <tns:file path="/home/emma/trust.jks" password="passw0rd"/>
  <tns:file path="/var/tmp/keystore.jks" passwordCipher="e71vKCg2pf" />

  <tns:qmgr name="QM_COORD" user="tim" mqUserId="user1" mqPassword="passw0rd"/>
  <tns:qmgr name="QM_COORD" user="tom" mqUserId="user1" mqPasswordCipher="e71vKCg2pf" />
  V 9.0.3 <tns:qmgr name="QM_COORD" user="ernest" mqUserId="ernest"
mqPassword="AveryL0ngPassw0rd2135" useMQCSPAAuthentication="true" />
</tns:mqmftCredentials>
-->

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

  <element name="mqmftCredentials" type="tns:mqmftCredentialsType"/>

  <complexType name="mqmftCredentialsType">
    <sequence>
      <choice minOccurs="0" maxOccurs="unbounded">
        <element name="logger" type="tns:loggerType"/>
        <element name="file" type="tns:fileType"/>
        <element name="qmgr" type="tns:mqUserPassType"/>
      </choice>
    </sequence>
  </complexType>

  <complexType name="loggerType">
    <attribute name="name" type="string" use="required"/>
    <attribute name="user" type="string" use="optional"/>
    <attribute name="userCipher" type="string" use="optional"/>
    <attribute name="password" type="string" use="optional"/>
    <attribute name="passwordCipher" type="string" use="optional"/>
  </complexType>

  <complexType name="fileType">
    <attribute name="path" type="string" use="required"/>
    <attribute name="password" type="string" use="optional"/>
    <attribute name="passwordCipher" type="string" use="optional"/>
  </complexType>

  <!-- Example XML:

  <tns:qmgr name="QM_COORD" user="tim" mqUserId="user1" mqPassword="passw0rd"/>
  <tns:qmgr name="QM_COORD" user="tom" mqUserIdCipher="xh5U7812x"
mqPasswordCipher="e71vKCg2pf"/>
  <tns:qmgr name="QM_COORD" mqUserId="defaultUser" mqPassword="passw0rd"/>
  V 9.0.3 <tns:qmgr name="QM_COORD" user="ernest" mqUserId="ernest"
mqPassword="AveryL0ngPassw0rd2135" useMQCSPAAuthentication="true"/>
-->

  <complexType name="mqUserPassType">
```

```

<attribute name="name" type="string" use="required"/>
<attribute name="user" type="string" use="optional"/>
<attribute name="mqUserId" type="string" use="optional"/>
<attribute name="mqUserIdCipher" type="string" use="optional"/>
<attribute name="mqPassword" type="string" use="optional"/>
<attribute name="mqPasswordCipher" type="string" use="optional"/>
V 9.0.3 <attribute name="useMQCSPAAuthentication" type="boolean" use="optional"/>
</complexType>
</schema>

```

MQMFTCredentials.xml ファイルについて

MQMFTCredentials.xml ファイルで使用するエレメントと属性を以下のリストにまとめます。

<mqmftCredentials>

XML 文書のルート・エレメント。

<file>

転送されるファイル。

属性	説明
path	アクセスされるキーまたはトラストストア・ファイルのパス。
パスワード	ファイルにアクセスするためのパスワード。

<logger>

ロギング・アクティビティを担当するロガー。

属性	説明
名前	ロガーの名前。
user	ロガーがそのデータベースに接続するために使用するユーザー名。
パスワード	ロガーがそのデータベースに接続するために使用するパスワード。

<qmgr>

IBM MQ キュー・マネージャー接続。

属性	説明
名前	関連付けられた IBM MQ キュー・マネージャーの名前。
user	オプション: 接続を要求しているユーザーの名前。
mqUserId または mqUserIdCipher	IBM MQ キュー・マネージャーに提供する平文のユーザー ID (mqUserId)、または難読化テキストのユーザー ID (mqUserIdCipher)。
mqPassword または mqPasswordCipher	IBM MQ キュー・マネージャーに提供する平文のパスワード (mqPassword)、または難読化テキストのパスワード (mqPasswordCipher)。

注: MQMFTCredentials.xml ファイルには機密情報が含まれる可能性があるため、作成する際はファイル・アクセス許可を必ず確認してください。サンドボックスを使用する際、それが除外されるよう設定してください。サンドボックスについて詳しくは、45 ページの『MFT エージェント・サンドボックスの処理』を参照してください。

関連概念

70 ページの『z/OS での MQMFTCredentials.xml の構成』

Managed File Transfer (MFT) がセキュリティーを有効にして構成されている場合、接続認証では、ユーザー ID とパスワードの資格情報を提供するために、すべての MFT エージェント、およびキュー・マネージャーに接続するコマンドが必要になります。

関連資料

42 ページの『MFT と IBM MQ の接続認証』

接続認証では、指定されたユーザー ID とパスワードを使用してアプリケーションを認証するようキュー・マネージャーを構成できます。関連付けられたキュー・マネージャーのセキュリティーが使用可能に設定されており、資格情報の詳細 (ユーザー ID とパスワード) が必要な場合、キュー・マネージャーと正常に接続するには、その前に接続認証フィーチャーを使用可能にしておく必要があります。接続認証は互換モードでも、MQCSP 認証モードでも実行できます。

494 ページの『fteObfuscate: 機密データの暗号化』

fteObfuscate コマンドは、資格情報ファイル内の機密データを暗号化します。これにより、他のユーザーが資格情報ファイルにアクセスした際に、内容が読み取られるのを防ぐことができます。

プロトコル・ブリッジの資格情報ファイルのフォーマット

Managed File Transfer Agent の構成ディレクトリーにある ProtocolBridgeCredentials.xml ファイルは、プロトコル・ブリッジ・エージェントがプロトコル・サーバーでの許可に使用するユーザー名および資格情報を定義します。

ProtocolBridgeCredentials.xml ファイルは ProtocolBridgeCredentials.xsd スキーマに準拠する必要があります。ProtocolBridgeCredentials.xsd スキーマ文書は、MQMFT インストール済み環境の `MQ_INSTALLATION_PATH/mqft/samples/schema` ディレクトリーにあります。

ProtocolBridgeCredentials.xml ファイルは、ユーザーが手動で作成する必要があります。このファイルは **fteCreateBridgeAgent** コマンドでは作成されなくなりました。MQMFT インストール済み環境の `MQ_INSTALLATION_PATH/mqft/samples` ディレクトリーにサンプル・ファイルがあります。

IBM WebSphere MQ 7.5 では、指定されたエージェントの `<server>` または `<serverHost>` エレメントを含む新しい `<agent>` エレメントが導入されました。

ProtocolBridgeCredentials.xml ファイルはエージェントが定期的に再ロードするので、このファイルに有効な変更を加えるとエージェントの動作に反映されます。デフォルトの再ロード間隔は 30 秒です。この間隔は、`agent.properties` ファイルの `xmlConfigReloadInterval` エージェント・プロパティを指定することにより変更できます。

スキーマ- 7.5 以降

以下のスキーマは、IBM MQ 8 の ProtocolBridgeCredentials.xml ファイルで有効なエレメントを記述します。

```
<schema targetNamespace="http://wmqfte.ibm.com/ProtocolBridgeCredentials" elementFormDefault="qualified"
  xmlns="https://www.w3.org/2001/XMLSchema" xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeCredentials">
  <!--
  <?xml version="1.0" encoding="UTF-8"?>
  <tns:credentials xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeCredentials"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeCredentials
  ProtocolBridgeCredentials.xsd">
    <tns:agent name="agent1">
      <tns:serverHost name="myserver">
        <tns:user name="fred" serverPassword="pwd" serverUserId="bill"/>
        <tns:user name="jane" serverUserId="june" hostKey="1F:2e:f3">
          <tns:privateKey associationName="test" keyPassword="pwd2">
            .... private key ...
          </tns:privateKey>
        </tns:user>
      </tns:serverHost>
    </tns:agent>

    <tns:agent name="agent2">
      <tns:server name="server*" pattern="wildcard">
        <tns:user name="fred" serverPassword="pwd" serverUserId="bill"/>
        <tns:user name="jane" serverUserId="june" hostKey="1F:2e:f3">
          <tns:privateKey associationName="test" keyPassword="pwd2">
            .... private key ...
          </tns:privateKey>
        </tns:user>
      </tns:server>
    </tns:agent>
```

```

    <tns:agent name="agent3">
      <tns:serverHost name="ftpsServer"
        keyStorePassword="keypass"
        trustStorePassword="trustpass">
        <tns:user name="fred" serverPassword="pwd" serverUserId="bill"/>
      </tns:serverHost>
    </tns:agent>
  </tns:credentials>
-->
<element name="credentials" type="tns:credentialsType"/>
<complexType name="credentialsType">
  <sequence>
    <element name="agent" type="tns:agentType" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
<complexType name="agentType">
  <choice minOccurs="0" maxOccurs="1">
    <element name="serverHost" type="tns:serverHostType" minOccurs="0" maxOccurs="unbounded"/>
    <element name="server" type="tns:serverType" minOccurs="0" maxOccurs="unbounded"/>
  </choice>
  <attribute name="name" type="string" use="required"/>
</complexType>
<complexType name="serverHostType">
  <sequence>
    <element ref="tns:user" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="name" type="string" use="required"/>
  <attribute name="keyStorePassword" type="string" use="optional"/>
  <attribute name="keyStorePasswordCipher" type="string" use="optional"/>
  <attribute name="trustStorePassword" type="string" use="optional"/>
  <attribute name="trustStorePasswordCipher" type="string" use="optional"/>
</complexType>
<complexType name="serverType">
  <sequence>
    <element ref="tns:user" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="name" type="string" use="required"/>
  <attribute name="pattern" type="tns:patternType" use="optional"/>
  <attribute name="keyStorePassword" type="string" use="optional"/>
  <attribute name="keyStorePasswordCipher" type="string" use="optional"/>
  <attribute name="trustStorePassword" type="string" use="optional"/>
  <attribute name="trustStorePasswordCipher" type="string" use="optional"/>
</complexType>
<element name="user" type="tns:userType"/>
<complexType name="userType">
  <sequence>
    <element ref="tns:privateKey" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="name" type="string" use="required"/>
  <attribute name="serverUserId" type="string" use="optional"/>
  <attribute name="serverUserIdCipher" type="string" use="optional"/>
  <attribute name="serverPassword" type="string" use="optional"/>
  <attribute name="serverPasswordCipher" type="string" use="optional"/>
  <attribute name="hostKey" use="optional">
    <simpleType>
      <restriction base="string">
        <pattern
          value="([a-fA-F0-9]){2}(:([a-fA-F0-9]){2})*">
        </pattern>
      </restriction>
    </simpleType>
  </attribute>
</complexType>
<element name="privateKey" type="tns:privateKeyType"/>
<complexType name="privateKeyType">
  <simpleContent>
    <extension base="string">
      <attribute name="keyPassword" type="string" use="optional"/>
      <attribute name="keyPasswordCipher" type="string" use="optional"/>
      <attribute name="associationName" type="string" use="required"/>
    </extension>
  </simpleContent>
</complexType>

```

```

    </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>

```

ProtocolBridgeCredentials.xml ファイルについて

ProtocolBridgeCredentials.xml ファイルで使用するエレメントと属性を以下のリストにまとめます。

<credentials>

プロトコル・サーバーに接続するためにプロトコル・ブリッジ・エージェントにより使用される資格情報を記述するエレメントを含む、グループ・エレメント。

<agent>

指定されたエージェントの <server> または <serverHost> 定義が含まれているエレメント。

<server>

プロトコル・ブリッジの接続先のプロトコル・サーバー。

<server> エレメントは、7.0.4 以前ではサポートされません。

属性	説明
名前	プロトコル・サーバーの名前。
pattern	ワイルドカードまたは正規表現を使用してプロトコル・サーバー名のパターンを指定している場合は、wildcard または regex を使用します。
trustStorePassword または trustStorePasswordCipher	<server> エレメントが FTPS サーバーを参照する場合は必須です。トラスストアへのアクセスに使用するパスワード。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。
keyStorePassword または keyStorePasswordCipher	オプション。鍵ストアへのアクセスに使用するパスワード。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。

<serverHost>

プロトコル・ブリッジの接続先のプロトコル・サーバーのホスト名。

ProtocolBridgeCredentials.xml ファイルには、<serverHost> エレメントまたは <server> エレメントを含めることができますが、この2つのタイプを組み合わせることはできません。<serverHost> を使用すると、名前はプロトコル・サーバーのホスト名とマッチングされます。<server> を使用すると、名前はプロトコル・サーバーの名前 (ProtocolBridgeProperties.xml ファイル内で定義されている) とマッチングされます。

属性	説明
名前	プロトコル・サーバーのホスト名または IP アドレス。
trustStorePassword または trustStorePasswordCipher	<serverHost> エレメントが FTPS サーバーを参照する場合は必須です。トラスストアへのアクセスに使用するパスワード。

属性	説明
	fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。
keyStorePassword または keyStorePasswordCipher	オプション。鍵ストアへのアクセスに使用するパスワード。このプロパティは、keyStore 属性を設定していない場合はオプションで、設定している場合は必須です。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。

<user>

Managed File Transfer のユーザー名からプロトコル・サーバーのユーザー名へのユーザー・マッピング。

属性	説明
名前	Managed File Transfer で使用されるユーザー名。
serverUserId または serverUserIdCipher	プロトコル・サーバーで使用されるユーザー名。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。
serverPassword または serverPasswordCipher	プロトコル・サーバーで使用されるユーザー名のパスワード。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。
hostKey	サーバーのホストの SSH 指紋。

<privateKey>

ユーザーの秘密鍵。

属性	説明
keyPassword または keyStorePasswordCipher	秘密鍵のパスワード。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。
associationName	トレースとロギングに使用される名前。

プロトコル・ブリッジ・プロパティ・ファイルのフォーマット

エージェント構成ディレクトリー内の ProtocolBridgeProperties.xml ファイルは、プロトコル・ファイル・サーバーのプロパティを定義します。

ProtocolBridgeProperties.xml ファイルは ProtocolBridgeProperties.xsd スキーマに準拠する必要があります。ProtocolBridgeProperties.xsd スキーマ文書は、Managed File Transfer インストール済み環境の MQ_INSTALLATION_PATH/mqft/samples/schema ディレクトリーにあります。テンプレート・ファイル ProtocolBridgeProperties.xml は、**fteCreateBridgeAgent** コマンドにより、エージェントの構成ディレクトリーに作成されます。

ProtocolBridgeProperties.xml ファイルはエージェントが定期的に再ロードするので、このファイルに有効な変更を加えるとエージェントの動作に反映されます。デフォルトの再ロード間隔は 30 秒です。この間隔は、agent.properties ファイルの xmlConfigReloadInterval エージェント・プロパティを指定することにより変更できます。

スキーマ

以下のスキーマは ProtocolBridgeProperties.xml ファイルを記述しています。

注: maxReconnectRetry 属性および reconnectWaitPeriod 属性は、IBM WebSphere MQ 7.5 または IBM WebSphere MQ File Transfer Edition 7.0.2 以降ではサポートされません。

```
<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>
    <sequence>
      <element name="limits" type="tns:generalLimitsType" minOccurs="0" maxOccurs="1" />
    </sequence>
  </sequence>
  <attributeGroup ref="tns:ftpServerAttributes"/>
</complexType>
```

```

<!--
  A container for all the information about an SFTP server
-->
<complexType name="sftpServerType">
  <sequence>
    <element name="limits" type="tns:sftpLimitsType" minOccurs="0" maxOccurs="1" />
  </sequence>
  <attributeGroup ref="tns:sftpServerAttributes"/>
</complexType>

<!--
  A container for all the information about a FTPS server
-->
<complexType name="ftpsServerType">
  <sequence>
    <element name="limits" type="tns:generalLimitsType" minOccurs="0" maxOccurs="1" />
  </sequence>
  <attributeGroup ref="tns:ftpsServerAttributes"/>
</complexType>

<!--
  A container for all the information about a SFG FTPS server
-->
<complexType name="ftpsSfgServerType">
  <sequence>
    <element name="limits" type="tns:generalLimitsType" minOccurs="0" maxOccurs="1" />
  </sequence>
  <attributeGroup ref="tns:ftpsServerAttributes"/>
</complexType>

<!--
  Attributes common to all server types
-->
<attributeGroup name="generalServerAttributes">
  <attribute name="name" type="tns:serverNameType" use="required"/>
  <attribute name="host" type="string" use="required"/>
  <attribute name="port" type="nonNegativeInteger" use="optional"/>
  <attribute name="platform" type="tns:platformType" use="required"/>
  <attribute name="fileEncoding" type="string" use="required"/>
  <attribute name="limitedWrite" type="boolean" use="optional"/>
  <attribute name="controlEncoding" type="string" use="optional"/>
</attributeGroup>

<!--
  Attributes common to ftp and ftps server types
-->
<attributeGroup name="ftpServerAttributes">
  <attributeGroup ref="tns:generalServerAttributes"/>
  <attribute name="timeZone" type="string" use="required"/>
  <attribute name="locale" type="tns:localeType" use="required"/>
  <attribute name="listFormat" type="tns:listFormatType" use="optional"/>
  <attribute name="listFileRecentDateFormat" type="tns:dateFormatType" use="optional"/>
  <attribute name="listFileOldDateFormat" type="tns:dateFormatType" use="optional"/>
  <attribute name="monthShortNames" type="tns:monthShortNamesType" use="optional"/>
</attributeGroup>

<!--
  Attributes common to ftps server types
-->
<attributeGroup name="ftpsServerAttributes">
  <attributeGroup ref="tns:ftpServerAttributes"/>
  <attribute name="ftpsType" type="tns:ftpsTypeType" use="optional"/>
  <attribute name="trustStore" type="string" use="required"/>
  <attribute name="trustStoreType" type="string" use="optional"/>
  <attribute name="keyStore" type="string" use="optional"/>
  <attribute name="keyStoreType" type="string" use="optional"/>
  <attribute name="ccc" type="boolean" use="optional"/>
  <attribute name="protFirst" type="boolean" use="optional"/>
  <attribute name="auth" type="string" use="optional"/>
  <attribute name="connectTimeout" type="nonNegativeInteger" use="optional"/>
</attributeGroup>

<!--
  A container for limit-type attributes for a server. Limit parameters
  are optional, and if not specified a system default will be used.
-->
<complexType name="generalLimitsType">
  <attributeGroup ref="tns:generalLimitAttributes"/>
</complexType>

```

```

<complexType name="sftpLimitsType">
  <attributeGroup ref="tns:generalLimitAttributes"/>
  <attribute name="connectionTimeout" type="nonNegativeInteger" use="optional"/>
</complexType>

<!--
  Attributes for limits common to all server types
-->
<attributeGroup name="generalLimitAttributes">
  <attribute name="maxListFileNames" type="positiveInteger" use="optional"/>
  <attribute name="maxListDirectoryLevels" type="nonNegativeInteger" use="optional"/>
  <attribute name="maxReconnectRetry" type="nonNegativeInteger" use="optional"/>
  <attribute name="reconnectWaitPeriod" type="nonNegativeInteger" use="optional"/>
  <attribute name="maxSessions" type="positiveInteger" use="optional"/>
  <attribute name="socketTimeout" type="nonNegativeInteger" use="optional"/>
</attributeGroup>

<!--
  The type for matching valid server names. Server names must be at least 2 characters in length
and
  are limited to alphanumeric characters and the following characters: ".", "_", "/" and "%".
-->
<simpleType name="serverNameType">
  <restriction base="string">
    <pattern value="[0-9a-zA-Z\._/%]{2,}" />
  </restriction>
</simpleType>

<!--
  The types of platform supported.
-->
<simpleType name="platformType">
  <restriction base="string">
  </restriction>
</simpleType>

<!--
  The type for matching a locale specification.
-->
<simpleType name="localeType">
  <restriction base="string">
    <pattern value="(.)[-_](.)/" />
  </restriction>
</simpleType>

<!--
  The types of list format supported (for FTP servers).
-->
<simpleType name="listFormatType">
  <restriction base="string">
  </restriction>
</simpleType>

<!--
  Date format for FTP client directory listing on an FTP server. This is
  the format to be passed to methods setDefaultDateFormatStr and
  setRecentDateFormatStr for Java class:
  org.apache.commons.net.ftp.FTPClientConfig
-->
<simpleType name="dateFormatType">
  <restriction base="string">
  </restriction>
</simpleType>

<!--
  A list of language-defined short month names can be specified. These are
  used for translating the directory listing received from the FTP server.
  The format is a string of three character month names separated by "|"
-->
<simpleType name="monthShortNamesType">
  <restriction base="string">
    <pattern value="(...\|){11}(...)" />
  </restriction>
</simpleType>

<!--
  The enumerations of the allowed FTPS types: "implicit" & "explicit"
  If not specified the default is "explicit"
-->
<simpleType name="ftpsTypeType">
  <restriction base="string">

```

```

        <enumeration value="explicit"/>
        <enumeration value="implicit"/>
    </restriction>
</simpleType>

<!--
    Attribute Group for SFTP Servers
-->
<attributeGroup name="sftpServerAttributes">
    <attributeGroup ref="tns:generalServerAttributes"/>
    <attribute name="cipherList" type="string" use="optional"/>
</attributeGroup>
</schema>

```

ProtocolBridgeProperties.xml ファイルについて

ProtocolBridgeProperties.xml ファイルで使用するエレメントと属性を以下のリストにまとめます。

<serverProperties>

XML 文書のルート・エレメント

<credentialsFile>

資格情報が含まれているファイルのパス。IBM WebSphere MQ 7.5 以降の場合は、このプロパティの値に環境変数を含めることができます。詳しくは、[535 ページの『MFT プロパティでの環境変数の使用』](#)を参照してください。

<defaultServer>

ファイル転送用のデフォルト・サーバーの働きをするプロトコル・ファイル・サーバー

<ftpServer>

FTP ファイル・サーバー

<sftpServer>

SFTP ファイル・サーバー

<ftpsServer>

FTPS ファイル・サーバー

すべてのタイプのプロトコル・ファイル・サーバーに適用される汎用サーバー属性:

属性	説明
名前	<p>必須。プロトコル・ファイル・サーバーの名前。プロトコル・サーバー名は、2 文字以上の長さでなければならず、大/小文字の区別はなく、英数字と以下の文字に制限されます。</p> <ul style="list-style-type: none"> • ピリオド (.) • 下線 (_) • スラッシュ (/) • パーセント記号 (%)
host	<p>必須。ファイルの送信先または送信元となるプロトコル・ファイル・サーバーのホスト名または IP アドレス。</p>
port	<p>オプション。ファイルの送信先または送信元となるプロトコル・ファイル・サーバーのポート番号。</p>
platform	<p>必須。ファイルの送信先または送信元となるプロトコル・ファイル・サーバーのプラットフォーム。UNIX または WINDOWS のいずれかを指定します。このプロパティは、FTP、FTPS、または SFTP サーバー上でのパスの入力方法に従って設定します。例えば、Windows で実行している FTP サーバーにログインするときに UNIX スタイルのパス (スラッシュを使用するパス) を入力しなければならない場合は、この値を WINDOWS ではなく UNIX に設定します。Windows で実行されるサーバーの多くは、UNIX スタイルのファイル・システムに対応しています。</p>

属性	説明
fileEncoding	必須。ファイル・サーバーで使用されている文字エンコード方式を定義します。このプロパティは、テキスト・モードでファイルを転送する場合に使用され、プラットフォーム間をファイルが移動する際に、エンコード手順が正しく変更されるようにします。例: UTF-8。
limitedWrite	オプション。ファイル・サーバーに書き込みを行う際のデフォルト・モードでは、一時ファイルを作成し、転送が完了した後にそのファイルをリネームします。書き込み専用として構成されたファイル・サーバーの場合、ファイルは、最終的な名前をそのまま使用して作成されます。このプロパティの値は、true または false のいずれかになります。デフォルトは false です。
controlEncoding	オプション。プロトコル・ファイル・サーバーに送信される制御メッセージの制御エンコード値。このプロパティは、使用されるファイル名のエンコードに影響し、プロトコル・ファイル・サーバーの制御エンコードとの互換性が必要です。デフォルトは UTF-8 です。

FTP サーバーと FTPS サーバーのみに適用される汎用属性:

属性	説明
timeZone	必須。ファイルの送信先または送信元となるプロトコル・ファイル・サーバーのタイム・ゾーン。例: America/New_York または Asia/Tokyo。
ロケール (locale)	必須。ファイルの送信先または送信元となるプロトコル・ファイル・サーバーで使用される言語。例: en_US または ja_JP。
listFormat	オプション。プロトコル・ファイル・サーバーから返される情報をリストするファイルの形式を定義するリスト形式。Windows または UNIX を使用します。デフォルトは UNIX です。
listFileRecentDateFormat	オプション。FTP サーバー上の FTP クライアント・ディレクトリー・リストの最新の日付形式 (1 年未満)。この属性と listFileOldDateFormat 属性を使用することで、プロトコル・ファイル・サーバーから返される、予期される日付形式を再定義することができます。デフォルトは、プロトコル・ファイル・サーバーで定義されている値です。
listFileOldDateFormat	オプション。FTP サーバー上の FTP クライアント・ディレクトリー・リストの古い日付形式 (1 年を超えている)。この属性と listFileRecentDateFormat 属性を使用することで、プロトコル・ファイル・サーバーから返される、予期される日付形式を再定義することができます。デフォルトは、プロトコル・ファイル・サーバーで定義されている値です。
monthShortNames	オプション。プロトコル・ファイル・サーバーから返される日付情報のデコードに使用される月名の置換リスト。このプロパティは、デフォルト・ロケールの月の値をオーバーライドする、12 個のコンマ区切りの名前のリストで構成されます。デフォルトは、プロトコル・ファイル・サーバーで定義されている値です。

FTP サーバーのみに適用される汎用属性:

属性	説明
passiveMode	オプション。FTP サーバーへの接続がパッシブかアクティブかを制御します。このプロパティの値を false に設定すると、接続はアクティブになります。この値を true に設定すると、接続はパッシブになります。デフォルトは false です。

FTPS サーバーのみに適用される汎用属性:

属性	説明
ftpstype	オプション。使用される FTPS プロトコルの形式が明示的か暗黙的かを指定します。デフォルトは explicit です。
trustStore	必須。FTPS サーバーによって提示される証明書が信頼できるかどうかの判別使用するトラストストアの場所。
trustStoreType	オプション。トラストストア・ファイルの形式。デフォルトは JKS です。
keyStore	オプション。FTPS サーバーからユーザー確認が行われた場合に、証明書の情報を提供するために使用する鍵ストアの場所。デフォルトでは、プロトコル・ブリッジは、クライアントの認証を必要とするように構成された FTPS サーバーに接続できません。
keyStoreType	オプション。鍵ストア・ファイルの形式。デフォルトは JKS です。
ccc	オプション。認証の完了後に保護されていない (暗号化されていない) コマンド・チャンネルを使用するかどうかを選択します。デフォルト値は false で、この場合は FTPS セッションの期間全体でコマンド・チャンネルが暗号化されたままになります。この属性は、ftpstype が explicit に設定されている場合のみ適用できます。
protFirst	オプション。USER/PASS コマンドを FTPS サーバーに発行するのを PBSZ/PROT コマンドの前にするか後にするかを指定します。デフォルト値は false です。この場合、USER/PASS コマンドが最初に送信され、その後に PBSZ/PROT コマンドが送信されます。この属性は、ftpstype が explicit に設定されている場合のみ適用できます。
auth	オプション。AUTH コマンドの一部として指定されるプロトコルを指定します。まず指定されたプロトコルを試行してから、デフォルトでは、FTPS サーバーが 504 応答コードで拒否しなくなるまで、TLS、SSL、TLS-C、または TLS-P を試行します。この属性は、ftpstype が explicit に設定されている場合のみ適用できます。

<limits>

すべてのタイプのサーバーに共通の属性や、サーバーのタイプに固有の属性のコンテナー・エレメント。以下の属性があります。

すべてのタイプのプロトコル・ファイル・サーバーに適用する汎用制限属性:

属性	説明
maxListFileNames	オプション。プロトコル・ファイル・サーバー上のディレクトリーでファイル名をスキャンする際に収集される名前の最大数。デフォルトは 999999999 です。
maxListDirectoryLevels	オプション。プロトコル・サーバー上でファイル名を繰り返しスキャンするディレクトリー・レベルの最大数。デフォルトは 1000 です。
maxReconnectRetry (この属性は非推奨になっています。)	非推奨。この属性は、IBM WebSphere MQ 7.5 または IBM WebSphere MQ File Transfer Edition 7.0.2 以降ではサポートされません。 オプション。プロトコル・ブリッジ・エージェントが試行を停止する前に、プロトコル・サーバーが再接続を試行する最大回数。デフォルトは 2 です。
reconnectWaitPeriod (この属性は非推奨になっています。)	非推奨。この属性は、IBM WebSphere MQ 7.5 または IBM WebSphere MQ File Transfer Edition 7.0.2 以降ではサポートされません。 オプション。再接続を試行する前に待機する時間 (秒単位)。デフォルトは 10 秒です。

属性	説明
maxSessions	オプション。プロトコル・サーバーに関するセッションの最大数。この数値は、プロトコル・ブリッジ・エージェントのソース転送と宛先転送の最大数の合計以上でなければなりません。デフォルトは、エージェント・プロパティ maxSourceTransfers、maxDestinationTransfers、および maxCommandHandlerThreads の値の合計に 1 を加えた値です。これらの 3 つのプロパティがデフォルト値の 25、25、および 5 を使用する場合、maxSessions のデフォルトは 56 になります。
socketTimeout	オプション。ソケット・タイムアウト (秒単位)。この属性の値は、ファイルのストリーミング中に使用されます。デフォルトは 30 秒です。

SFTP サーバーのみに適用される制限属性:

属性	説明
connectionTimeout	オプション。接続要求に対するプロトコル・ファイル・サーバーからの応答を待機する時間 (秒単位)。タイムアウトは、プロトコル・ファイル・サーバーが使用可能でないことを示します。デフォルト値は 30 秒です。
cipherList	<p>オプション。プロトコル・ブリッジ・エージェントと SFTP サーバーの間で通信するために使用される暗号のコンマ区切りリストを指定します。暗号は、このリストで指定された順序で呼び出されます。暗号は、使用前にサーバーとクライアントで使用可能になっていなければなりません。</p> <p>プロトコル・ブリッジ・エージェントがサポートする暗号は、以下のとおりです。</p> <ul style="list-style-type: none"> • blowfish-cbc • 3des-cbc • aes128-cbc • aes192-cbc • aes256-cbc • aes128-ctr • aes192-ctr • aes256-ctr • 3des-ctr • arcfour • arcfour128 • arcfour256 <p>デフォルトでは、プロトコル・ブリッジ・エージェントによって使用される暗号のリストは、aes128-cbc, aes192-cbc, aes256-cbc です。</p>

Connect:Direct 資格情報ファイル・フォーマット

Managed File Transfer Agent 構成ディレクトリーの ConnectDirectCredentials.xml ファイルは、Connect:Direct エージェントが Connect:Direct ノードに対して自身を許可するのに使用するユーザー名および資格情報を定義します。

ConnectDirectCredentials.xml ファイルは ConnectDirectCredentials.xsd スキーマに準拠する必要があります。ConnectDirectCredentials.xsd スキーマ文書は、MQMFT インストール済み環境の MQ_INSTALLATION_PATH/mqft/samples/schema ディレクトリーにあります。サンプルの ConnectDirectCredentials.xml ファイルは、MQMFT インストール済み環境の MQ_INSTALLATION_PATH/mqft/samples/credentials ディレクトリーにあります。

ファイル `ConnectDirectCredentials.xml` はエージェントによって定期的に再ロードされ、ファイルに有効な変更が加えられていれば、それはエージェントの動作に反映されます。デフォルトの再ロード間隔は 30 秒です。この間隔は、`agent.properties` ファイルでエージェント・プロパティ `xmlConfigReloadInterval` を指定することによって変更できます。

スキーマ

次のスキーマは、`ConnectDirectCredentials.xml` ファイル内で有効なエレメントについて示します。

```
<?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"/>
    </complexType>
```

```

<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>

```

ConnectDirectCredentials.xml ファイルについて

ConnectDirectCredentials.xml ファイルで使用するエレメントと属性を以下のリストにまとめます。

<credentials>

Connect:Direct ブリッジ・エージェントが Connect:Direct ノードに接続するときに使用する資格情報を記述するエレメントを含むグループ・エレメント。

<agent>

名前付きエージェントの <pnode> 定義のエレメントが含まれているグループ・エレメント。

<pnode>

Connect:Direct 転送の 1 次ノード (PNODE)。このノードが 2 次ノード (SNODE) への接続を開始します。

属性	説明
名前	Connect:Direct ノードの名前。この属性の値は、多数のノード名と一致するパターンにすることができます。
pattern	name 属性の値として使用されるパターンのタイプを指定します。pattern 属性の有効な値は以下のとおりです。 <ul style="list-style-type: none"> wildcard - ワイルドカードを使用 regex - Java 正規表現を使用

<user>

転送要求を実行依頼する IBM MQ ユーザー。

属性	説明
名前	Managed File Transfer で使用されるユーザー名。この属性の値は、多数のユーザー名と一致するパターンにすることができます。
ignorecase	名前の大/小文字を無視するかどうかを指定します。ignorecase 属性の有効な値は以下のとおりです。 <ul style="list-style-type: none"> true - 名前に大/小文字の区別がない false - 名前に大/小文字の区別がある

属性	説明
pattern	name 属性の値として使用されるパターンのタイプを指定します。pattern 属性の有効な値は以下のとおりです。 <ul style="list-style-type: none"> • wildcard - ワイルドカードを使用 • regex - Java 正規表現を使用
cdUserId または cdUserIdCipher	Connect:Direct ブリッジが、関連付けられている Connect:Direct ノードに接続するために使用するユーザー名。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。
cdPassword または cdPasswordCipher	cdUserId 属性によって指定されたユーザー名に関連付けられているパスワード。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。
pnodeUserId または pnodeUserIdCipher	Connect:Direct 1 次ノードで使用するユーザー名。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。
pnodePassword または pnodePasswordCipher	pnodeUserId 属性によって指定されたユーザー名に関連付けられているパスワード。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。

<snode>

Connect:Direct ファイル転送の間、2 次ノード (SNODE) の役割を果たす Connect:Direct ノード。

属性	説明
名前	Connect:Direct ノードの名前。この属性の値は、多数のノード名と一致するパターンにすることができます。
pattern	name 属性の値として使用されるパターンのタイプを指定します。pattern 属性の有効な値は以下のとおりです。 <ul style="list-style-type: none"> • wildcard - ワイルドカードを使用 • regex - Java 正規表現を使用
userId または userIdCipher	ファイル転送の間、このノードに接続するために使用するユーザー名。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。
password または passwordCipher	userId 属性によって指定されたユーザー名に関連付けられているパスワード。 fteObfuscate コマンドが使用された場合は、暗号化された属性を使用する必要があります。

例

この例では、Connect:Direct ブリッジ・エージェントが Connect:Direct ノード pnode1 に接続します。接頭部 fteuser で始まり、その後に単一文字が続く IBM MQ ユーザー (例えば、fteuser2) が Connect:Direct ブリッジを含む転送を要求すると、Connect:Direct ブリッジ・エージェントはユーザー名 cduser とパスワード passw0rd を使用して Connect:Direct ノード pnode1 に接続します。Connect:Direct ノード pnode1 が転送における自らの役割を実行するときには、ユーザー名 pnodeuser およびパスワード passw0rd1 を使用します。

Connect:Direct 転送における 2 次ノードの名前が接頭部 FISH で始まる場合、ノード pnode1 はユーザー名 fishuser およびパスワード passw0rd2 を使用して 2 次ノードに接続します。Connect:Direct 転送に

おける 2 次ノードの名前が接頭部 CHIPS で始まる場合、ノード pnode1 はユーザー名 chipsuser およびパスワード passw0rd3 を使用して 2 次ノードに接続します。

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:credentials xmlns:tns="http://wmmqfte.ibm.com/ConnectDirectCredentials"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmmqfte.ibm.com/ConnectDirectCredentials
ConnectDirectCredentials.xsd">
  <tns:agent name="CDAGENT01">
    <tns:pnode name="pnode1" pattern="wildcard">
      <tns:user name="fteuser?" pattern="wildcard" ignorecase="true"
        cdUserId="cduser" cdPassword="passw0rd"
        pnodeUserId="pnodeuser" pnodePassword="passw0rd1">
      <tns:snode name="FISH*" pattern="wildcard"
        userId="fishuser" password="passw0rd2"/>
      <tns:snode name="CHIPS*" pattern="wildcard"
        userId="chipsuser" password="passw0rd3"/>
    </tns:user>
  </tns:pnode>
</tns:agent>
</tns:credentials>
```

Connect:Direct プロセスの定義ファイルのフォーマット

Connect:Direct ブリッジ・エージェントの構成ディレクトリーにある ConnectDirectProcessDefinitions.xml ファイルでは、ファイル転送の一部として開始するユーザー定義 Connect:Direct プロセスを指定します。

ConnectDirectProcessDefinitions.xml ファイルは ConnectDirectProcessDefinitions.xsd スキーマに準拠する必要があります。ConnectDirectProcessDefinitions.xsd スキーマ文書は、MFT インストール済み環境の MQ_INSTALLATION_PATH/mqft/samples/schema ディレクトリーにあります。ConnectDirectProcessDefinitions.xml テンプレート・ファイルは、**fteCreateCDAgent** コマンドにより、エージェントの構成ディレクトリーに作成されます。

ファイル ConnectDirectProcessDefinitions.xml はエージェントによって定期的に再ロードされ、ファイルに有効な変更が加えられていれば、それはエージェントの動作に反映されます。デフォルトの再ロード間隔は 30 秒です。この間隔は、agent.properties ファイルの xmlConfigReloadInterval エージェント・プロパティを指定することにより変更できます。

スキーマ

次のスキーマは、ConnectDirectProcessDefinitions.xml ファイル内で有効なエレメントについて示します。

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

```

ConnectDirectProcessDefinitions.xml ファイルについて

ConnectDirectProcessDefinitions.xml ファイルで使用するエレメントと属性を以下のリストにまとめます。

cdProcess

XML 文書のルート・エレメント。

processSet

一連のユーザー定義プロセスに関するすべての情報を格納したグループ・エレメント。

condition

processSet エレメントに含まれているプロセス・セットを使用するかどうかを決定するための転送の検査条件を格納したグループ・エレメント。

match

変数の値が特定の値に合致するかどうかを検査するための条件。

属性	説明
variable	変数を指定します。この変数の値が value 属性の値と比較されます。この変数は、組み込みシンボルです。詳しくは、660 ページの『ユーザー定義 Connect:Direct プロセスで使用する置換変数』を参照してください。
値	variable 属性で指定する変数の値と突き合わせるパターンを指定します。
pattern	value 属性で指定する変数の値と突き合わせるパターンを指定します。pattern 属性の有効な値は以下のとおりです。 <ul style="list-style-type: none"> wildcard - ワイルドカードを使用

属性	説明
	<ul style="list-style-type: none"> • regex - Java 正規表現を使用 この属性は任意指定であり、デフォルトは wildcard です。

defined

変数が定義されているかどうかを検査するための条件。

属性	説明
variable	変数を指定します。この変数が存在すると、この一致条件は満たされたこととなります。この変数は、組み込みシンボルです。詳しくは、660 ページの『ユーザー定義 Connect:Direct プロセスで使用する置換変数』を参照してください。

process

一致項目が見つかったときに呼び出す Connect:Direct プロセスの場所に関する情報を格納したグループ・エレメント。

transfer

転送要求の実行中に呼び出す Connect:Direct プロセス。

属性	説明
process	オプション。転送要求の実行中に呼び出す Connect:Direct プロセスを格納したファイルの名前を指定します。ファイル・パスは、Connect:Direct ブリッジ・エージェントの構成ディレクトリーを基準にした相対パスです。この属性は任意指定であり、デフォルトは、MFT によって生成されたプロセスを使用する、という動作です。IBM WebSphere MQ 7.5 以降の場合は、このプロパティーの値に環境変数を含めることができます。詳しくは、535 ページの『MFT プロパティーでの環境変数の使用』を参照してください。

例

この例では、3 つの processSet エレメントがあります。

最初の processSet エレメントは、転送要求の **%FTESNODE** 変数の値がパターン **Client*** と一致し、**%FTESUSER** 変数の値が **Admin** である場合に、Connect:Direct ブリッジ・エージェントが *agent_configuration_directory/AdminClient.cdp* にある Connect:Direct プロセスを転送の一部として実行依頼することを指定します。

2 番目の processSet エレメントは、転送要求にパターン **Client*** と一致する値を持つ **%FTESNODE** 変数がある場合に、Connect:Direct ブリッジ・エージェントが *agent_configuration_directory/Client.cdp* にある Connect:Direct プロセスを転送の一部として実行依頼することを指定します。Connect:Direct ブリッジ・エージェントは、各 processSet エレメントを定義順に読み取り、一致項目が見つかったら、最初の一致項目を使用し、その他の一致項目は検索しません。1 番目と 2 番目の両方の processSet の条件に合致する転送要求の場合、Connect:Direct ブリッジ・エージェントは、1 番目の processSet で指定されているプロセスだけを呼び出します。

3 番目の processSet エレメントは、条件がないので、すべての転送に合致します。1 番目と 2 番目の processSet の条件に合致しない転送要求の場合、Connect:Direct ブリッジ・エージェントは、3 番目の条件で指定されている Connect:Direct プロセスを実行します。転送の一部として呼び出されるこのプロセスは、*agent_configuration_directory/Default.cdp* にあります。

```
<?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:processSet>
</tns:cdprocess>
```

```

        </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 ノードのプロパティ・ファイルのフォーマット

Connect:Direct ブリッジ・エージェントの構成ディレクトリーにある ConnectDirectNodeProperties.xml ファイルでは、ファイル転送にかかわっているリモート Connect:Direct ノードに関する情報を指定します。

ConnectDirectNodeProperties.xml ファイルは ConnectDirectNodeProperties.xsd スキーマに準拠する必要があります。ConnectDirectNodeProperties.xsd スキーマ文書は、MFT インストール済み環境の MQ_INSTALLATION_PATH/mqft/samples/schema ディレクトリーにあります。ConnectDirectNodeProperties.xml テンプレート・ファイルは、**fteCreateCDAgent** コマンドにより、エージェントの構成ディレクトリーに作成されます。

ファイル ConnectDirectNodeProperties.xml はエージェントによって定期的に再ロードされ、ファイルに有効な変更が加えられていれば、それはエージェントの動作に反映されます。デフォルトの再ロード間隔は 30 秒です。この間隔は、agent.properties ファイルの xmlConfigReloadInterval エージェント・プロパティを指定することにより変更できます。

スキーマ

次のスキーマは、ConnectDirectNodeProperties.xml ファイル内で有効なエレメントについて示します。

```

<?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>

```

ConnectDirectNodeProperties.xml ファイルについて

ConnectDirectNodeProperties.xml ファイルで使用するエレメントと属性を以下のリストにまとめます。

nodeProperties

XML 文書のルート・エレメント。

credentialsFile

機密情報が保管されている資格情報ファイルのパス。IBM WebSphere MQ 7.5 以降の場合は、このプロパティの値に環境変数を含めることができます。詳しくは、[535 ページの『MFT プロパティでの環境変数の使用』](#)を参照してください。

ノード

Connect:Direct ノードを 1 つ以上指定します。

属性	説明
名前	node エレメントで指定する定義を使用する Connect:Direct ノードの名前を識別するパターン。パターン・マッチングには、大/小文字の区別がありません。
pattern	name 属性で指定する変数の値と突き合わせるパターンを指定します。pattern 属性の有効な値は以下のとおりです。 <ul style="list-style-type: none"> wildcard - ワイルドカードを使用 regex - Java 正規表現を使用 MFT で使用する正規表現のタイプについては、 659 ページの『MFT が使用する正規表現』 を参照してください。
タイプ	name 属性で指定するパターンに適合する Connect:Direct ノードのオペレーティング・システムのタイプを指定します。type 属性の有効な値は以下のとおりです。 <ul style="list-style-type: none"> Windows - ノードは Windows で実行されています UNIX - ノードは UNIX または Linux で実行されています  z/OS - z/OS、zos、os/390、os390 - ノードは z/OS で実行されています この属性の値には、大/小文字の区別がありません。

例

この例では、Connect:Direct 資格情報ファイル名を ConnectDirectCredentials.xml として指定します。サンプル・コードには、以下のプラットフォーム接続を指定します。

- 「"cdnodew"」で始まる名前の Connect:Direct ノードがすべて Windows プラットフォームで稼働している。
- 「"cdnodeu"」で始まる名前を持つすべての Connect:Direct ノードは、UNIX プラットフォーム上で実行されます。
- 「"cdnodez"」で始まる名前の Connect:Direct ノードがすべて z/OS プラットフォームで稼働している。
- 他のすべての Connect:Direct ノードは UNIX プラットフォームで稼働している。

Connect:Direct ブリッジ・エージェントは、このファイルの先頭から末尾に向かって一致項目を検索し、最初に見つかった一致項目を使用します。

```
<?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:nodeProperties>
```

```

<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 スキーマ・ファイル

このスキーマは、他の多くの Managed File Transfer スキーマで使用される要素およびタイプを定義します。

スキーマ

```

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

Licensed Materials - Property of IBM

5724-H72

Copyright IBM Corp. 2008, 2023. All Rights Reserved.

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

<!--
This schema defines elements and types used by many of the other MQMFT schemas.
For more information about MQMFT XML message formats, see
https://www.ibm.com/docs/SSEP7X_7.0.4/com.ibm.wmqfte.doc/message_formats.htm
-->
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
  <!--
    Defines the version type 1.00 - 99.00
  <transaction version= 1.00
  -->
  <xsd:simpleType name="versionType">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="[0-9]+\.[0-9][0-9]"/>
    </xsd:restriction>
  </xsd:simpleType>
  <!--
    Defines the transaction reference
  <transaction version= 1.00 ID="414d5120514d3120202020202020205ecf0a4920011802"
  -->
  <xsd:simpleType name="IDType">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="[0-9a-fA-F]{48}"/>
    </xsd:restriction>
  </xsd:simpleType>
  <!--
    This is an alias for hostUserIDType.
    Here to allow addition of attributes on originator elements
  -->
  <xsd:complexType name="origRequestType">
    <xsd:complexContent>
      <xsd:extension base="hostUserIDType">
        <xsd:sequence>
          <xsd:element name="webBrowser" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
          <xsd:element name="webUserID" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <!--
    Defines a Delete originator as a machine and user pair
  <hostName>myMachine</hostName>
  <userName>myUserId</userName>
  -->
  <xsd:complexType name="origDeleteType">
    <xsd:sequence>
      <xsd:element name="delete" type="hostUserIDType" maxOccurs="1" minOccurs="0"/>

```

```

    </xsd:sequence>
</xsd:complexType>
<!--
    Defines a machine, user, MQMD userID triple
    <hostName>myMachine</hostName>
    <userID>myUserId</userID>
    <mqmdUserID>MQMDUSERID</mqmdUserID>
-->
<xsd:complexType name="hostUserIDType">
  <xsd:sequence>
    <xsd:element name="hostName" type="xsd:string" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="userID" type="xsd:string" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="mqmdUserID" type="xsd:string" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
<!--
    Define the destinationAgent with agent and queue manager name
    <destinationAgent agent="there" QMgr="far" agentType="BRIDGE" bridgeURL="ftp://
server.example.net:21" bridgeNode="DEST_NODE"/>
    optional agentType attribute expected to be one of STANDARD, BRIDGE, WEB_GATEWAY,
EMBEDDED, CD_BRIDGE
-->
<xsd:complexType name="agentType">
  <xsd:attribute name="agent" type="xsd:string" use="required"/>
  <xsd:attribute name="agentType" type="xsd:string" use="optional"/>
  <xsd:attribute name="QMgr" type="xsd:string" use="optional"/>
  <xsd:attribute name="bridgeURL" type="xsd:string" use="optional"/>
  <xsd:attribute name="bridgeNode" type="xsd:string" use="optional"/>
  <xsd:attribute name="pnode" type="xsd:string" use="optional"/>
  <xsd:attribute name="snode" type="xsd:string" use="optional"/>
</xsd:complexType>
<!--
    Defines the status type; attr/resultCode and 0 or many supplements
    There may also be additional command specific data, either: transfer, ping or call data
    <status resultCode="8011">
      <supplement>Azionamento del USB</supplement>
      <supplement>morto come norweign azzurro</supplement>
    </status>
-->
<xsd:complexType name="statusType">
  <xsd:sequence>
    <xsd:element name="supplement" type="xsd:string" maxOccurs="unbounded"
minOccurs="0"/>
    <xsd:choice>
      <xsd:element name="fileSpace" type="fileSpaceReplyType" minOccurs="0"
maxOccurs="1"/>
    </xsd:choice>
  </xsd:sequence>
  <xsd:attribute name="resultCode" type="resultCodeType" use="required"/>
</xsd:complexType>
<!--
    Defines the fileSpace type for use with communication between a web agent
and a web gateway
    <fileSpace name="" location=""><Quota bytes=""></fileSpace>
-->
<xsd:complexType name="fileSpaceReplyType">
  <xsd:attribute name="name" use="required" type="xsd:string"/>
  <xsd:attribute name="location" use="required" type="xsd:string"/>
  <xsd:attribute name="quota" use="required" type="xsd:long"/>
</xsd:complexType>
<!--
    Defines the destinationAgent with agent and queue manager name, plus connection
details.
    <destinationAgent agent="there" QMgr="far"/>
-->
<xsd:complexType name="agentClientType">
  <xsd:attribute name="agent" type="xsd:string" use="required"/>
  <xsd:attribute name="QMgr" type="xsd:string" use="optional"/>
  <xsd:attribute name="hostName" type="xsd:string" use="optional"/>
  <xsd:attribute name="portNumber" type="xsd:nonNegativeInteger" use="optional"/>
  <xsd:attribute name="channel" type="xsd:string" use="optional"/>
</xsd:complexType>
<!--
    Defines the fileURI type as string
    <file encoding="UTF8" EOL="CR">C:/from/here.txt</file>
-->
<xsd:complexType name="fileType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="fileSpace" type="fileSpaceNameType" use="optional"/>
      <xsd:attribute name="alias" type="xsd:string" use="optional"/>
      <xsd:attribute name="encoding" type="encodingType" use="optional"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

```

        <xsd:attribute name="EOL" type="EOLType" use="optional"/>
        <xsd:attribute name="size" type="xsd:long" use="optional"/>
        <xsd:attribute name="last-modified" type="xsd:dateTime" use="optional"/>
        <xsd:attribute name="delimiter" type="xsd:string" use="optional"/>
        <xsd:attribute name="delimiterType" type="xsd:string" use="optional"/>
        <xsd:attribute name="delimiterPosition" type="delimiterPositionType"
use="optional"/>
        <xsd:attribute name="includeDelimiterInFile" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="keepTrailingSpaces" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="truncateRecords" type="xsd:boolean" use="optional"/>
    </xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<!--
    Defines the filesystem type as string
    <filesystem>
        <name>tarquin</name>
    </filesystem>
-->
<xsd:complexType name="filesystemType">
    <xsd:sequence>
        <xsd:element name="name" type="filesystemNameType"/>
    </xsd:sequence>
</xsd:complexType>
<!--
    Defines a name element
    <name>bob</name>
-->
<xsd:simpleType name="filesystemNameType">
    <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<!--
    Defines the accepted choices for the persistent attribute.
-->
<xsd:simpleType name="persistenceType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="true"/>
        <xsd:enumeration value="false"/>
        <xsd:enumeration value="qdef"/>
    </xsd:restriction>
</xsd:simpleType>
<!--
    Defines the queueURI type as string with all supported attributes.
    <queue>QUEUE@QM</queue>
-->
<xsd:complexType name="queueType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute name="persistent" type="persistenceType" use="optional"/>
            <xsd:attribute name="eofMarker" type="xsd:boolean" use="optional"/>
            <xsd:attribute name="setMqProps" type="xsd:boolean" use="optional"/>
            <xsd:attribute name="split" type="xsd:boolean" use="optional"/>
            <xsd:attribute name="useGroups" type="xsd:boolean" use="optional"/>
            <xsd:attribute name="delimiter" type="xsd:string" use="optional"/>
            <xsd:attribute name="delimiterType" type="xsd:string" use="optional"/>
            <xsd:attribute name="delimiterPosition" type="delimiterPositionType"
use="optional"/>
            <xsd:attribute name="includeDelimiterInMessage" type="xsd:boolean"
use="optional"/>
            <xsd:attribute name="groupId" type="groupIdType" use="optional"/>
            <xsd:attribute name="messageId" type="messageIdType" use="optional"/>
            <xsd:attribute name="messageCount" type="xsd:nonNegativeInteger"
use="optional"/>
            <xsd:attribute name="messageLength" type="xsd:nonNegativeInteger"
use="optional"/>
            <xsd:attribute name="waitTime" type="xsd:nonNegativeInteger" use="optional"/>
            <xsd:attribute name="encoding" type="encodingType" use="optional"/>
            <xsd:attribute name="EOL" type="EOLType" use="optional"/>
            <xsd:attribute name="unrecognisedCodePage" type="unrecognisedCodePageType"
use="optional"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
<!--
    Defines the accepted values for the delimiterPosition attribute.
-->
<xsd:simpleType name="delimiterPositionType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="postfix"/>
        <xsd:enumeration value="prefix"/>
    </xsd:restriction>
</xsd:simpleType>

```

```

<!--
  Defines the groupId type
  <queue groupId="414d5120514d31202020202020202020205ecf0a4920011802">
  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="414d5120514d31202020202020202020205ecf0a4920011802">
  Also allow a substitution variable of the form ${variable}
  -->
<xsd:simpleType name="messageIdType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[0-9a-fA-F]{48}|${.*\}"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- Defines the accepted values for the unrecognisedCodePage attribute. -->
<xsd:simpleType name="unrecognisedCodePageType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="fail"/>
    <xsd:enumeration value="binary"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines a single source file reference
  <source type="file" recursive="false" disposition="leave">
    <file>filename</file>
  </source>
  -->
<xsd:complexType name="fileSourceType">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="file" type="fileType"/>
      <xsd:element name="queue" type="queueType"/>
    </xsd:choice>
  </xsd:sequence>
  <xsd:attribute name="type" type="SourceType" use="optional"/>
  <xsd:attribute name="recursive" type="xsd:boolean" use="optional"/>
  <xsd:attribute name="disposition" type="sourceDispositionType" use="optional"/>
  <xsd:attribute name="correlationString1" type="xsd:string" use="optional"/>
  <xsd:attribute name="correlationNum1" type="xsd:nonNegativeInteger" use="optional"/>
  <xsd:attribute name="correlationBoolean1" type="xsd:boolean" use="optional"/>
</xsd:complexType>
<!--
  Defines the enumeration values for source type
  type="file|queue"
  -->
<xsd:simpleType name="SourceType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="file"/>
    <xsd:enumeration value="directory"/>
    <xsd:enumeration value="queue"/>
    <xsd:enumeration value="dataset"/>
    <xsd:enumeration value="pds"/>
    <xsd:enumeration value="filespace"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines the enumeration values for source disposition
  disposition="leave|delete"
  -->
<xsd:simpleType name="sourceDispositionType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="leave"/>
    <xsd:enumeration value="delete"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines a single destination file reference
  <destination type="file" exist="overwrite">
    <file>filename</file>
  </destination>
  -->
<xsd:complexType name="fileDestinationType">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="file" type="fileType"/>
      <xsd:element name="filespace" type="filespaceType"/>
    </xsd:choice>
  </xsd:sequence>

```

```

        <xsd:element name="queue" type="queueType"/>
    </xsd:choice>
    <xsd:element name="attributes" type="attributeType" minOccurs="0" maxOccurs="1"/>
</xsd:sequence>
<xsd:attribute name="type" type="DestinationType" use="optional"/>
<xsd:attribute name="exist" type="existType" use="optional"/>
<xsd:attribute name="correlationString1" type="xsd:string" use="optional"/>
<xsd:attribute name="correlationNum1" type="xsd:nonNegativeInteger" use="optional"/>
<xsd:attribute name="correlationBoolean1" type="xsd:boolean" use="optional"/>
</xsd:complexType>
<!--
    Defines the enumeration values for destination file type
    type="file|directory|queue|dataset|pds|filespace"
    'dataset' and 'pds' only apply to z/OS environments.
-->
<xsd:simpleType name="DestinationType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="file"/>
        <xsd:enumeration value="directory"/>
        <xsd:enumeration value="queue"/>
        <xsd:enumeration value="dataset"/>
        <xsd:enumeration value="pds"/>
        <xsd:enumeration value="filespace"/>
    </xsd:restriction>
</xsd:simpleType>
<!--
    Defines the enumerations values for file exists on destination behavior
    exist="error|overwrite"
-->
<xsd:simpleType name="existType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="error"/>
        <xsd:enumeration value="overwrite"/>
    </xsd:restriction>
</xsd:simpleType>
<!--
    Defines one or more file attributes
    <destination encoding=? CFLF=?>
        <file>filename</file>
        <attributes>
            <attribute>DIST(MIRRORED,UPDATE)</attribute>
        </attributes>
    </destination/>
-->
<xsd:complexType name="attributeType">
    <xsd:sequence>
        <xsd:element name="attribute" type="xsd:string" maxOccurs="unbounded"
minOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>
<!--
    Defines a single file reference
    <source encodings=? CFLF=?>
        <file>filename</file>
        <checksum method="MD5">3445678</checksum>
    </source/>
    .. or ..
    <destination encoding=? CFLF=?>
        <file>filename</file>
        <checksum method="MD5">3445678</checksum>
    </destination/>
-->
<xsd:complexType name="fileCheckSumType">
    <xsd:sequence>
        <xsd:element name="file" type="fileType"/>
        <xsd:element name="checksum" type="checksumType" maxOccurs="1" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
<!--
    Defines the checksum type and method
    <checksum method="MD5|none">3445678</checksum>
-->
<xsd:complexType name="checksumType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute name="method" type="checkSumMethod" use="required"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>

```

```

<!--
  Defines the enumeration values for checksumMethod
  <checksum method="MD5|none">3445678</checksum>
  Note: uppercase is used since MD5 is acronym and normally written uppercase.
-->
<xsd:simpleType name="checkSumMethod">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="none"/>
    <xsd:enumeration value="MD5"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines the enumeration values for agentRole
  agentRole="sourceAgent|destinationAgent"
-->
<xsd:simpleType name="agentRoleType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="sourceAgent"/>
    <xsd:enumeration value="destinationAgent"/>
    <xsd:enumeration value="callAgent"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines the enumeration values for mode.
  text, binary or a substitution variable
  <item mode="binary|text|${variableName}">
-->
<xsd:simpleType name="modeType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="binary|text|${.*\}"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines the enumeration values for EOL
  <file EOL="LF|CRLF">
-->
<xsd:simpleType name="EOLType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="LF"/>
    <xsd:enumeration value="CRLF"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines the encoding type as a string
-->
<xsd:simpleType name="encodingType">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<!--
  <schedule>
    <submit timebase="source|admin">2008-12-07T16:07</submit>
    <repeat>
      <frequency interval="hours">2</frequency>
      <expireTime>2008-12-0816:07</exipreTime>
    </repeat>
  </schedule>
-->
<xsd:complexType name="scheduleType">
  <xsd:sequence>
    <xsd:element name="submit" type="submitType" maxOccurs="1" minOccurs="1"/>
    <xsd:element name="repeat" type="repeatType" maxOccurs="1" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<!--
  <submit timebase="source|admin|UTC">2008-12-07T16:07</submit>
-->
<xsd:complexType name="submitType">
  <xsd:simpleContent>
    <xsd:extension base="noZoneTimeType">
      <xsd:attribute name="timebase" type="timebaseType" use="required"/>
      <xsd:attribute name="timezone" type="xsd:string" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<!--
  <repeat>
    <frequency interval="hours">2</frequency>
    ..optionally..
    <expireTime>2008-12-0816:07</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:sequence>

```

```

        <xsd:element name="preDestinationData" type="actionReplyType" minOccurs="0"
maxOccurs="1"/>
        <xsd:element name="postDestinationData" type="actionReplyType" minOccurs="0"
maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>
<!--
    Define the action reply type information
    <actionReply>
        <runCommandReply resultCode="1">
            <stdout></stdout>
            <stderr>
                <line>permission denied</line>
            </stderr>
        </runCommandReply>
    </actionReply>
-->
<xsd:complexType name="actionReplyType">
    <xsd:choice>
        <xsd:element name="runCommandReply" type="commandReplyType" maxOccurs="1"
minOccurs="0"/>
    </xsd:choice>
</xsd:complexType>
<!--
    Defines command specific reply information, to be included as part the general reply
    <commandReply resultCode="0">
        <stdout>
            <line>first line of output text</line>
            <line>second line of output text</line>
        </stdout>
        <stderr>
            <line>line of error text</line>
        </stderr>
    </commandReply>
-->
<xsd:complexType name="commandReplyType">
    <xsd:sequence>
        <xsd:element name="stdout" type="textLinesType" maxOccurs="1" minOccurs="1"/>
        <xsd:element name="stderr" type="textLinesType" maxOccurs="1" minOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="resultCode" type="xsd:int" use="required"/>
</xsd:complexType>
<!-- Defines type for lines of text -->
<xsd:complexType name="textLinesType">
    <xsd:sequence>
        <xsd:element name="line" type="xsd:string" maxOccurs="unbounded" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
<!--
    Defines the ping agent command specific reply information, to be included as part the
    general reply
    <pingAgentReply resultCode="0">
        <agentVersion>Build level: f000-20090408-1200</agentVersion>
    </pingAgentReply>
-->
<xsd:complexType name="pingAgentReplyType">
    <xsd:sequence>
        <xsd:element name="agentVersion" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
<!--
    Defines sequence of exit elements
    <exit ...
    <exit ...
-->
<xsd:complexType name="exitGroupType">
    <xsd:sequence>
        <xsd:element name="exit" type="exitType" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
<!--
    Defines the outcome of calling a command
    <command ...
    <callResult ...
-->
<xsd:complexType name="callGroupType">
    <xsd:sequence>
        <xsd:element name="command" type="commandType" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="callResult" type="callResultType" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>
<!--

```

```

    Defines either the successful call of a command, or a failed attempt to call a command
    <callResultType outcome="success|failure|error" retries="X">
      <result ... />
    </callResultType>
  -->
<xsd:complexType name="callResultType">
  <xsd:sequence>
    <xsd:element name="result" type="resultType" minOccurs="1" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="returnCode" type="xsd:integer" use="optional"/>
  <xsd:attribute name="retries" type="xsd:integer" use="optional"/>
  <xsd:attribute name="outcome" type="outcomeType" use="required"/>
</xsd:complexType>
<!--
  Defines the information recorded for the successful call of a command
  <result...>
    <stdout...
    <stderr...
    <error...
  </result...>
  -->
<xsd:complexType name="resultType">
  <xsd:sequence>
    <xsd:element name="stdout" type="outputType" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="stderr" type="outputType" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="error" type="xsd:string" maxOccurs="1" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="returnCode" type="xsd:integer" use="optional"/>
  <xsd:attribute name="outcome" type="outcomeType" use="required"/>
  <xsd:attribute name="time" type="xsd:dateTime" use="required"/>
</xsd:complexType>
<!-- Enumeration of call outcomes - success, failure or error -->
<xsd:simpleType name="outcomeType">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="success"/>
    <xsd:enumeration value="failure"/>
    <xsd:enumeration value="error"/>
  </xsd:restriction>
</xsd:simpleType>
<!--
  Defines the information recorded for each line of standard output / standard error
  generated by calling a program
  <line>line 1</line>
  <line>line 2</line>
  etc.
  -->
<xsd:complexType name="outputType">
  <xsd:sequence>
    <xsd:element name="line" type="xsd:string" maxOccurs="unbounded" minOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
<!--
  Defines the information recorded for an unsuccessful program call.
  -->
<xsd:complexType name="callFailedType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string"/>
  </xsd:simpleContent>
</xsd:complexType>
<!--
  Defines the exit type; records the transfer exit class name and a status message
  <exit name="class com.example.exit.StartExit">
    <status ...
  </exit>
  -->
<xsd:complexType name="exitType">
  <xsd:sequence>
    <xsd:element name="status" type="exitStatusType" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="name" type="xsd:string" use="required"/>
</xsd:complexType>
<!--
  Defines exit status to record whether exit voted to proceed or cancel transfer.
  <status resultCode="proceed">
    <supplement>go ahead</supplement>
  </status>
  -->
<xsd:complexType name="exitStatusType">
  <xsd:sequence>
    <xsd:element name="supplement" type="xsd:string" maxOccurs="unbounded"
minOccurs="0"/>
  </xsd:sequence>

```

```

    <xsd:attribute name="resultCode" type="exitResultEnumType" use="optional"/>
  </xsd:complexType>
  <!--
    Defines the enumeration for transfer exit result values.
    <status resultCode="proceed">
  -->
  <xsd:simpleType name="exitResultEnumType">
    <xsd:restriction base="xsd:token">
      <xsd:enumeration value="proceed"/>
      <xsd:enumeration value="cancelTransfer"/>
      <xsd:enumeration value="cancelTask"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>

```

注：IBM MQ 9.0 から、Managed File Transfer は Web ゲートウェイや Web エージェントをサポートしなくなりました。

関連概念

733 ページの『[MFT で使用する XML メッセージ形式](#)』

Managed File Transfer では、エージェントへの指示、モニターに関する情報の記録、スケジューリング、転送、および構成に使用される情報の定義など、さまざまな目的のために XML 形式のメッセージを使用します。これらの目的のために使用される XML 形式の論理構造は XML スキーマにより記述されます。

MFT アプリケーション開発リファレンス

ファイル転送前後のプログラムの実行

fteCreateTransfer を使用してプログラムを開始する例

fteCreateTransfer コマンドを使用して、転送前または転送後に実行するプログラムを指定することができます。

fteCreateTransfer を使用する以外にも、転送前または転送後にプログラムを起動する方法があります。詳細内容は [を参照してください](#)。

これらの例はすべて、以下の構文を使用してプログラムを指定します。

```
[type:]commandspec[, [retrycount][, [retrywait][, successrc]]]
```

この構文について詳しくは、[437 ページの『fteCreateTransfer: 新規ファイル転送の開始』](#)を参照してください。

実行可能プログラムの実行

以下の例は、mycommand という実行可能プログラムを指定し、そのプログラムに 2 つの引数 a および b を渡します。

```
mycommand(a,b)
```

このプログラムを転送の開始前にソース・エージェント AGENT1 で実行するには、次のコマンドを使用します。

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -presrc mycommand(a,b)
destinationSpecification sourceSpecification
```

実行可能プログラムの実行と再試行

以下の例は、simple という実行可能プログラムを指定しています。このプログラムは引数を取りません。retrycount には値 1 が指定され、retrywait には値 5 が指定されています。これらの値は、そのプロ

グラムが成功を表す戻りコードを返さない場合には、5秒後に1回再試行されることを意味しています。successrcには値が指定されていないため、成功を表す戻りコードはデフォルト値の0のみです。

```
executable:simple,1,5
```

このプログラムを転送の完了後にソース・エージェント AGENT1 で実行するには、次のコマンドを使用します。

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -postsrc executable:simple,1,5  
destinationSpecification sourceSpecification
```

Ant スクリプトの実行と成功を表す戻りコードの指定

以下の例は、myscript という Ant スクリプトを指定し、そのスクリプトに2つのプロパティを渡します。このスクリプトは、**fteAnt** コマンドを使用して実行されます。successrc の値は >2&<7&!5|0|14 として指定されます。この値は、戻りコード 0、3、4、6、および 14 が成功を示すことを指定します。

```
antscript:myscript(prop1=fred,prop2=bob),,,>2&<7&!5|0|14
```

このプログラムを転送の開始前に宛先エージェント AGENT2 で実行するには、次のコマンドを使用します。

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -predst  
"antscript:myscript(prop1=fred,prop2=bob),,,>2&<7&!5|0|14"destinationSpecification sourceSpecification
```

Ant スクリプトの実行と呼び出すターゲットの指定

以下の例は、script2 という Ant スクリプトを指定し、呼び出す2つのターゲット target1 および target2 を指定します。プロパティ prop1 も値 recmfm(F,B) を指定して渡されます。この値の中のコンマ (,) と括弧は、円記号 (¥) を使用してエスケープされています。

```
antscript:script2(target1,target2,prop1=recmfm\F,B),,,>2&<7&!5|0|14
```

このプログラムを転送の完了後に宛先エージェント AGENT2 で実行するには、次のコマンドを使用します。

```
fteCreateTransfer -sa AGENT1 -da AGENT2  
-postdst "antscript:script2(target1,target2,prop1=recmfm\F,B),,,>2&<7&!5|0|14"  
destinationSpecification sourceSpecification
```

Ant スクリプトでのメタデータの使用

転送の以下の呼び出しに Ant タスクを指定できます。

- ソースの前
- ソースの後
- 宛先の前
- 宛先の後

Ant タスクの実行時には、環境変数を使用して転送のユーザー・メタデータを提供できます。このデータを使用してアクセスできます。以下にコードの例を示します。

```
<property environment="environment"/>  
<echo>${environment.mymetadata}</echo>
```

mymetadata は、転送に挿入するメタデータの名前です。

JCL スクリプトの実行

以下の例では ZOSBATCH という JCL スクリプトを指定しています。retrycount には値 3 が指定され、retrywait には値 30 が指定され、successrc には値 0 が指定されています。これらの値は、そのスクリプトが成功を表す戻りコード 0 を返さない場合には、30 秒おきに 3 回再試行されることを意味しています。

```
jcl:ZOSBATCH,3,30,0
```

ZOSBATCH は、MYSYS.JCL という PDS メンバーであり、agent.properties ファイルには commandPath=.....:/'MYSYS.JCL':... という行が含まれています。

このプログラムを転送の完了後にソース・エージェント AGENT1 で実行するには、次のコマンドを使用します。

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -postsrc jcl:ZOSBATCH,3,30,0  
destinationSpecification sourceSpecification
```

関連概念

[250 ページの『MFT で実行するプログラムの指定』](#)

Managed File Transfer Agent が実行されているシステムで、プログラムを実行することができます。ファイル転送要求の一部として、転送の開始前または終了後のいずれかにプログラムを実行するように指定することができます。また、管理対象呼び出し要求を実行依頼することで、ファイル転送要求に含まれないプログラムを開始することも可能です。

関連資料

[437 ページの『fteCreateTransfer: 新規ファイル転送の開始』](#)

fteCreateTransfer コマンドは、コマンド行から新規のファイル転送を作成して開始します。このコマンドは、ファイル転送の即時開始、将来の日時におけるファイル転送のスケジュール、ファイル転送の 1 回以上の繰り返し、および特定の条件に基づくファイル転送の起動を実行できます。

MFT での Apache Ant の使用

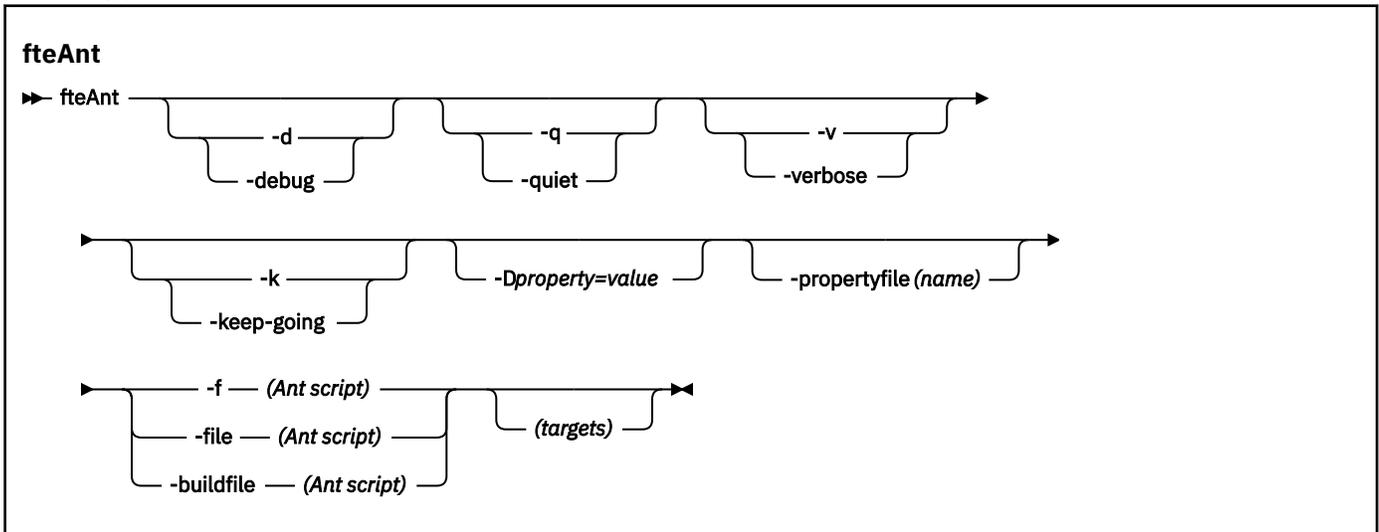
fteAnt: MFT で Ant タスクを実行します。

fteAnt コマンドは、Managed File Transfer の Ant タスクが使用可能な環境で Ant スクリプトを実行します。

目的

fteAnt コマンドは、Managed File Transfer がインストールされた環境で Ant スクリプトを実行する場合に使用します。標準の **ant** コマンドとは異なり、**fteAnt** ではスクリプト・ファイルを定義する必要があります。

構文



Parameters

-debug または **-d**

オプション。デバッグ出力を生成します。

-quiet または **-q**

オプション。最小出力を生成します。

-verbose または **-v**

オプション。詳細出力を生成します。

-keep-going または **-k**

オプション。失敗したターゲットに依存しないすべてのターゲットを実行します。

-D property=value

オプション。一定のプロパティの値を使用します。 **-D** で設定されたプロパティは、プロパティ・ファイル内に設定されたプロパティに優先します。

com.ibm.wmqfte.propertyset プロパティは、Ant タスクに使用される構成オプションのセットを指定するために使用します。このプロパティの値には、デフォルトでない調整キュー・マネージャーの名前を使用します。そうすることで、Ant タスクは、このデフォルトではない調整キュー・マネージャーに関連付けられた構成オプションのセットを使用します。このプロパティを指定しない場合、デフォルトの調整キュー・マネージャーに基づいたデフォルトの構成オプションのセットが使用されます。Ant タスクに対して **cmdqm** 属性を指定すると、**fteAnt** コマンド用に指定された構成オプションのセットよりもこの属性が優先されます。この振る舞いは、デフォルトの構成オプションのセットを使用するか、または **com.ibm.wmqfte.propertyset** プロパティでセットを指定するかにかかわらず適用されます。

-propertyfile (name)

オプション。すべてのプロパティを、優先順位を持つ **-D** プロパティが指定されたファイルからロードします。

-f (Ant スクリプト)、**-file (Ant スクリプト)**、または **-buildfile (Ant スクリプト)**

必須。実行する Ant スクリプトの名前を指定します。

targets

オプション。Ant スクリプトの実行元の 1 つ以上のターゲットの名前。このパラメーターに値を指定しない場合は、スクリプトのデフォルト・ターゲットが実行されます。

-バージョン

オプション。Managed File Transfer コマンドおよび Ant のバージョンを表示します。

-?または-h

オプション。コマンド構文を表示します。

例

この例では、Ant スクリプト `fte_script.xml` 内のターゲット **copy** が実行され、このコマンドにより、デバッグ出力が標準出力に書き込まれます。

```
fteAnt -d -f fte_script.xml copy
```

戻りコード

0

コマンドは正常に完了しました。

1

コマンドは失敗しました。

他の状況戻りコードは、Ant スクリプトからも、例えば、Ant fail タスクを使用するなどして指定できます。

詳しくは、[失敗](#)を参照してください。

MFT 提供の Ant タスク

Managed File Transfer では、数多くの Ant タスクが用意されており、これらのタスクを使用して、ファイル転送機能にアクセスできます。

タスク

- [848 ページの『fte:awaitoutcome の Ant タスク』](#)
- [fte:call](#)
- [fte:cancel](#)
- [fte:filecopy](#)
- [fte:filemove](#)
- [fte:ignoreoutcome](#)
- [fte:ping](#)
- [fte:uuid](#)

ネスト・パラメーター

次のネスト・パラメーターには、ネストされたエレメント・セットを記述します。これらは、提供 Ant タスクのいくつかにわたって共通です。

- [fte:filespec](#)
- [fte:metadata](#)
- [プログラム呼び出しのパラメーター](#)

fte:awaitoutcome の Ant タスク

fte:filecopy、**fte:filemove**、または **fte:call** のいずれかの操作が完了するのを待機します。

属性

id

必須。結果を待機する対象の転送を指定します。これは通常、[fte:filecopy](#)、[fte:filemove](#)、[fte:call](#) のタスクの `idProperty` 属性によって設定されたプロパティです。

rcproperty

必須。 **fte:awaitoutcome** タスクの戻りコードを保管するためのプロパティの名前を指定します。

timeout

オプション。操作が完了するまで待機する最大時間(秒単位)。タイムアウトの最小値は 1 秒です。タイムアウト値を指定しなかった場合には、**fte:awaitoutcome** タスクは操作の結果が決定するまで無期限で待機します。

例

次の例では、ファイル・コピーが開始され、その ID が `copy.id` プロパティに保管されています。コピーの進行中には、他の処理を行うことができます。**fte:awaitoutcome** ステートメントを使用して、`copy` 操作が完了するまで待機しています。この **fte:awaitoutcome** ステートメントでは、`copy.id` プロパティに保管されている ID を使用して、待機対象の操作を指定しています。この **fte:awaitoutcome** は、`copy` 操作の結果を示す戻りコードを、`copy.result` というプロパティに保管します。

```
<!-- 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>
```

関連資料

[252 ページの『MFT での Apache Ant の使用』](#)

Managed File Transfer は、ファイル転送機能を Apache Ant ツールに組み込むために使用できるタスクを提供します。

[848 ページの『MFT 提供の Ant タスク』](#)

Managed File Transfer では、数多くの Ant タスクが用意されており、これらのタスクを使用して、ファイル転送機能にアクセスできます。

fte:call の Ant タスク

fte:call タスクを使用して、スクリプトおよびプログラムをリモートで呼び出すことができます。

このタスクによって、**fte:call** 要求をエージェントに送信できます。エージェントは、スクリプトまたはプログラムを実行してその結果を返すことで、この要求を処理します。呼び出すコマンドは、エージェントにアクセス可能である必要があります。`agent.properties` ファイル内の `commandPath` プロパティ値に、必ず呼び出すコマンドの場所を含むようにしてください。コマンドがネストされているエレメントによって指定されたパス情報は、`commandPath` プロパティで指定された場所と相対的な位置になければなりません。デフォルトでは、`commandPath` は空であるため、エージェントほどのコマンドも呼び出すことができません。このプロパティの詳細については、[365 ページの『commandPath MFT プロパティ』](#)を参照してください。

`agent.properties` ファイルについて詳しくは、[550 ページの『MFT agent.properties ファイル』](#)を参照してください。

属性

エージェント

必須。**fte:call** 要求の実行依頼先のエージェントを指定します。エージェント情報は、`agentname@qmgrname` という形式で指定します(ここで、`agentname` はエージェントの名前、`qmgrname` は、当該エージェントが直接接続されている先のキュー・マネージャーの名前)。

cmdqm

オプション。要求の実行依頼先のコマンド・キュー・マネージャー。この情報は、`qmgrname@host@port@channel` という形式で指定します。ここで各ストリングは以下のとおりです。

- `qmgrname` は、キュー・マネージャーの名前です。
- `host` は、キュー・マネージャーを実行しているシステムのホスト名です (オプション)。
- `port` は、キュー・マネージャーが listen しているポート番号です (オプション)。
- `channel` は、使用するオプションの SVRCONN チャンネルです

コマンド・キュー・マネージャーの `host`、`port`、または `channel` 情報を省略した場合には、`command.properties` ファイルに指定された接続情報が使用されます。詳しくは、[546 ページの『MFT command.properties ファイル』](#)を参照してください。

com.ibm.wmqfte.propertySet プロパティを使用すると、使用する `command.properties` ファイルを指定できます。詳細については、[com.ibm.wmqfte.propertySet](#) を参照してください。

`cmdqm` 属性を使用しなかった場合には、タスクはデフォルトで `com.ibm.wmqfte.ant.commandQueueManager` プロパティを使用します (このプロパティが設定されている場合)。`com.ibm.wmqfte.ant.commandQueueManager` プロパティが設定されていない場合には、`command.properties` ファイルで定義されたデフォルトのキュー・マネージャーへの接続が試行されます。`com.ibm.wmqfte.ant.commandQueueManager` プロパティの形式は、`cmdqm` 属性と同じです。つまり、`qmgrname@host@port@channel` です。

idproperty

`outcome` に `defer` を指定していない場合は、オプションです。転送 ID を割り当てるプロパティの名前を指定します。転送 ID は、転送要求が実行依頼された時点で生成されます。この転送 ID を使用して、転送の進行の追跡、転送で生じた問題の診断、および転送の取り消しを行うことができます。

`outcome` プロパティに `ignore` を指定した場合には、このプロパティは指定できません。ただし、`outcome` プロパティに `defer` を指定した場合には、`idproperty` を指定する必要があります。

jobname

オプション。ジョブ名を **fte:call** 要求に割り当てます。ジョブ名を使用して、論理転送グループを作成できます。[861 ページの『fte:uuid の Ant タスク』](#) タスクを使用して、疑似固有ジョブ名を生成します。`jobname` 属性を使用しなかった場合には、タスクはデフォルトで `com.ibm.wmqfte.ant.jobName` プロパティ値を使用します (このプロパティが設定されている場合)。このプロパティが設定されていない場合には、**fte:call** 要求に関連付けられるジョブ名はありません。

origuser

オプション。**fte:call** 要求に関連付ける発信ユーザー ID を指定します。`origuser` 属性を使用しなかった場合には、タスクはデフォルトで Ant スクリプトを実行するために使用されるユーザー ID を使用します。

outcome

オプション。Ant スクリプトに制御を返す前に、タスクが **fte:call** 操作の完了まで待機するかどうかを決定します。以下のいずれかのオプションを指定します。

await

タスクは、戻る前に **fte:call** 操作が完了するまで待機します。`outcome` に `await` が指定されている場合、`idproperty` 属性はオプションです。

defer

タスクは、**fte:call** 要求がサブミットされるとすぐに戻り、`awaitoutcome` タスクまたは `ignoreoutcome` タスクのいずれかを使用して、呼び出し操作の結果が後で処理されることを想定します。`defer` の `outcome` が指定されている場合、`idproperty` 属性は必須です。

ignore

fte:call 操作の結果が重要ではない場合には、値 **ignore** を指定できます。この値を指定した場合、タスクは、コマンドの結果を追跡するためのリソースを割り当てずに、**fte:call** 要求が実行依頼されるとすぐに戻ります。ignore の outcome が指定されている場合、idproperty 属性は指定できません。

outcome 属性を指定しなかった場合には、タスクはデフォルトで **await** を使用します。

rcproperty

オプション。 **fte:call** 要求の結果コードを割り当てるプロパティの名前を指定します。結果コードには、**fte:call** 要求の全体的結果が反映されます。

outcome プロパティに **ignore** または **defer** を指定した場合には、このプロパティは指定できません。ただし、outcome プロパティに **await** を指定した場合には、**rcproperty** を指定する必要があります。

ネスト・エレメントとして指定するパラメーター

fte:command

エージェントで呼び出すコマンドを指定します。特定の **fte:call** 操作に関連付けることができる **fte:command** エレメントは1つのみです。呼び出すコマンドは、エージェントの **agent.properties** ファイル内の **commandPath** プロパティで指定されたパス内に存在している必要があります。

fte:metadata

call 操作に関連付けるメタデータを指定できます。このメタデータは、call 操作で生成されたログ・メッセージに記録されます。特定の転送エレメントには、単一のメタデータ・ブロックのみを関連付けることができます。ただし、このブロックには、多くのメタデータを含めることができます。

例

次の例では、キュー・マネージャー QM1 で実行されている AGENT1 でコマンドを呼び出す方法を示します。呼び出すコマンドは、スクリプト **command.sh** であり、このスクリプトは単一の引数 **xyz** を指定して呼び出されています。コマンド **command.sh** は、エージェントの **agent.properties** ファイル内の **commandPath** プロパティで指定されたパス内に存在している必要があります。

```
<fte:call cmdqm="QM0@localhost@1414@SYSTEM.DEF.SVRCONN"
  agent="AGENT1@QM1"
  rcproperty="call.rc"
  origuser="bob"
  jobname="${job.id}">

  <fte:command command="command.sh" successrc="1" retrycount="5" retrywait="30">
    <fte:arg value="xyz"/>
  </fte:command>

  <fte:metadata>
    <fte:entry name="org.foo.accountName" value="BDG3R"/>
  </fte:metadata>

</fte:call>
```

関連資料

[252 ページの『MFT での Apache Ant の使用』](#)

Managed File Transfer は、ファイル転送機能を Apache Ant ツールに組み込むために使用できるタスクを提供します。

[848 ページの『MFT 提供の Ant タスク』](#)

Managed File Transfer では、数多くの Ant タスクが用意されており、これらのタスクを使用して、ファイル転送機能にアクセスできます。

fte:cancel の Ant タスク

Managed File Transfer 管理対象転送または管理対象呼び出しを取り消します。管理対象転送は、**fte:filecopy** タスクまたは **fte:filemove** タスクを使用して作成された可能性があります。管理対象呼び出しは、**fte:call** タスクを使用して作成された可能性があります。

属性

agent

必須。 **fte:cancel** 要求の実行依頼先のエージェントを指定します。この値は *agentname@qmgrname* の形式です。ここで *agentname* はエージェントの名前で、*qmgrname* は、このエージェントが直接接続しているキュー・マネージャーの名前です。

cmdqm

オプション。要求の実行依頼先のコマンド・キュー・マネージャー。この情報は、*qmgrname@host@port@channel* という形式で指定します。ここで各ストリングは以下のとおりです。

- *qmgrname* は、キュー・マネージャーの名前です。
- *host* は、キュー・マネージャーを実行しているシステムのホスト名です (オプション)。
- *port* は、キュー・マネージャーが *listen* しているポート番号です (オプション)。
- *channel* は、使用するオプションの SVRCONN チャンネルです

コマンド・キュー・マネージャーの *host*、*port*、または *channel* 情報を省略した場合には、*command.properties* ファイルに指定された接続情報が使用されます。詳しくは、[546 ページの『MFT command.properties ファイル』](#)を参照してください。

com.ibm.wmqfte.propertySet プロパティを使用すると、使用する *command.properties* ファイルを指定できます。詳細については、[com.ibm.wmqfte.propertySet](#) を参照してください。

cmdqm 属性を使用しなかった場合には、タスクはデフォルトで *com.ibm.wmqfte.ant.commandQueueManager* プロパティを使用します (このプロパティが設定されている場合)。 *com.ibm.wmqfte.ant.commandQueueManager* プロパティが設定されていない場合には、*command.properties* ファイルで定義されたデフォルトのキュー・マネージャーへの接続が試行されます。 *com.ibm.wmqfte.ant.commandQueueManager* プロパティの形式は、*cmdqm* 属性と同じです。つまり、*qmgrname@host@port@channel* です。

id

必須。取り消す転送の転送 ID を指定します。転送 ID は、転送要求が *fte:filecopy* タスクと *fte:filemove* タスクの両方によって実行依頼される時点で生成されます。

origuser

オプション。 **cancel** 要求に関連付ける発信ユーザー ID を指定します。 *origuser* 属性を使用しない場合、タスクはデフォルトで Ant スクリプトを実行するために使用されるユーザー ID を使用します。

例

次の例では、**fte:cancel** 要求をコマンド・キュー・マネージャー *qm0* に送信します。**fte:cancel** 要求のターゲットは、キュー・マネージャー *qm1* 上の *agent1* で、転送 ID は *transfer.id* 変数によって設定されています。この要求は、"bob" ユーザー ID を使用して実行されます。

```
<fte:cancel cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
  agent="agent1@qm1"
  id="${transfer.id}"
  origuser="bob"/>
```

関連資料

[252 ページの『MFT での Apache Ant の使用』](#)

Managed File Transfer は、ファイル転送機能を Apache Ant ツールに組み込むために使用できるタスクを提供します。

[848 ページの『MFT 提供の Ant タスク』](#)

Managed File Transfer では、数多くの Ant タスクが用意されており、これらのタスクを使用して、ファイル転送機能にアクセスできます。

fte:filecopy の Ant タスク

fte:filecopy タスクは、Managed File Transfer エージェント間でファイルをコピーします。ファイルはソース・エージェントから削除されません。

属性

cmdqm

オプション。要求の実行依頼先のコマンド・キュー・マネージャー。この情報は、`qmgrname@host@port@channel` という形式で指定します。ここで各ストリングは以下のとおりです。

- `qmgrname` は、キュー・マネージャーの名前です。
- `host` は、キュー・マネージャーを実行しているシステムのホスト名です (オプション)。
- `port` は、キュー・マネージャーが listen しているポート番号です (オプション)。
- `channel` は、使用するオプションの SVRCONN チャネルです

コマンド・キュー・マネージャーの `host`、`port`、または `channel` 情報を省略した場合には、`command.properties` ファイルに指定された接続情報が使用されます。詳しくは、[546 ページの『MFT command.properties ファイル』](#)を参照してください。

com.ibm.wmqfte.propertySet プロパティを使用すると、使用する `command.properties` ファイルを指定できます。詳細については、[com.ibm.wmqfte.propertySet](#) を参照してください。

`cmdqm` 属性を使用しなかった場合には、タスクはデフォルトで `com.ibm.wmqfte.ant.commandQueueManager` プロパティを使用します (このプロパティが設定されている場合)。`com.ibm.wmqfte.ant.commandQueueManager` プロパティが設定されていない場合には、`command.properties` ファイルで定義されたデフォルトのキュー・マネージャーへの接続が試行されます。`com.ibm.wmqfte.ant.commandQueueManager` プロパティの形式は、`cmdqm` 属性と同じです。つまり、`qmgrname@host@port@channel` です。

dst

必須。copy 操作の宛先エージェントを指定します。この情報は、`agentname@qmgrname` という形式で指定します (ここで、`agentname` は宛先エージェントの名前、`qmgrname` は、当該エージェントが直接接続されている先のキュー・マネージャーの名前)。

idproperty

`outcome` に `defer` を指定していない場合は、オプションです。転送 ID を割り当てるプロパティの名前を指定します。転送 ID は、転送要求が実行依頼された時点で生成されます。この転送 ID を使用して、転送の進行の追跡、転送で生じた問題の診断、および転送の取り消しを行うことができます。

`outcome` プロパティに `ignore` を指定した場合には、このプロパティは指定できません。ただし、`outcome` プロパティに `defer` を指定した場合には、`idproperty` を指定する必要があります。

jobname

オプション。ジョブ名を copy 要求に割り当てます。ジョブ名を使用して、論理転送グループを作成できます。[861 ページの『fte:uuid の Ant タスク』](#) タスクを使用して、疑似固有ジョブ名を生成します。`jobname` 属性を使用しなかった場合には、タスクはデフォルトで `com.ibm.wmqfte.ant.jobName` プロパティ値を使用します (このプロパティが設定されている場合)。このプロパティが設定されていない場合には、copy 要求に関連付けられるジョブ名はありません。

origuser

オプション。copy 要求に関連付ける発信ユーザー ID を指定します。`origuser` 属性を使用しなかった場合には、タスクはデフォルトで Ant スクリプトを実行するために使用されるユーザー ID を使用します。

outcome

オプション。タスクが、Ant スクリプトに制御を返す前に、copy 操作が完了するまで待機するかどうかを決定します。以下のいずれかのオプションを指定します。

await

タスクは、戻る前に copy 操作が完了するまで待機します。outcome に await が指定されている場合、idproperty 属性はオプションです。

defer

タスクは、copy 要求が実行依頼されるとすぐに戻り、後から [awaitoutcome](#) タスクまたは [860 ページの『fte:ignoreoutcome の Ant タスク』](#) タスクを使用して copy 操作の結果を処理することを想定します。defer の outcome が指定されている場合、idproperty 属性は必須です。

ignore

copy 操作の結果が重要ではない場合には、値 ignore を指定できます。この値を指定した場合、タスクは、転送結果を追跡するためのリソースを割り当てずに、copy 要求が実行依頼されるとすぐに戻ります。ignore の outcome が指定されている場合、idproperty 属性は指定できません。

outcome 属性を指定しなかった場合には、タスクはデフォルトで await を使用します。

priority

オプション。copy 要求に関連付ける優先順位を指定します。一般に、優先順位が高い転送要求が、優先順位が低い要求より優先されます。優先順位の値は、0 以上 9 以下の範囲で指定する必要があります。優先順位値 0 は最低の優先順位であり、値 9 は最高の優先順位です。priority 属性を指定しなかった場合には、転送の優先順位はデフォルトで、0 になります。

rcproperty

オプション。copy 要求の結果コードを割り当てるプロパティの名前を指定します。結果コードには、copy 要求の全体的結果が反映されます。

outcome プロパティに ignore または defer を指定した場合には、このプロパティは指定できません。ただし、await の結果を指定する場合は、rcproperty を指定する必要があります。

V 9.0.1 transferRecoveryTimeout

オプション。停止したファイル転送のリカバリーをソース・エージェントが試行し続ける時間 (秒単位) を設定します。以下のいずれかのオプションを指定します。

-1

エージェントは、停止した転送のリカバリーを、転送が完了するまで試行し続けます。このオプションを使用すると、このプロパティを設定しない場合のエージェントのデフォルトの動作と同じになります。

0

エージェントは、リカバリーに入るとすぐにファイル転送を停止します。

>0

エージェントは、指定された正整数値で設定された時間 (秒単位) だけ、停止した転送のリカバリーを試行し続けます。例:

```
<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>
```

これは、エージェントがリカバリーに入ってから 6 時間にわたって転送のリカバリーを試行し続けることを示しています。この属性の最大値は 999999999 です。

このように指定した場合、転送のリカバリー・タイムアウト値は転送単位で設定されます。Managed File Transfer ネットワーク内のすべての転送が対象になるグローバルな値を設定するには、プロパティ

ーを `agent.properties` ファイルに追加します。詳しくは、[246 ページの『ファイル転送のリカバリーのタイムアウト・オプション』](#)を参照してください。

src

必須。copy 操作のソース・エージェントを指定します。この情報は、`agentname@qmgrname` という形式で指定します (ここで、`agentname` はソース・エージェントの名前、`qmgrname` は、当該エージェントが直接接続されている先のキュー・マネージャーの名前)。

ネスト・エレメントとして指定するパラメーター

fte:filespec

必須。コピーするファイルを識別するファイル指定を少なくとも 1 つ指定する必要があります。必要に応じて複数のファイル指定を指定できます。詳しくは、『[fte:filespec](#)』のトピックを参照してください。

fte:metadata

copy 操作に関連付けるメタデータを指定できます。このメタデータは転送とともに渡され、転送によって生成されたログ・メッセージに記録されます。特定の転送エレメントには、単一のメタデータ・ブロックのみを関連付けることができます。ただし、このブロックには、多くのメタデータを含めることができます。詳しくは、『[fte:metadata](#)』のトピックを参照してください。

fte:presrc

転送開始前にソース・エージェントで行うプログラム呼び出しを指定します。特定の転送に関連付けることができる `fte:presrc` エレメントは 1 つのみです。詳しくは、『[プログラム呼び出し](#)』のトピックを参照してください。

fte:predst

転送開始前に宛先エージェントで行うプログラム呼び出しを指定します。特定の転送に関連付けることができる `fte:predst` エレメントは 1 つのみです。詳しくは、『[プログラム呼び出し](#)』のトピックを参照してください。

fte:postsrc

転送完了後にソース・エージェントで行うプログラム呼び出しを指定します。特定の転送に関連付けることができる `fte:postsrc` エレメントは 1 つのみです。詳しくは、『[プログラム呼び出し](#)』のトピックを参照してください。

fte:postdst

転送完了後に宛先エージェントで行うプログラム呼び出しを指定します。特定の転送に関連付けることができる `fte:postdst` エレメントは 1 つのみです。詳しくは、『[プログラム呼び出し](#)』のトピックを参照してください。

`fte:presrc`、`fte:predst`、`fte:postsrc`、`fte:postdst`、および出口が成功状態に戻さない場合、規則では以下で指定された順序になります。

1. ソース開始出口を実行します。ソース開始出口が失敗すると、転送は失敗し、それ以降何も実行されません。
2. 事前ソース呼び出しを実行します (存在する場合)。事前ソース呼び出しが失敗すると、転送は失敗し、それ以降何も実行されません。
3. 宛先開始出口を実行します。宛先開始出口が失敗すると、転送は失敗し、それ以降何も実行されません。
4. 事前宛先呼び出しを実行します (存在する場合)。事前宛先呼び出しが失敗すると、転送は失敗し、それ以降何も実行されません。
5. ファイル転送を実行します。
6. 宛先終了出口を実行します。これらの出口に失敗状況はありません。
7. 正常に転送された場合 (一部のファイルが正常に転送され、転送が成功したと判断される場合) は、事後宛先呼び出しがあれば、それを実行します。事後宛先呼び出しが失敗すると、転送は失敗します。
8. ソース終了出口を実行します。これらの出口に失敗状況はありません。
9. 正常に転送された場合は、事後ソース呼び出しがあれば、それを実行します。事後ソース呼び出しが失敗すると、転送は失敗します。

例

次の例では、agent1 と agent2 間の基本ファイル転送を示します。ファイル転送を開始するコマンドは、クライアント・トランスポート・モード接続を使用して `qm0`、というキュー・マネージャーに送信されます。ファイル転送操作の結果は、`copy.result` というプロパティに割り当てられています。

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

この例は、同じファイル転送を示していますが、転送の完了後にソース・エージェントでメタデータとプログラムの追加が行われるようになります。

```
<fte:filecopy cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
  src="agent1@qm1" dst="agent2@qm2"
  rcproperty="copy.result">
  <fte:metadata>
    <fte:entry name="org.example.departId" value="ACCOUNTS"/>
    <fte:entry name="org.example.batchGroup" value="A1"/>
  </fte:metadata>
  <fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/file.bin"/>
  <fte:postsrc command="/home/fteuser2/scripts/post.sh" successsrc="1" >
    <fte:arg value="/home/fteuser2/file.bin"/>
  </fte:postsrc>
</fte:filecopy>
```

関連概念

246 ページの『ファイル転送のリカバリーのタイムアウト・オプション』

停止したファイル転送のリカバリーをソース・エージェントが試行し続ける時間を秒単位で設定できます。転送が成功しないままエージェントが再試行間隔のタイムアウトに達した場合、その転送は失敗します。

関連資料

252 ページの『MFT での Apache Ant の使用』

Managed File Transfer は、ファイル転送機能を Apache Ant ツールに組み込むために使用できるタスクを提供します。

848 ページの『MFT 提供の Ant タスク』

Managed File Transfer では、数多くの Ant タスクが用意されており、これらのタスクを使用して、ファイル転送機能にアクセスできます。

fte:filemove の Ant タスク

fte:filemove タスクは、Managed File Transfer エージェント間でファイルを移動します。ファイルがソース・エージェントから宛先エージェントに正常に転送されると、そのファイルはソース・エージェントから削除されます。

属性

cmdqm

オプション。要求の実行依頼先のコマンド・キュー・マネージャー。この情報は、`qmgrname@host@port@channel` という形式で指定します。ここで各ストリングは以下のとおりです。

- `qmgrname` は、キュー・マネージャーの名前です。
- `host` は、キュー・マネージャーを実行しているシステムのホスト名です (オプション)。
- `port` は、キュー・マネージャーが listen しているポート番号です (オプション)。
- `channel` は、使用するオプションの SVRCONN チャネルです

コマンド・キュー・マネージャーの *host*、*port*、または *channel* 情報を省略した場合には、`command.properties` ファイルに指定された接続情報が使用されます。詳しくは、[546 ページの『MFT command.properties ファイル』](#)を参照してください。

com.ibm.wmqfte.propertySet プロパティを使用すると、使用する `command.properties` ファイルを指定できます。詳細については、[com.ibm.wmqfte.propertySet](#) を参照してください。

`cmdqm` 属性を使用しなかった場合には、タスクはデフォルトで `com.ibm.wmqfte.ant.commandQueueManager` プロパティを使用します (このプロパティが設定されている場合)。`com.ibm.wmqfte.ant.commandQueueManager` プロパティが設定されていない場合には、`command.properties` ファイルで定義されたデフォルトのキュー・マネージャーへの接続が試行されます。`com.ibm.wmqfte.ant.commandQueueManager` プロパティの形式は、`cmdqm` 属性と同じです。つまり、`qmgrname@host@port@channel` です。

dst

必須。copy 操作の宛先エージェントを指定します。この情報は、`agentname@qmgrname` という形式で指定します (ここで、`agentname` は宛先エージェントの名前、`qmgrname` は、当該エージェントが直接接続されている先のキュー・マネージャーの名前)。

idproperty

`outcome` に `defer` を指定していない場合は、オプションです。転送 ID を割り当てるプロパティの名前を指定します。転送 ID は、転送要求が実行依頼された時点で生成されます。この転送 ID を使用して、転送の進行の追跡、転送で生じた問題の診断、および転送の取り消しを行うことができます。

`outcome` プロパティに `ignore` を指定した場合には、このプロパティは指定できません。ただし、`outcome` プロパティに `defer` を指定した場合には、`idproperty` を指定する必要があります。

jobname

オプション。ジョブ名を `move` 要求に割り当てます。ジョブ名を使用して、論理転送グループを作成できます。`fte:uuid` タスクを使用して、疑似固有ジョブ名を生成します。`jobname` 属性を使用しなかった場合には、タスクはデフォルトで `com.ibm.wmqfte.ant.jobName` プロパティ値を使用します (このプロパティが設定されている場合)。このプロパティが設定されていない場合には、`move` 要求に関連付けられるジョブ名はありません。

origuser

オプション。`move` 要求に関連付ける発信ユーザー ID を指定します。`origuser` 属性を使用しなかった場合には、タスクはデフォルトで Ant スクリプトを実行するために使用されるユーザー ID を使用します。

outcome

オプション。タスクが、Ant スクリプトに制御を返す前に、`move` 操作が完了するまで待機するかどうかを決定します。以下のいずれかのオプションを指定します。

await

タスクは、戻る前に `move` 操作が完了するまで待機します。`outcome` に `await` が指定されている場合、`idproperty` 属性はオプションです。

defer

タスクは、`move` 要求が実行依頼されるとすぐに戻り、後から [848 ページの『fte:awaitoutcome の Ant タスク』](#) タスクまたは [860 ページの『fte:ignoreoutcome の Ant タスク』](#) タスクを使用して `move` 操作の結果を処理することを想定します。`defer` の `outcome` が指定されている場合、`idproperty` 属性は必須です。

ignore

`move` 操作の結果が重要ではない場合には、値 `ignore` を指定できます。この値を指定した場合、タスクは、転送結果を追跡するためのリソースを割り当てずに、`move` 要求が実行依頼されるとすぐに戻ります。`ignore` の `outcome` が指定されている場合、`idproperty` 属性は指定できません。

`outcome` 属性を指定しなかった場合には、タスクはデフォルトで `await` を使用します。

priority

オプション。move 要求に関連付ける優先順位を指定します。一般に、優先順位が高い転送要求が、優先順位が低い要求より優先されます。優先順位の値は、0 以上 9 以下の範囲で指定する必要があります。優先順位値 0 は最低の優先順位であり、値 9 は最高の優先順位です。priority 属性を指定しなかった場合には、転送の優先順位はデフォルトで、0 になります。

rcproperty

オプション。move 要求の結果コードを割り当てるプロパティの名前を指定します。結果コードには、move 要求の全体的結果が反映されます。

outcome プロパティに ignore または defer を指定した場合には、このプロパティは指定できません。ただし、outcome プロパティに await を指定した場合には、rcproperty を指定する必要があります。

V 9.0.1 transferRecoveryTimeout

オプション。停止したファイル転送のリカバリーをソース・エージェントが試行し続ける時間 (秒単位) を設定します。以下のいずれかのオプションを指定します。

-1

エージェントは、停止した転送のリカバリーを、転送が完了するまで試行し続けます。このオプションを使用すると、このプロパティを設定しない場合のエージェントのデフォルトの動作と同じになります。

0

エージェントは、リカバリーに入るとすぐにファイル転送を停止します。

>0

エージェントは、指定された正整数値で設定された時間 (秒単位) だけ、停止した転送のリカバリーを試行し続けます。例:

```
<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
```

これは、エージェントがリカバリーに入ってから 6 時間にわたって転送のリカバリーを試行し続けることを示しています。この属性の最大値は 999999999 です。

このように指定した場合、転送のリカバリー・タイムアウト値は転送単位で設定されます。Managed File Transfer ネットワーク内のすべての転送が対象になるグローバルな値を設定するには、プロパティを agent.properties ファイルに追加します。詳しくは、[246 ページの『ファイル転送のリカバリーのタイムアウト・オプション』](#)を参照してください。

src

必須。move 操作のソース・エージェントを指定します。この情報は、agentname@qmgrname という形式で指定します (ここで、agentname はソース・エージェントの名前、qmgrname は、当該エージェントが直接接続されている先のキュー・マネージャーの名前)。

ネスト・エレメントとして指定するパラメーター

fte:filespec

必須。移動対象ファイルを特定する少なくとも 1 つのファイル指定を指定する必要があります。必要に応じて複数のファイル指定を指定できます。詳しくは、『fte:filespec』のトピックを参照してください。

fte:metadata

オプション。ファイルの move 操作に関連付けるメタデータを指定できます。このメタデータは転送とともに渡され、転送によって生成されたログ・メッセージに記録されます。特定の転送エレメントには、単一のメタデータ・ブロックのみを関連付けることができます。ただし、このブロックには、多くのメタデータを含めることができます。詳しくは、『fte:metadata』のトピックを参照してください。

fte:presrc

オプション。転送開始前にソース・エージェントで行うプログラム呼び出しを指定します。特定の転送に関連付けることができる **fte:presrc** エレメントは1つのみです。詳しくは、『[プログラム呼び出し](#)』のトピックを参照してください。

fte:predst

オプション。転送開始前に宛先エージェントで行うプログラム呼び出しを指定します。特定の転送に関連付けることができる **fte:predst** エレメントは1つのみです。詳しくは、『[プログラム呼び出し](#)』のトピックを参照してください。

fte:postsrc

オプション。転送完了後にソース・エージェントで行うプログラム呼び出しを指定します。特定の転送に関連付けることができる **fte:postsrc** エレメントは1つのみです。詳しくは、『[プログラム呼び出し](#)』のトピックを参照してください。

fte:postdst

オプション。転送完了後に宛先エージェントで行うプログラム呼び出しを指定します。特定の転送に関連付けることができる **fte:postdst** エレメントは1つのみです。詳しくは、『[プログラム呼び出し](#)』のトピックを参照してください。

fte:presrc、**fte:predst**、**fte:postsrc**、**fte:postdst**、および出口が成功状態に戻さない場合、規則では以下で指定された順序になります。

1. ソース開始出口を実行します。ソース開始出口が失敗すると、転送は失敗し、それ以降何も実行されません。
2. 事前ソース呼び出しを実行します (存在する場合)。事前ソース呼び出しが失敗すると、転送は失敗し、それ以降何も実行されません。
3. 宛先開始出口を実行します。宛先開始出口が失敗すると、転送は失敗し、それ以降何も実行されません。
4. 事前宛先呼び出しを実行します (存在する場合)。事前宛先呼び出しが失敗すると、転送は失敗し、それ以降何も実行されません。
5. ファイル転送を実行します。
6. 宛先終了出口を実行します。これらの出口に失敗状況はありません。
7. 正常に転送された場合 (一部のファイルが正常に転送され、転送が成功したと判断される場合) は、事後宛先呼び出しがあれば、それを実行します。事後宛先呼び出しが失敗すると、転送は失敗します。
8. ソース終了出口を実行します。これらの出口に失敗状況はありません。
9. 正常に転送された場合は、事後ソース呼び出しがあれば、それを実行します。事後ソース呼び出しが失敗すると、転送は失敗します。

例

次の例では、**agent1** と **agent2** 間の基本ファイル移動を示します。ファイル移動を開始するコマンドは、クライアント・トランスポート・モード接続を使用して **qm0**、というキュー・マネージャーに送信されます。ファイル転送操作の結果は、**move.result** というプロパティに割り当てられています。

```
<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>
```

関連概念

[246 ページの『ファイル転送のリカバリーのタイムアウト・オプション』](#)

停止したファイル転送のリカバリーをソース・エージェントが試行し続ける時間を秒単位で設定できます。転送が成功しないままエージェントが再試行間隔のタイムアウトに達した場合、その転送は失敗します。

関連資料

[252 ページの『MFT での Apache Ant の使用』](#)

Managed File Transfer は、ファイル転送機能を Apache Ant ツールに組み込むために使用できるタスクを提供します。

[848 ページの『MFT 提供の Ant タスク』](#)

Managed File Transfer では、数多くの Ant タスクが用意されており、これらのタスクを使用して、ファイル転送機能にアクセスできます。

fte:ignoreoutcome の Ant タスク

fte:filecopy、**fte:filemove**、または **fte:call** コマンドの結果を無視します。 **fte:filecopy**、**fte:filemove**、または **fte:call** タスクで `defer` の結果を得るように指定すると、Ant タスクでこの結果を追跡するためのリソースが割り振られます。この結果がもはや必要ない場合は、**fte:ignoreoutcome** タスクでこうしたリソースを解放することができます。

属性

id

必須。もはや必要ない出力を識別します。通常、この ID は、[853 ページの『fte:filecopy の Ant タスク』](#)、[856 ページの『fte:filemove の Ant タスク』](#)、または [849 ページの『fte:call の Ant タスク』](#) タスクの `idproperty` 属性を使用して設定するプロパティを使用して指定します。

例

この例では、`fte:ignoreoutcome` タスクを使用して、以前の [853 ページの『fte:filecopy の Ant タスク』](#) タスクの出力を追跡するために割り振られていたリソースをどのように解放できるかを示します。

```
<!-- issue a file copy request -->
<fte:filecopy cmdqmqm="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}"/>
```

関連資料

[252 ページの『MFT での Apache Ant の使用』](#)

Managed File Transfer は、ファイル転送機能を Apache Ant ツールに組み込むために使用できるタスクを提供します。

[848 ページの『MFT 提供の Ant タスク』](#)

Managed File Transfer では、数多くの Ant タスクが用意されており、これらのタスクを使用して、ファイル転送機能にアクセスできます。

fte:ping の Ant タスク

この IBM MQ Managed File Transfer Ant タスクでは、エージェントに ping を送信して応答を引き出し、そのエージェントが転送を処理できるかどうかを確認します。

属性

agent

必須。 **fte:ping** 要求の送信先のエージェントを指定します。この値は `agentname@qmgrname` の形式です。ここで `agentname` はエージェントの名前で、`qmgrname` は、このエージェントが直接接続しているキュー・マネージャーの名前です。

cmdqmqm

オプション。要求の実行依頼先のコマンド・キュー・マネージャー。この情報は、`qmgrname@host@port@channel` という形式で指定します。ここで各ストリングは以下のとおりです。

- `qmgrname` は、キュー・マネージャーの名前です。
- `host` は、キュー・マネージャーを実行しているシステムのホスト名です (オプション)。
- `port` は、キュー・マネージャーが `listen` しているポート番号です (オプション)。
- `channel` は、使用するオプションの `SVRCONN` チャンネルです

コマンド・キュー・マネージャーの `host`、`port`、または `channel` 情報を省略した場合には、`command.properties` ファイルに指定された接続情報が使用されます。詳しくは、[546 ページの『MFT command.properties ファイル』](#)を参照してください。

`com.ibm.wmqfte.propertySet` プロパティを使用すると、使用する `command.properties` ファイルを指定できます。詳細については、[com.ibm.wmqfte.propertySet](#) を参照してください。

`cmdqm` 属性を使用しなかった場合には、タスクはデフォルトで `com.ibm.wmqfte.ant.commandQueueManager` プロパティを使用します (このプロパティが設定されている場合)。`com.ibm.wmqfte.ant.commandQueueManager` プロパティが設定されていない場合には、`command.properties` ファイルで定義されたデフォルトのキュー・マネージャーへの接続が試行されます。`com.ibm.wmqfte.ant.commandQueueManager` プロパティの形式は、`cmdqm` 属性と同じです。つまり、`qmgrname@host@port@channel` です。

rcproperty

必須。ping 操作の戻りコードを保管するためのプロパティの名前を指定します。

timeout

オプション。タスクがエージェントの応答を待つ最大時間 (秒単位)。最小タイムアウトは 0 秒ですが、エージェントが応答するまでコマンドは永久に待つようにする場合はタイムアウトにマイナス 1 を指定することもできます。`timeout` に値が指定されない場合、デフォルトで、エージェントの応答を 5 秒まで待ちます。

例

この例では、**fte:ping** 要求を `qm1` によってホスティングされる `agent1` に送信します。**fte:ping** 要求では、エージェントの応答を 15 秒間待機します。**fte:ping** 要求の結果は、`ping.rc` という名前のプロパティに保管されます。

```
<fte:ping agent="agent1@qm1" rcproperty="ping.rc" timeout="15"/>
```

戻りコード

0

コマンドは正常に完了しました。

2

コマンドはタイムアウトになりました。

関連資料

[252 ページの『MFT での Apache Ant の使用』](#)

Managed File Transfer は、ファイル転送機能を Apache Ant ツールに組み込むために使用できるタスクを提供します。

[848 ページの『MFT 提供の Ant タスク』](#)

Managed File Transfer では、数多くの Ant タスクが用意されており、これらのタスクを使用して、ファイル転送機能にアクセスできます。

fte:uuid の Ant タスク

疑似乱数固有 ID を生成して、所定のプロパティに割り当てます。例えば、この ID を使用して、その他のファイル転送操作のジョブ名を生成できます。

属性

length

必須。生成する UUID の長さを示す数値。この長さの値には、**prefix** パラメーターで指定される接頭部の長さは含まれません。

property

必須。生成された UUID を割り当てるプロパティの名前。

prefix

オプション。生成された UUID に付加する接頭部。この接頭部は、**length** パラメーターで指定する UUID の長さには含まれません。

例

この例では、ABC という文字で始まり、16 個の疑似ランダム 16 進文字が続く UUID を定義します。この UUID は、`uuid.property` という名前のプロパティに割り当てられます。

```
<fte:uuid length="16" property="uuid.property" prefix="ABC"/>
```

関連資料

252 ページの『MFT での Apache Ant の使用』

Managed File Transfer は、ファイル転送機能を Apache Ant ツールに組み込むために使用できるタスクを提供します。

848 ページの『MFT 提供の Ant タスク』

Managed File Transfer では、数多くの Ant タスクが用意されており、これらのタスクを使用して、ファイル転送機能にアクセスできます。

fte:filespec Ant のネストされたエレメント

fte:filespec パラメーターは、その他のタスクでネストされたエレメントとして使用されます。

fte:filespec を使用すると、1 つ以上のソース・ファイル、ディレクトリー  またはデータ・セット、および宛先とのマッピングを記述できます。一般的には、このエレメントは、移動またはコピーする一連のファイル、ディレクトリー 、またはデータ・セットを表す場合に使用されます。

ネスト対象タスク:

- [fte:filecopy](#) タスク
- [fte:filemove](#) タスク

ソース指定属性

`srcfilespec` または `srcqueue` のいずれかを指定する必要があります。

srcfilespec

ファイル操作のソースを指定します。この属性の値は、ワイルドカードを含むことができます。

srcqueue

転送のソースがキューであることを指定します。転送すると、この属性によって指定されたキューに保管されているメッセージからデータが移動します。**fte:filespec** タスクが **fte:filecopy** タスク内にネストされている場合は、この属性を指定できません。

`srcqueue` 属性は、ソース・エージェントがプロトコル・ブリッジ・エージェントである場合はサポートされません。

宛先指定属性

`dstdir`、`dstds`、`dstfilespace`、`dstfile`、`dstqueue`、または `dstpds` のいずれかを指定する必要があります。

dstdir

ファイル操作の宛先としてディレクトリーを指定します。

dstds

ファイル操作の宛先としてデータ・セットを指定します。

この属性がサポートされるのは、宛先エージェントが z/OS プラットフォームで実行中の場合のみです。

dstfile

ファイル操作の宛先としてファイルを指定します。

dstfilespace

ファイル操作の宛先としてファイル・スペースを指定します。

この属性が適用されるのは、宛先エージェントが、Web ゲートウェイ・ファイル・スペースに対するアクセス権を持つ IBM MQ 8.0 Web エージェントである場合のみです。

dstpds

ファイル操作の宛先として区分データ・セットを指定します。

この属性がサポートされるのは、宛先エージェントが z/OS プラットフォームで実行中の場合のみです。

dstqueue

ファイルからメッセージ操作の宛先としてキューを指定します。この指定に QUEUE@QUEUEMANAGER というフォーマットでキュー・マネージャー名を含めることもできます。キュー・マネージャーを指定しない場合は、enableClusterQueueInputOutput エージェント・プロパティを true に設定しない限り、宛先エージェントのキュー・マネージャーが使用されます。enableClusterQueueInputOutput プロパティが true に設定されている場合、宛先エージェントは、標準的な IBM MQ 手順を使用して、キューが配置されている場所を判別します。対象のキュー・マネージャーに存在する有効なキュー名を指定する必要があります。

dstqueue 属性を指定する場合、srcqueue 属性は指定できません。これらの属性は相互に排他的です。

dstqueue 属性は、宛先エージェントがプロトコル・ブリッジ・エージェントである場合はサポートされません。

ソース・オプション属性

srcencoding

オプション。転送するファイルで使用される文字セット・エンコード方式。

この属性を指定できるのは、変換属性の値が text. に設定されている場合のみです。

srcencoding 属性を指定しない場合、テキスト転送にはソース・システムの文字セットが使用されます。

srceol

オプション。転送されるファイルで使用される行の終わり区切り文字。有効な値は以下のとおりです。

- CRLF - 行の終わり区切り文字として、改行文字が後に続く復帰文字が使用されます。この規則は、Windows システムの場合の標準です。
- LF - 行末区切り文字として改行文字を使用します。この規則は、UNIX システムの場合の標準です。

この属性を指定できるのは、conversion 属性が値 text に設定されている場合のみです。srceol 属性を指定しない場合、ソース・エージェントのオペレーティング・システムに基づいて、テキスト転送で正しい値が自動的に決定されます。

srckeeptrailingspaces

オプション。テキスト・モード転送の一部として固定長形式のデータ・セットから読み取られるソース・レコードの末尾スペースを、保持するかどうかを決定します。有効な値は以下のとおりです。

- true - 末尾スペースは保持されます。

- `false` - 末尾スペースは削除されます。

`srckeeptrailingspaces` 属性を指定しない場合、デフォルト値の `false` が指定されます。

この属性を指定できるのは、`srcfilespec` 属性も指定し、変換属性の値を `text.` に設定した場合のみです。

srcmsgdelimbytes

オプション。1つのバイナリー・ファイルに複数のメッセージを追加するときに区切り文字として挿入する1つ以上のバイト値を指定します。それぞれの値は、`x`という接頭部を付けた00からFFの範囲の2桁の16進数字として指定する必要があります。複数バイトの場合はコンマで区切る必要があります。例えば、`srcmsgdelimbytes="x08,xA4"`のように指定します。`srcmsgdelimbytes` 属性を指定できるのは、`srcqueue` 属性も指定した場合に限られます。`conversion` 属性の値として `text` を指定した場合は、`srcmsgdelimbytes` 属性を指定できません。

srcmsgdelimtext

オプション。1つのテキスト・ファイルに複数のメッセージを追加するときに区切り文字として挿入するテキストのシーケンスを指定します。ストリング・リテラルのJava エスケープ・シーケンスを区切り文字に含めることもできます。例えば、`srcmsgdelimtext="\u007d\n"` です。ソース・エージェントによって各メッセージの後にテキスト区切り文字が挿入されます。テキスト区切り文字は、転送のソース・エンコード方式に基づいてバイナリー・フォーマットにエンコードされます。各メッセージがバイナリー・フォーマットで読み取られ、エンコードされた区切り文字がバイナリー・フォーマットでメッセージに追加され、結果がバイナリー・フォーマットで宛先エージェントに転送されます。ソース・エージェントのコード・ページにシフトイン状態とシフトアウト状態が含まれていれば、エージェントは、各メッセージがメッセージの末尾でシフトアウト状態になると想定します。宛先エージェントでは、ファイルからファイルへのテキスト転送の場合と同じ要領でバイナリー・データが変換されます。`srcmsgdelimtext` 属性を指定できるのは、`srcqueue` 属性も指定し、`conversion` 属性の値として `text` を指定した場合に限られます。

srcmsgdelimposition

オプション。テキストまたはバイナリー区切り文字が挿入される位置を指定します。有効な値は以下のとおりです。

- `prefix` - 区切り文字は、宛先ファイルの各メッセージのデータの前に挿入されます。
- `postfix` - 区切り文字は、宛先ファイルの各メッセージのデータの後に挿入されます。

`srcmsgdelimposition` 属性を指定できるのは、`srcmsgdelimbytes` または `srcmsgdelimtext` 属性のいずれかをも指定した場合に限られます。

srcmsggroups

オプション。メッセージを IBM MQ グループ ID によってグループ化する動作を指定します。完全に揃った最初のグループが宛先ファイルに書き込まれます。この属性を指定しない場合は、ソース・キューに存在するすべてのメッセージが宛先ファイルに書き込まれます。`srcmsggroups` 属性を指定できるのは、`srcqueue` 属性も指定した場合に限られます。

srcqueuetimeout

オプション。以下のいずれかの条件が満たされるのを待つ時間を秒単位で指定します。

- 新しいメッセージがキューに書き込まれるという条件。
- `srcmsggroups` 属性を指定した場合は、完全に揃ったグループがキューに書き込まれるという条件。

`srcqueuetimeout` の値で指定した時間内にどちらの条件も満たされなければ、ソース・エージェントは、キューからの読み取りを停止して、転送を完了します。`srcqueuetimeout` 属性を指定しなければ、ソース・エージェントは、ソース・キューが空の場合にソース・キューからの読み取りをただちに停止します。あるいは、`srcmsggroups` 属性が指定されているのであれば、完全に揃ったグループがキューに存在しない場合にソース・キューからの読み取りをただちに停止します。`srcqueuetimeout` 属性を指定できるのは、`srcqueue` 属性も指定した場合に限られます。

`srcqueuetimeout` 値の設定については、[682 ページ](#)の『メッセージからファイルへの転送の待機時間を指定する際のガイダンス』を参照してください。

srcrcdelimbytes

オプション。1つのバイナリー・ファイルにレコード単位のソース・ファイルから複数のレコードを追加するときに区切り文字として挿入する、1つ以上のバイト値を指定します。それぞれの値は、接頭部 *x* を付けた 00 から FF の範囲の 2 桁の 16 進数字として指定する必要があります。複数バイトの場合はコンマで区切る必要があります。以下に例を示します。

```
srcrcdelimbytes="x08,xA4"
```

`srcrcdelimbytes` 属性は、転送のソース・ファイルがレコード単位 (例えば z/OS データ・セットなど) で、かつ、宛先ファイルがレコード単位ではない通常のファイルの場合にのみ指定できます。`conversion` 属性の値として `text` を指定した場合は、`srcrcdelimbytes` 属性を指定できません。

srcrcdelimpos

オプション。バイナリー区切り文字が挿入される位置を指定します。有効な値は以下のとおりです。

- `prefix` - 区切り文字は、宛先ファイルの、ソースのレコード単位ファイルの各レコードからのデータの前に挿入されます。
- `postfix` - 区切り文字は、宛先ファイルの、ソースのレコード単位ファイルの各レコードからのデータの後に挿入されます。

`srcrcdelimpos` 属性を指定できるのは、`srcrcdelimbytes` 属性も指定した場合に限られます。

宛先オプション属性**dstencoding**

オプション。転送されるファイルに使用する文字セット・エンコード方式。

この属性を指定できるのは、変換属性の値が `text.` に設定されている場合のみです。

`dstencoding` 属性を指定しない場合、テキスト転送には宛先システムの文字セットが使用されます。

dsteol

オプション。転送されるファイルに使用する行の終わり区切り文字。有効な値は以下のとおりです。

- `CRLF` - 行の終わり区切り文字として、改行文字が後に続く復帰文字が使用されます。この規則は、Windows システムの場合の標準です。
- `LF` - 行末区切り文字として改行文字を使用します。この規則は、UNIX システムの場合の標準です。

この属性を指定できるのは、変換属性の値が `text.` に設定されている場合のみです。

`dsteol` 属性を指定しない場合、宛先エージェントのオペレーティング・システムに基づいて、テキスト転送で正しい値が自動的に決定されます。

dstmsgdelimbytes

オプション。バイナリー・ファイルを複数のメッセージに分割するときに使用する 16 進数区切り文字を指定します。メッセージは、すべて同じ IBM MQ グループ ID を持ちます。グループの最後のメッセージは IBM MQ `LAST_MSG_IN_GROUP` フラグ・セットを持ちます。区切り文字として 16 進数バイトを指定するためのフォーマットは、`xNN` です (*N* は、0-9 または a-f の範囲の文字です)。区切り文字として一連の 16 進数バイトを指定することもできます。その場合は、16 進数バイトのコンマ区切りリストを指定します (例: `x3e,x20,x20,xbf`)。

`dstmsgdelimbytes` 属性を指定できるのは、バイナリー・モードの転送で `dstqueue` 属性も指定した場合に限られます。`dstmsgsize`、`dstmsgdelimbytes`、および `dstmsgdelimpattern` 属性のいずれか 1 つのみを指定できます。

dstmsgdelimpattern

オプション。テキスト・ファイルを複数のメッセージに分割するときに使用する Java 正規表現を指定します。メッセージは、すべて同じ IBM MQ グループ ID を持ちます。グループの最後のメッセージは IBM MQ `LAST_MSG_IN_GROUP` フラグ・セットを持ちます。区切りとして正規表現を指定するためのフォーマットは、`(regular_expression)` (正規表現を括弧で囲んだ形式) または

"*regular_expression*" (正規表現を二重引用符で囲んだ形式) です。詳しくは、[659 ページの『MFT が使用する正規表現』](#)を参照してください。

デフォルトでは、正規表現にマッチング可能なストリングの長さは、宛先エージェントによって 5 文字に制限されています。この動作を変更するには、`maxDelimiterMatchLength` エージェント・プロパティを使用します。詳しくは、[551 ページの『拡張エージェント・プロパティ』](#)を参照してください。

`dstmsgdelimpattern` 属性を指定できるのは、テキスト・モードの転送で `dstqueue` 属性も指定した場合に限られます。`dstmsgsize`、`dstmsgdelimbytes`、および `dstmsgdelimpattern` 属性のいずれか 1 つのみを指定できます。

dstmsgdelimposition

オプション。テキストまたはバイナリー区切り文字が入ると想定される位置を指定します。有効な値は以下のとおりです。

- `prefix` - 区切り文字は各行の先頭にあることが想定されています。
- `postfix` - 区切り文字は各行の末尾にあることが想定されています。

`dstmsgdelimposition` 属性を指定できるのは、`dstmsgdelimpattern` 属性も指定した場合に限られます。

dstmsgincludedelim

オプション。ファイルを複数のメッセージに分割するために使用する区切り文字をそれらのメッセージに組み込むかどうかを指定します。`dstmsgincludedelim` 属性を指定すると、区切り文字の前にあるファイル・データが含まれているメッセージの末尾に区切り文字が組み込まれます。デフォルトでは、メッセージに区切り文字は組み込まれません。`dstmsgincludedelim` 属性を指定できるのは、`dstmsgdelimpattern` および `dstmsgdelimbytes` 属性のいずれかをも指定した場合に限られます。

dstmsgpersist

オプション。宛先キューに書き込むメッセージを永続メッセージにするかどうかを指定します。有効な値は以下のとおりです。

- `true` - 宛先キューに永続メッセージを書き込みます。これはデフォルト値です。
- `false` - 宛先キューに非永続メッセージを書き込みます。
- `qdef` - 永続性の値を宛先キューの `DefPersistence` 属性から取り込みます。

この属性を指定できるのは、`dstqueue` 属性も指定されている場合のみです。

dstmsgprops

オプション。転送で宛先キューに書き込む最初のメッセージで IBM MQ メッセージ・プロパティを設定するかどうかを指定します。指定可能な値は以下のとおりです。

- `true` - 転送で作成する最初のメッセージでメッセージ・プロパティを設定します。
- `false` - 転送で作成する最初のメッセージでメッセージ・プロパティを設定しません。これはデフォルト値です。

詳しくは、[677 ページの『MFT が宛先キューに書き込むメッセージで設定する MQ メッセージ・プロパティ』](#)を参照してください。

この属性を指定できるのは、`dstqueue` 属性も指定されている場合のみです。

dstmsgsize

オプション。ファイルを複数の固定長メッセージに分割するかどうかを指定します。すべてのメッセージは、同じ IBM MQ グループ ID を持ちます。グループの最後のメッセージには IBM MQ `LAST_MSG_IN_GROUP` フラグが設定されます。メッセージのサイズは、`dstmsgsize` の値で指定します。`dstmsgsize` のフォーマットは、`lengthunits` です (`length` は正整数値、`units` は以下のいずれかの値です)。

- `B` - バイト。指定できる最小値は、宛先メッセージのコード・ページの 1 文字あたりの最大バイト数の値の 2 倍です。
- `K` - キビバイト。1024 バイトに相当します。
- `M` - メビバイト。1024 キビバイトに相当します。

ファイルをテキスト・モードで転送していて、ファイルが2バイト文字セットまたはマルチバイト文字セットのファイルである場合、そのファイルは、指定のメッセージ・サイズに最も近い文字境界で複数のメッセージに分割されます。

dstmsgsize 属性を指定できるのは、dstqueue 属性も指定した場合に限られます。dstmsgsize、dstmsgdelimbytes、および dstmsgdelimpattern 属性のいずれか1つのみを指定できます。

dstunsupportedcodepage

オプション。dstqueue 属性で指定された宛先キュー・マネージャーが、テキスト転送としてファイル・データをキューに転送する際に使用されるコード・ページをサポートしていない場合に実行するアクションを指定します。この属性で有効な値は以下のとおりです。

- **binary** - 転送を続行しますが、転送中のデータにコード・ページ変換を適用しません。この値の指定は、変換属性を **text** に設定しないことと同等です。
- **fail** - 転送操作を続行しません。ファイルは転送に失敗したものとして記録されます。これがデフォルトです。

dstunsupportedcodepage 属性を指定できるのは、dstqueue 属性も指定し、conversion 属性の値として **text** を指定した場合に限られます。

dsttruncaterecords

オプション。LRECL データ・セット属性よりも長い宛先レコードが切り捨てられることを指定します。true に設定すると、それらのレコードは切り捨てられます。false に設定すると、それらのレコードは折り返されます。デフォルト設定は **false** です。このパラメーターは、宛先がデータ・セットであるテキスト・モードの転送のみに有効です。

その他の属性

checksum

オプション。転送されたファイルのチェックサムを計算するために使用されるアルゴリズムを決定します。

- **MD5** - MD5 ハッシュ・アルゴリズムが使用されます。
- **NONE** - チェックサム・アルゴリズムは使用されません。

checksum 属性を指定しない場合、デフォルト値の **MD5** が使用されます。

conversion

オプション。ファイル転送の際にファイルに適用される変換のタイプを指定します。指定可能な値は以下のとおりです。

- **binary** - 変換は適用されません。
- **text** - ソース・システムと宛先システムの間でコード・ページ変換が適用されます。行区切り文字も変換されます。適用される変換には、srcencoding、dstencoding、srceol、および dsteol 属性が影響を及ぼします。

conversion 属性を指定しない場合、デフォルト値の **binary** が指定されます。

overwrite

オプション。操作で既存の宛先ファイル  またはデータ・セットを上書きできるかどうかを決定します。値 **true** を指定すると、既存の宛先ファイル  またはデータ・セットはすべて上書きされます。値 **false** を指定した場合、宛先に重複するファイル  またはデータ・セットが存在していると、操作は失敗することになります。overwrite 属性を指定しない場合、デフォルト値の **false** が指定されます。

recurse

オプション。サブディレクトリーでファイル転送を繰り返すかどうかを決定します。値 **true** を指定すると、サブディレクトリーで転送を繰り返します。値 **false** を指定した場合、サブディレクトリーで転送を繰り返しません。recurse 属性を指定しない場合、デフォルト値の **false** が指定されます。

例

この例では、file1.bin のソース・ファイルと file2.bin. の宛先ファイルを指定して fte:filespec を指定します。

```
<fte:filespec srcfilespec="/home/fteuser/file1.bin" dstfile="/home/fteuser/file2.bin"/>
```

関連資料

252 ページの『MFT での Apache Ant の使用』

Managed File Transfer は、ファイル転送機能を Apache Ant ツールに組み込むために使用できるタスクを提供します。

848 ページの『MFT 提供の Ant タスク』

Managed File Transfer では、数多くの Ant タスクが用意されており、これらのタスクを使用して、ファイル転送機能にアクセスできます。

fte:metadata Ant nested element

メタデータは、ファイル転送操作で、追加のユーザー定義情報を渡す場合に使用します。

Managed File Transfer でのメタデータの使用法について詳しくは、『872 ページの『MFT ユーザー出口のメタデータ』』を参照してください。

ネスト対象タスク:

- [fte:filecopy](#) タスク
- [fte:filemove](#) タスク
- [fte:call](#) タスク

ネスト・エレメントとして指定するパラメーター

fte:entry

fte:metadata ネストされた要素内に少なくとも 1 つの項目を指定する必要があります。複数の entry を指定することもできます。entry はキー名を値に関連付けます。キーは fte:metadata のブロック内で固有でなければなりません。

entry の属性

name

必須。当該 entry に所属するキーの名前。この名前は、fte:metadata・エレメント内にネストされたすべての **entry** パラメーターにわたって固有である必要があります。

value

必須。当該 entry に割り当てる値。

例

この例は、2 つの項目を含む fte:metadata 定義を示しています

```
<fte:metadata>
  <fte:entry name="org.foo.partColor" value="red"/>
  <fte:entry name="org.foo.partSize" value="medium"/>
</fte:metadata>
```

関連資料

252 ページの『MFT での Apache Ant の使用』

Managed File Transfer は、ファイル転送機能を Apache Ant ツールに組み込むために使用できるタスクを提供します。

848 ページの『MFT 提供の Ant タスク』

Managed File Transfer では、数多くの Ant タスクが用意されており、これらのタスクを使用して、ファイル転送機能にアクセスできます。

プログラム呼び出しのネスト・エレメント

プログラムは、5つのネストされたエレメント、`fte:presrc`、`fte:predst`、`fte:postdst`、`fte:postsrc`、および `fte:command` の1つを使用して開始できます。これらのネストされたエレメントは、エージェントにその処理の一部で外部プログラムを呼び出すように指示します。プログラムを開始する前に、コマンドを実行するエージェントの `agent.properties` ファイルで `commandPath` プロパティにより指定された場所にコマンドがあることを確認する必要があります。

プログラム呼び出しの各エレメントは異なる名前を持っていても、同じ属性セットおよび同じネストされたエレメントのセットを共有します。プログラムは、**`fte:filecopy`**、**`fte:filemove`**、および **`fte:command`** Ant タスクによって開始できます。

Connect:Direct ブリッジ・エージェントからプログラムを呼び出すことはできません。

プログラムを呼び出すことのできる Ant タスクは、以下のとおりです。

- `fte:filecopy` タスクは、`fte:predst`、`fte:postdst`、`fte:presrc`、および `fte:postsrc` というネストされたエレメントを使用して、プログラム呼び出しパラメーターをネストします。
- `fte:filemove` タスクは、`fte:predst`、`fte:postdst`、`fte:presrc`、および `fte:postsrc` というネストされたエレメントを使用して、プログラム呼び出しパラメーターをネストします。
- `fte:call` タスクは、`fte:command` というネストされたエレメントを使用して、プログラム呼び出しパラメーターをネストします。

属性

`command`

必須。呼び出すプログラムの名前を示します。エージェントがコマンドを呼び出すことができるようにするには、そのコマンドを、エージェントの `agent.properties` ファイル内の `commandPath` プロパティで指定された場所に入れておく必要があります。詳しくは、365 ページの『[commandPath MFT プロパティ](#)』を参照してください。 `command` 属性に指定されるパス情報は、`commandPath` プロパティによって指定される場所に対する相対パスであるとみなされます。 `type` が `executable` である場合、実行可能プログラムが想定され、そうでない場合は、その呼び出しタイプに適したスクリプトが想定されます。

`retrycount`

オプション。プログラムが成功を示す戻りコードを戻さなかった場合に、プログラムの呼び出しを再試行する回数。 `command` 属性で指定されたプログラムは、この回数まで呼び出されます。この属性に指定する値は、負数以外でなければなりません。 `retrycount` 属性を指定しない場合、デフォルト値のゼロが使用されます。

`retrywait`

オプション。プログラム呼び出しを再度試行するまでの待機時間 (秒数)。 `command` 属性で指定されたプログラムが成功を示す戻りコードを戻さず、 `retrycount` 属性にゼロ以外の値が指定されている場合、このパラメーターにより再試行間で待機する時間が決定されます。この属性に指定する値は、負数以外でなければなりません。 `retrywait` 属性を指定しない場合、デフォルト値のゼロが使用されます。

`successrc`

オプション。この属性の値は、プログラム呼び出しが正常に実行される条件を決定するために使用されます。コマンドの処理戻りコードは、この式を使用して評価されます。この値は、1つ以上の式を組み合わせ、ブールの OR を意味する垂直バー文字 (|)、またはブールの AND を意味するアンパサンド文字 (&) で構成することができます。各式は、以下のいずれかのタイプの式とすることができます。

- 処理戻りコードとの等価テストを示す数値。
- 処理戻りコードとの大なりテストを示す、接頭部に ">" 文字が付いた数値。

- 数値とプロセスの戻りコードとの間のより小さいテストを示すために、先頭に "<" 文字が付いた数値です。
- "!" の接頭部が付いた数値 数値とプロセスの戻りコードとの間の不等号テストを示す文字。

例えば、>2&<7&!5|0|14 は、成功した戻りコードが 0、3、4、6、14 であると解釈されます。これ以外の戻りコードは、すべて失敗と解釈されます。successrc 属性を指定しない場合、デフォルト値のゼロが使用されます。これは、ゼロの戻りコードを戻した場合にのみ、コマンドは正常に実行されたと判断されるという意味です。

タイプ

オプション。この属性の値は、呼び出されているプログラムのタイプを指定します。以下のいずれかのオプションを指定します。

executable

タスクは、実行可能プログラムを呼び出します。arg ネスト・エレメントを使用して追加の引数を指定することができます。プログラムは、commandPath 上、および実行許可が設定されている適切な場所においてアクセス可能であると想定されます。UNIX スクリプトは、シェル・プログラムを指定する限り呼び出すことができます(例えば、シェル・スクリプト・ファイルの最初の行は #!/bin/sh です)。stderr または stdout に書き込まれるコマンド出力は、その呼び出しの Managed File Transfer ログに送信されます。ただし、そのデータ出力量は、エージェントの構成によって制限されます。デフォルトでは、10K バイトのデータまでですが、エージェントのプロパティー maxCommandOutput を使用してこのデフォルトを指定変更することができます。

antscript

タスクは、**fteAnt** コマンドを使用して、指定された Ant スクリプトを実行します。プロパティーは、property ネスト・エレメントを使用して指定できます。Ant ターゲットは target ネスト・エレメントを使用して指定できます。Ant スクリプトは、commandPath 上でアクセス可能と想定されます。stderr または stdout に書き込まれる Ant 出力は、その呼び出しに対する Managed File Transfer ログに送信されます。ただし、そのデータ出力量は、エージェントの構成によって制限されます。デフォルトでは、10K バイトのデータまでですが、エージェントのプロパティー maxCommandOutput を使用してこのデフォルトを指定変更することができます。

z/OS jcl

値 jcl は z/OS でのみサポートされ、指定された z/OS JCL スクリプトを実行します。JCL はジョブとして実行依頼され、ジョブ・カードがあることが必要です。ジョブが正常に実行依頼されると、Managed File Transfer ログに書き込まれる JCL コマンド出力に、JOB *job_name(job_id)* というテキストが入ります。ここで、

- *job_name* は、JCL のジョブ・カードによって特定されるジョブの名前です。
- *job_id* は、z/OS システムが生成したジョブ ID です。

ジョブを正常に実行依頼できないと JCL スクリプト・コマンドは失敗し、失敗の理由(例えば、ジョブ・カードが無い、など)を示すメッセージをログに書き込みます。ジョブが実行されたか、あるいは正常に完了したかどうかを認識するには、SDSF などのシステム・サービスを使用します。Managed File Transfer はジョブの実行依頼のみを行うため、情報を提供しません。ジョブを実行するタイミングやジョブ出力の提示方法は、システムが決定します。JCL スクリプトはバッチ・ジョブとして実行依頼されるため、jcl を presrc または predst ネスト・エレメントに指定することは実際には意味を成しません。なぜなら、ジョブの実行依頼が正常に行われたことがわかるだけで、実行が正常に完了してから転送が開始したかどうかはわからないからです。タイプが jcl の、ネストされた有効なエレメントはありません。

以下の例は、JCL ジョブを示しています。

```
//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
```

ネスト・エレメントとして指定するパラメーター

fte:arg

type 属性の値が executable の場合のみ有効です。ネストされた fte:arg エレメントを使用すると、プログラム呼び出しの一部として呼び出されるプログラムに渡される引数を指定します。このプログラム実引数は、fte:arg エレメントにより fte:arg エレメントの出現する順序で指定された値から構成されます。ゼロ個以上の fte:arg エレメントをプログラム呼び出しのネストされたエレメントとして指定するように選択できます。

fte:property

type 属性の値が antscript の場合のみ有効です。ネストされた fte:property エレメントの name 属性および value 属性を使用して、Ant スクリプトに名前値ペアを渡します。ゼロまたは 1 個以上の fte:property エレメントをプログラム呼び出しのネスト・エレメントに指定するように選択できます。

fte:target

type 属性の値が antscript の場合のみ有効です。呼び出す Ant スクリプトのターゲットを指定します。ゼロまたは 1 個以上の fte:target エレメントをプログラム呼び出しのネスト・エレメントに指定するように選択できます。

Arg 属性

値

必須。呼び出されるプログラムに渡される引数の値。

プロパティ属性

名前

必須。Ant スクリプトに渡すプロパティの名前。

値

必須。Ant スクリプトに渡されるプロパティ名に関連付ける値。

例

この例では、[fte:filecopy](#) タスクの一部として指定される fte:postsrc プログラム呼び出しを示します。プログラム呼び出しは、post.sh というプログラムのためのもので、/home/fteuser2/file.bin. の単一の引数を提供します。

```
<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>
```

この例は、[fte:call](#) タスクの一部として指定される fte:command プログラム呼び出しを示しています。このプログラム呼び出しは、command.sh という実行可能プログラムに対するもので、これにはコマンド行引数は渡されません。command.sh が成功を示す 1 の戻りコードを戻さない場合、コマンドは、30 秒後に再試行されます。

```
<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>
```

この例は、`fte:call` タスクの一部として指定される `fte:command` プログラム呼び出しを示しています。プログラム呼び出しは、`script.xml` という Ant スクリプト内のコピーおよび圧縮のターゲットに対するもので、これには 2 つのプロパティが渡されます。

```
<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>
```

関連概念

250 ページの『MFT で実行するプログラムの指定』

Managed File Transfer Agent が実行されているシステムで、プログラムを実行することができます。ファイル転送要求の一部として、転送の開始前または終了後のいずれかにプログラムを実行するように指定することができます。また、管理対象呼び出し要求を実行依頼することで、ファイル転送要求に含まれないプログラムを開始することも可能です。

関連資料

252 ページの『MFT での Apache Ant の使用』

Managed File Transfer は、ファイル転送機能を Apache Ant ツールに組み込むために使用できるタスクを提供します。

848 ページの『MFT 提供の Ant タスク』

Managed File Transfer では、数多くの Ant タスクが用意されており、これらのタスクを使用して、ファイル転送機能にアクセスできます。

カスタマイズのための MFT ユーザー出口での処理

MFT ユーザー出口のメタデータ

Managed File Transfer のユーザー出口ルーチンに提供できるメタデータには、環境、転送、およびファイル・メタデータの 3 つの異なるタイプがあります。このメタデータは、Java のキー/値ペアのマップとして示されます。

環境メタデータ

環境メタデータは、すべてのユーザー出口ルーチンに渡されます。また、ユーザー出口ルーチンの呼び出し元エージェント・ランタイム環境を記述します。このメタデータは読み取り専用で、ユーザー出口ルーチンでは更新できません。

表 71. 環境メタデータ	
キー	説明
AGENT_CONFIGURATION_DIRECTORY_KEY	エージェントの構成情報が入っているディレクトリーの名前。
AGENT_PRODUCT_DIRECTORY_KEY	エージェント・コードがインストールされているディレクトリーの名前。
AGENT_VERSION_KEY	出口ルーチンを呼び出すエージェント・ランタイムのバージョン番号。

表 1 で指定されているキー名および値名は、EnvironmentMetaDataConstants インターフェースで定義されている定数です。

転送メタデータ

転送メタデータは、すべてのユーザー出口ルーチンに渡されます。メタデータは、システム提供の値とユーザー提供の値で構成されます。システム提供値を変更しても、その変更は無視されます。ソース転送開始ユーザー出口のユーザー提供の初期値は、転送の定義時に提供する値に基づくものです。ソース・エージェントは、ソース転送開始ユーザー出口の処理の一部として、ユーザー提供の値を変更することができます。このユーザー出口は、ファイル転送全体が開始する前に呼び出されます。これらの変更は、その転送に関連した他の出口ルーチンへの以降の呼び出しで使用されます。転送メタデータは転送全体に適用されます。

すべてのユーザー出口で転送メタデータから値を読み取ることができますが、転送メタデータを変更できるのはソース転送開始ユーザー出口だけです。

転送メタデータを使用して異なるファイル転送の間で情報を伝搬することはできません。

システム提供の転送メタデータについては、表 2 で詳しく扱われています。

キー	説明
DESTINATION_AGENT_KEY	転送の宛先であるエージェントの名前。
JOB_NAME_KEY	転送要求に関連したジョブ名。
MQMD_USER_KEY	転送要求を実行依頼するために使用されるメッセージからの MQMD ユーザー・フィールド。
ORIGINATING_HOST_KEY	転送要求で発信ホスト名として指定されるホスト名。
ORIGINATING_USER_KEY	転送要求で発信ユーザー ID として指定されるユーザー名。
SOURCE_AGENT_KEY	転送のソースであるエージェントの名前。
TRANSFER_ID_KEY	転送の ID。

表 2 で指定されているキー名および値名は、TransferMetaDataConstants インターフェースで定義されている定数です。

ファイル・メタデータ

ファイル・メタデータは、ファイル仕様の一部として、ソース転送開始出口に渡されます。ソース・ファイルと宛先ファイル用に別個のファイル・メタデータがあります。

ファイル・メタデータを使用して異なるファイル転送の間で情報を伝搬することはできません。

キー	許可値	説明
CONVERT_LINE_SEPARATORS		テキスト転送で使用されるキー値。ソース・データに含まれる CRLF (復帰改行) または LF (改行) 行分離文字シーケンスが、宛先での行分離文字シーケンスに変換されるかどうかを示します。
DELIMITER_KEY		レコード単位のデータを通常のファイルに転送するとき使用されるキー値。レコード・データを分離するための区切り文字を定義します。 メッセージからファイルへの転送およびファイルからメッセージへの転送にも使用されます。
DELIMITER_POSITION_KEY	DELIMITER_POSITION_PREFIX_VALUE DELIMITER_POSITION_POSTFIX_VALUE	区切り文字の位置 (接頭部または接尾部) を定義するために、DELIMITER_KEY と一緒に使用します。

表 73. ファイル・メタデータ (続き)		
キー	許可値	説明
DELIMITER_TYPE_KEY	DELIMITER_TYPE_BINARY_VALUE DELIMITER_TYPE_TEXT_VALUE DELIMITER_TYPE_SIZE_VALUE	区切り文字のタイプを定義するために、DELIMITER_KEY と一緒に使用します。
DESTINATION_EXIST_KEY	DESTINATION_EXIST_KEY_ERROR_VALUE DESTINATION_EXIST_KEY_OVERWRITE_VALUE	宛先ファイルが存在する場合のファイル転送動作を決定します。
FILE_ALIAS_KEY		転送中のファイルの別名を定義するために使用されるキー値。
FILE_CHECKSUM_METHOD_KEY	FILE_CHECKSUM_METHOD_NONE_VALUE FILE_CHECKSUM_METHOD_MD5_VALUE	ファイルの転送時に使用するチェックサム方式を決定します。
FILE_CONVERSION_KEY	FILE_CONVERSION_TEXT_VALUE FILE_CONVERSION_BINARY_VALUE	ファイル内容に適用される変換のタイプを決定します。
FILE_ENCODING_KEY		テキスト・ファイルで使用されるエンコード方式を決定します。
FILE_END_OF_LINE_KEY	FILE_END_OF_LINE_LF_VALUE FILE_END_OF_LINE_CRLF_VALUE	行の終わりを示す文字シーケンスを決定します。<LF> か <CR><LF> のいずれかです。
FILE_SPACE_ALIAS		ファイル・スペースに含まれているファイルの別名を指定します。 注: このメタデータを使用できるのは、FILE_TYPE_KEY が FILE_TYPE_FILE_SPACE_VALUE の場合に限られます。
FILE_SPACE_NAME		ファイル・スペースの名前を指定します。 注: このメタデータを使用できるのは、FILE_TYPE_KEY が FILE_TYPE_FILE_SPACE_VALUE の場合に限られます。
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	宛先ファイル、キュー、またはファイル・スペースを指定します。
GROUP_ID_KEY		メッセージからファイルへの転送で使用されるキー値。ソース・キューから読み取るメッセージ・グループを決定します。この属性が有効なのは、USE_GROUPS_KEY の値が USE_GROUPS_TRUE_VALUE の場合のみです。
INCLUDE_DELIMITER_IN_MESSAGE_KEY	INCLUDE_DELIMITER_IN_MESSAGE_TRUE_VALUE INCLUDE_DELIMITER_IN_MESSAGE_FALSE_VALUE	ファイルからメッセージへの転送で使用されるキー値。ファイルを複数のメッセージに分割するために使用した区切り文字を、それらのメッセージの末尾に組み込むかどうかを決定します。この属性が有効なのは、DELIMITER_TYPE_KEY の値が DELIMITER_TYPE_BINARY_VALUE または DELIMITER_TYPE_TEXT_VALUE の場合のみです。
INSERT_RECORD_LINE_SEPARATOR_KEY		レコード単位のファイルからテキストを転送するときに使用されるキー値。各レコードの後のデータに行分離文字を挿入するかどうかを指定します。

表 73. ファイル・メタデータ (続き)		
キー	許可値	説明
KEEP_TRAILING_SPACES_KEY	KEEP_TRAILING_SPACES_TRUE_VALUE KEEP_TRAILING_SPACES_FALSE_VALUE	固定長形式のデータ・セットから読み取られたレコードから末尾スペースを削除するかどうかを決定するために使用されるキー値。
NEW_RECORD_ON_LINE_SEPARATOR_KEY		レコード単位のファイルへのテキスト転送に使用されるキー値。データに含まれる行分離文字をレコード・データと一緒に組み込むか、または行分離文字で新規レコードの作成を指示するか (行分離文字は書き込まれません) を指定します。
PERSISTENT_KEY	PERSISTENT_TRUE_VALUE PERSISTENT_FALSE_VALUE PERSISTENT_QDEF_VALUE	ファイルからメッセージへの転送で使用されるキー値。メッセージを永続メッセージにするかどうかを決定します。
SET_MQ_PROPS_KEY	SET_MQ_PROPS_TRUE_VALUE SET_MQ_PROPS_FALSE_VALUE	ファイルからメッセージへの転送で使用されるキー値。ファイルの最初のメッセージで IBM MQ メッセージ・プロパティを設定するかどうか、およびエラーの発生時にメッセージをキューに書き込むかどうかを決定します。
UNRECOGNISED_CODE_PAGE_KEY	UNRECOGNISED_CODE_PAGE_FAIL_VALUE UNRECOGNISED_CODE_PAGE_BINARY_VALUE	ファイルからメッセージへの転送で使用されるキー値。宛先キュー・マネージャーがデータのコード・ページを認識できない場合に、テキスト・モードの転送が失敗するか、それとも変換が実行されるかを決定します。
USE_GROUPS_KEY	USE_GROUPS_TRUE_VALUE USE_GROUPS_FALSE_VALUE	メッセージからファイルへの転送で使用されるキー値。完全に揃ったメッセージ・グループだけをソース・キューから転送するかどうかを決定します。
WAIT_TIME_KEY		メッセージからファイルへの転送で使用されるキー値。ソース・エージェントが以下のいずれかの状態になるまで待機する時間を秒単位で決定します。 <ul style="list-style-type: none"> USE_GROUPS_KEY の値が FALSE の場合、キューがもともと空であるか、後で空になった場合に、メッセージがソース・キューに出現するまで。 USE_GROUPS_KEY の値が TRUE の場合、完全に揃ったグループがソース・キューに出現するまで。

表 3 で指定されているキー名および値名は、FileMetaDataConstants インターフェースで定義されている定数です。

MFT リソース・モニター・ユーザー出口

リソース・モニターのユーザー出口を使用して、関連タスクが開始される前に、モニターのトリガー条件が満たされた場合に実行するようカスタム・コードを構成できます。

ユーザー出口コードから直接新しい転送を呼び出すことは推奨されません。ユーザー出口はエージェントの再始動に対して回復力がないため、場合によってはファイルが複数回転送されることになります。

リソース・モニターのユーザー出口は、ユーザー出口の既存インフラストラクチャーを使用します。モニター・ユーザー出口は、モニターがトリガーしてから呼び出されますが、この呼び出しは、対応するタスクがモニターのタスクによって実行される前に行われます。これにより、ユーザー出口は実行されるタスクを変更して、タスクを処理するかどうかを決定できます。モニター・タスクは、モニター・メタデータを更新することで変更できます。更新されたモニター・メタデータは、元のモニターの作成によって作成されたタスク文書で変数置換に使用されます。別の方法として、モニター出口は、パラメーターとして渡されるタスク定義 XML スtring を置換または更新できます。モニター出口は、タスクに対する結果コード (「proceed」または「cancel」のいずれか) を返すことができます。cancel が返された場合、タスクは開始されず、モニター対象リソースがトリガー条件と一致するまでモニターは再開されません。リソースが変更されなければ、トリガーは開始しません。他のユーザー出口と同様に、モニター出口はまとめてチェックできます。出口の 1 つが cancel の結果コードを返すと、結果全体が cancel となり、タスクは開始されません。

- 環境メタデータのマップ (他のユーザー出口と同じ)
- 不変システム・メタデータおよび可変ユーザー・メタデータを含むモニター・メタデータのマップ。不変システム・メタデータは、以下のとおりです。
 - FILENAME - トリガー条件を満たしたファイルの名前
 - FILEPATH - トリガー条件を満たしたファイルへのパス
 - FILESIZE (バイト単位 - このメタデータは存在しない場合がある) - トリガー条件を満たしたファイルのサイズ
 - LASTMODIFIEDDATE (地域別) - トリガー条件を満たしたファイルの最終変更日。エージェントを実行しているタイム・ゾーンの現地日付が ISO 8601 の日付形式で表示されます。
 - LASTMODIFIEDTIME (地域別) - トリガー条件を満たしたファイルの最終変更時刻 (地域別形式)。エージェントを実行しているタイム・ゾーンの現地時間が ISO 8601 の時間形式で表示されます。
 - LASTMODIFIEDDATEUTC - トリガー・ファイルの最終変更日 (世界共通形式)。この日付は、UTC タイム・ゾーンに変換された現地日付として表され、ISO 8601 日付として書式設定されます。
 - LASTMODIFIEDTIMEUTC - トリガー条件を満たしたファイルの最終変更時間 (世界共通形式)。この日付は、UTC タイム・ゾーンに変換された現地時間として表され、ISO 8601 時間として書式設定されます。
 - AGENTNAME - モニター・エージェント名
- モニター・トリガーの結果として実行されるタスクを表す XML String。

モニター出口は、以下のデータを返します。

- さらに進行するかどうか (proceed または cancel) を示す標識
- トリガー条件を満たしたログ・メッセージに挿入する String

モニター出口コードを実行した結果、パラメーターとして最初に渡されたモニター・メタデータおよびタスク定義 XML String も更新されている場合があります。

エージェント・プロパティ monitorExitClasses (agent.properties ファイル内) の値は、ロードするモニター出口クラスを指定します。それぞれの出口クラスはコンマで区切ります。以下に例を示します。

```
monitorExitClasses=testExits.TestExit1,testExits.testExit2
```

モニター・ユーザー出口のインターフェースは、以下のとおりです。

```
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);
}

```

モニター・メタデータ内の IBM 予約値の定数は、以下のとおりです。

```

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";

}

```

ユーザー出口の例

このクラスの例では、MonitorExit インターフェースを実装します。この例では、カスタム置換変数を *REDIRECTEDAGENT* というモニター・メタデータに追加し、そこに 24 時間制の時刻が奇数であれば LONDON という値を、偶数であれば PARIS という値を取り込みます。モニター出口の結果コードは、常に proceed を戻すように設定されています。

```

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;
    }
}

```

REDIRECTEDAGENT 置換変数を使用するモニターに対応するタスクは、以下のようなものになる可能性があります。

```
<?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>
```

この転送が開始される前に、<destinationAgent>エレメントのエージェント属性の値が LONDON または PARIS のいずれかに置き換えられます。

モニター出口クラスで置換変数とタスク定義 XML (大文字) を指定する必要があります。

関連概念

[256 ページの『ユーザー出口での MFT のカスタマイズ』](#)

Managed File Transfer のフィーチャーは、ユーザー出口ルーチンと呼ばれる独自のプログラムを使用してカスタマイズできます。

[872 ページの『MFT ユーザー出口のメタデータ』](#)

Managed File Transfer のユーザー出口ルーチンに提供できるメタデータには、環境、転送、およびファイル・メタデータの 3 つの異なるタイプがあります。このメタデータは、Java のキー/値ペアのマップとして示されます。

[882 ページの『MFT ユーザー出口の Java インターフェース』](#)

ユーザー出口ルーチンの Java インターフェースに関する参照情報については、このセクションのトピックを参照してください。

関連資料

[879 ページの『ユーザー出口用の MFT エージェント・プロパティ』](#)

agent.properties ファイルの標準プロパティに加え、ユーザー出口ルーチン用に特別に用意されたいくつかの拡張プロパティがあります。これらのプロパティはデフォルトでは組み込まれないため、これらのいずれかを使用する場合は、agent.properties ファイルを手動で編集する必要があります。

agent.properties ファイルをそのエージェントの実行中に変更した場合、エージェントを停止および再開してその変更が反映されるようにしてください。

ユーザー出口用の MFT エージェント・プロパティ

agent.properties ファイルの標準プロパティに加え、ユーザー出口ルーチン用に特別に用意されたいくつかの拡張プロパティがあります。これらのプロパティはデフォルトでは組み込まれないため、これらのいずれかを使用する場合は、agent.properties ファイルを手動で編集する必要があります。

agent.properties ファイルをそのエージェントの実行中に変更した場合、エージェントを停止および再開してその変更が反映されるようにしてください。

IBM WebSphere MQ 7.5 以降の場合は、ファイルまたはディレクトリーの場所を表す Managed File Transfer の一部のプロパティで環境変数を使用できます。これにより、製品の一部の実行時に使用されるファイルまたはディレクトリーの場所を、環境の変更(プロセスを実行しているユーザーなど)に合わせ

て変更することができます。詳しくは、[535 ページの『MFT プロパティでの環境変数の使用』](#)を参照してください。

ユーザー出口ルーチン・プロパティ

ユーザー出口ルーチンは、以下の表にリストされている順序で呼び出されます。agent.properties ファイルについて詳しくは、[拡張エージェント・プロパティ: ユーザー出口ルーチン](#)を参照してください。

プロパティ名	説明
sourceTransferEndExitClasses	ソース転送終了出口ルーチンを実装するクラスのコンマ区切りリストを指定します。
sourceTransferStartExitClasses	ソース転送開始出口ルーチンを実装するクラスのコンマ区切りリストを指定します。
destinationTransferStartExitClasses	宛先転送開始ユーザー出口ルーチンを実装するクラスのコンマ区切りリストを指定します。
destinationTransferEndExitClasses	宛先転送ユーザー出口ルーチンを実装するクラスのコンマ区切りリストを指定します。
exitClassPath	<p>ユーザー出口ルーチンのクラスパスの役割を果たす、プラットフォーム固有のディレクトリーの文字区切りリストを指定します。</p> <p>エージェントの出口ディレクトリーは、このクラスパスにある項目の前に検索されます。</p> <p>このプロパティを Windows で使用する場合、パスの区切り文字としてバックslash 文字 (\) ではなく、フォワードslash 文字 (/) を使用します。例えば次のようになります。</p> <pre>exitClassPath=C:/mycomp/mqft/exits/encryptFileExit.jar; C:/mycomp/mqft/exits/fileFilter.jar.</pre> <p>IBM WebSphere MQ 7.5 以降の場合は、このプロパティの値に環境変数を含めることができます。</p>
exitNativeLibraryPath	<p>ユーザー出口ルーチンのネイティブ・ライブラリー・パスの役割を果たす、プラットフォーム固有のディレクトリーの文字区切りリストを指定します。</p> <p>IBM WebSphere MQ 7.5 以降の場合は、このプロパティの値に環境変数を含めることができます。</p>
monitorExitClasses	モニター出口ルーチンを実装するクラスのコンマ区切りリストを指定します。詳しくは、 875 ページの『MFT リソース・モニター・ユーザー出口』 を参照してください。
protocolBridgeCredentialExitClasses	プロトコル・ブリッジ資格情報ユーザー出口ルーチンを実装するクラスのコンマ区切りリストを指定します。詳しくは、 225 ページの『出口クラスを使用したファイル・サーバーの資格情報のマップ』 を参照してください。
protocolBridgePropertiesExitClasses	<p>プロトコル・ブリッジ・サーバー・プロパティ・ユーザー出口ルーチンを実装するクラスのコンマ区切りリストを指定します。</p> <p>詳しくは、219 ページの『ProtocolBridgePropertiesExit2: プロトコル・ファイル・サーバー・プロパティの検索』を参照してください。</p>
IOExitClasses	入出力ユーザー出口ルーチンを実装するクラスのコンマ区切りリストを指定します。IOExit インターフェースを実装するクラスのみリストします。つまり、IOExitResourcePath や IOExitChannel などの他の入出力ユーザー出口インターフェースを実装するクラスはリストしないでください。詳しくは、 260 ページの『MFT 転送入出力ユーザー出口の使用』 を参照してください。

出口の起動順序

ソース出口および宛先出口は、次の順序で起動されます。

1. SourceTransferStartExit
2. DestinationTransferStartExit

3. DestinationTransferEndExit

4. SourceTransferEndExit

ソース出口および宛先出口のチェーニング

複数の出口を指定する場合は、リスト中の最初の出口が最初に起動され、次に2番目の出口、という順序で起動されます。最初の出口で発生した変更内容は次に起動される出口の入力として渡されます。後続の出口についても同様です。例えば、ソース転送開始出口が2つあり、最初の出口による転送メタデータへの変更内容は、2番目の出口に入力されます。それぞれの出口が独自の結果を返します。特定のタイプのすべての出口が転送結果コードとして PROCEED を返すと、全体的な結果は PROCEED になります。1つ以上の出口が CANCEL_TRANSFER を返すと、全体的な結果は CANCEL_TRANSFER になります。出口から返された結果コードおよびストリングはすべて、転送ログに出力されます。

ソース転送開始出口からの全体的な結果が PROCEED であった場合は、出口が行った変更を使用して転送が開始します。全体的な結果が CANCEL_TRANSFER であった場合、ソース転送終了出口が起動された後、転送は取り消されます。転送ログにおける完了状況は、「取り消し済み」になります。

宛先転送開始出口からの全体的な結果が PROCEED であった場合は、出口が行った変更を使用して転送が開始します。全体的な結果が CANCEL_TRANSFER であった場合、宛先転送終了出口が起動された後、ソース転送終了出口が起動されます。最後に転送が取り消されます。転送ログにおける完了状況は、「取り消し済み」になります。

ソースまたは宛先出口が、チェーンで、または実行順に、情報を次の出口に渡す必要がある場合は、転送メタデータを更新することによって行う必要があります。転送メタデータの使用は、出口インプリメンテーションに固有です。例えば、出口が戻りの結果を CANCEL_TRANSFER に設定し、転送がキャンセルされたことを次の出口に伝達する必要がある場合は、他の出口が理解できるように転送メタデータの値を設定することによって行う必要があります。

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
```

関連概念

[256 ページの『ユーザー出口での MFT のカスタマイズ』](#)

Managed File Transfer のフィーチャーは、ユーザー出口ルーチンと呼ばれる独自のプログラムを使用してカスタマイズできます。

[872 ページの『MFT ユーザー出口のメタデータ』](#)

Managed File Transfer のユーザー出口ルーチンに提供できるメタデータには、環境、転送、およびファイル・メタデータの3つの異なるタイプがあります。このメタデータは、Java のキー/値ペアのマップとして示されます。

[882 ページの『MFT ユーザー出口の Java インターフェース』](#)

ユーザー出口ルーチンの Java インターフェースに関する参照情報については、このセクションのトピックを参照してください。

関連資料

[875 ページの『MFT リソース・モニター・ユーザー出口』](#)

リソース・モニターのユーザー出口を使用して、関連タスクが開始される前に、モニターのトリガー条件が満たされた場合に実行するようカスタム・コードを構成できます。

[535 ページの『MFT プロパティーでの環境変数の使用』](#)

IBM WebSphere MQ 7.5 以降、ファイルまたはディレクトリーの場所を表す Managed File Transfer プロパティーで環境変数を使用できます。これにより、製品の一部の実行時に使用されるファイルまたはディレクトリーの場所を、環境の変更に合わせて変えることができます。例えば、プロセスを実行するユーザーの変更などです。

[550 ページの『MFT agent.properties ファイル』](#)

各 Managed File Transfer Agent には、独自のプロパティ・ファイル `agent.properties` があります。このファイルには、エージェントがキュー・マネージャーに接続するために使用する情報が含まれている必要があります。`agent.properties` ファイルには、エージェントの動作を変更するプロパティを含めることもできます。

MFT ユーザー出口の Java インターフェース

ユーザー出口ルーチンの Java インターフェースに関する参照情報については、このセクションのトピックを参照してください。

`CDCredentialExit.java` インターフェース

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 インターフェース

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;
    }
}

```

関連資料

[911 ページの『SourceTransferStartExit.java インターフェース』](#)

[885 ページの『DestinationTransferStartExit.java インターフェース』](#)

[884 ページの『DestinationTransferEndExit.java インターフェース』](#)

[904 ページの『MonitorExit.java インターフェース』](#)

[905 ページの『ProtocolBridgeCredentialExit.java インターフェース』](#)

関連情報

[ユーザー出口での MFT のカスタマイズ](#)

[DestinationTransferEndExit.java インターフェース](#)

DestinationTransferEndExit.java

```
/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2008, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitpoint.api;

/**
 * An interface that is implemented by classes that want to be invoked as part of
 * user exit routine processing. This interface defines a method that will be
 * invoked immediately after completing a transfer on the agent acting as the
 * destination of the transfer.
 */
public interface DestinationTransferEndExit {

    /**
     * Invoked immediately after the completion of a transfer on the agent acting as
     * the destination of the transfer.
     *
     * @param transferExitResult
     *        a result object reflecting whether or not the transfer completed
     *        successfully.
     *
     * @param sourceAgentName
     *        the name of the agent acting as the source of the transfer.
     *
     * @param destinationAgentName
     *        the name of the agent acting as the destination of the
     *        transfer. This is the name of the agent that the
     *        implementation of this method will be invoked from.
     *
     * @param environmentMetaData
     *        meta data about the environment in which the implementation
     *        of this method is running. This information can only be read,
     *        it cannot be updated by the implementation. The constants
     *        defined in EnvironmentMetaDataConstants class can
     *        be used to access the data held by this map.
     *
     * @param transferMetaData
     *        meta data to associate with the transfer. The information can
     *        only be read, it cannot be updated by the implementation. This
     *        map may also contain keys with IBM reserved names. These
     *        entries are defined in the TransferMetaDataConstants
     *        class and have special semantics.
     *
     * @param fileResults
     *        a list of file transfer result objects that describe the source
     *        file name, destination file name and result of each file transfer
     *        operation attempted.
     *
     * @return
     *        an optional description to enter into the log message describing
```

```

*          transfer completion. A value of <code>null</code> can be used
*          when no description is required.
*/
String onDestinationTransferEnd(TransferExitResult transferExitResult,
                                String sourceAgentName,
                                String destinationAgentName,
                                Map<String, String>environmentMetaData,
                                Map<String, String>transferMetaData,
                                List<FileTransferResult>fileResults);
}

```

関連概念

256 ページの『[ユーザー出口での MFT のカスタマイズ](#)』

Managed File Transfer のフィーチャーは、ユーザー出口ルーチンと呼ばれる独自のプログラムを使用してカスタマイズできます。

関連資料

911 ページの『[SourceTransferStartExit.java インターフェース](#)』

910 ページの『[SourceTransferEndExit.java インターフェース](#)』

885 ページの『[DestinationTransferStartExit.java インターフェース](#)』

904 ページの『[MonitorExit.java インターフェース](#)』

905 ページの『[ProtocolBridgeCredentialExit.java インターフェース](#)』

DestinationTransferStartExit.java インターフェース

DestinationTransferStartExit.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2008, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitpoint.api;

/**
 * An interface that is implemented by classes that want to be invoked as part of
 * user exit routine processing. This interface defines a method that will be
 * invoked immediately prior to starting a transfer on the agent acting as the
 * destination of the transfer.
 */
public interface DestinationTransferStartExit {

    /**
     * Invoked immediately prior to starting a transfer on the agent acting as
     * the destination of the transfer.
     *
     * @param sourceAgentName
     *         the name of the agent acting as the source of the transfer.
     *
     * @param destinationAgentName
     *         the name of the agent acting as the destination of the
     *         transfer. This is the name of the agent that the
     *         implementation of this method will be invoked from.
     *
     * @param environmentMetaData
     *         meta data about the environment in which the implementation
     *         of this method is running. This information can only be read,
     *         it cannot be updated by the implementation. The constants
     *         defined in <code>EnvironmentMetaDataConstants</code> class can
     *         be used to access the data held by this map.
     *
     * @param transferMetaData
     *         meta data to associate with the transfer. The information can
     *         only be read, it cannot be updated by the implementation. This

```

```

*          map may also contain keys with IBM reserved names. These
*          entries are defined in the <code>TransferMetaDataConstants</code>
*          class and have special semantics.
*
* @param fileSpecs
*          a list of file specifications that govern the file data to
*          transfer. The implementation of this method can modify the
*          entries in this list and the changes will be reflected in the
*          files transferred. However, new entries may not be added and
*          existing entries may not be removed.
*
* @return   a transfer exit result object which is used to determine if the
*          transfer should proceed, or be cancelled.
*/
TransferExitResult onDestinationTransferStart(String sourceAgentName,
                                             String destinationAgentName,
                                             Map<String, String> environmentMetaData,
                                             Map<String, String> transferMetaData,
                                             List<Reference<String>> fileSpecs);

```

関連概念

[256 ページの『ユーザー出口での MFT のカスタマイズ』](#)

Managed File Transfer のフィーチャーは、ユーザー出口ルーチンと呼ばれる独自のプログラムを使用してカスタマイズできます。

関連資料

[911 ページの『SourceTransferStartExit.java インターフェース』](#)

[910 ページの『SourceTransferEndExit.java インターフェース』](#)

[884 ページの『DestinationTransferEndExit.java インターフェース』](#)

[904 ページの『MonitorExit.java インターフェース』](#)

[905 ページの『ProtocolBridgeCredentialExit.java インターフェース』](#)

[FileTransferResult.java インターフェース](#)

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();
}

```

関連資料

[911 ページの『SourceTransferStartExit.java インターフェース』](#)

[885 ページの『DestinationTransferStartExit.java インターフェース』](#)

[884 ページの『DestinationTransferEndExit.java インターフェース』](#)

[904 ページの『MonitorExit.java インターフェース』](#)

[905 ページの『ProtocolBridgeCredentialExit.java インターフェース』](#)

関連情報

[ユーザー出口での MFT のカスタマイズ](#)

[IOExit.java インターフェース](#)

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;

```

関連概念

[256 ページの『ユーザー出口での MFT のカスタマイズ』](#)

Managed File Transfer のフィーチャーは、ユーザー出口ルーチンと呼ばれる独自のプログラムを使用してカスタマイズできます。

関連資料

[260 ページの『MFT 転送入出力ユーザー出口の使用』](#)

Managed File Transfer 転送入出力ユーザー出口を使用してカスタム・コードを構成し、Managed File Transfer 転送用の基礎ファイル・システムの入出力処理を実行することができます。

IOExitChannel.java インターフェース

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;
}

```

関連概念

[256 ページの『ユーザー出口での MFT のカスタマイズ』](#)

Managed File Transfer のフィーチャーは、ユーザー出口ルーチンと呼ばれる独自のプログラムを使用してカスタマイズできます。

関連資料

[260 ページの『MFT 転送入出力ユーザー出口の使用』](#)

Managed File Transfer 転送入出力ユーザー出口を使用してカスタム・コードを構成し、Managed File Transfer 転送用の基礎ファイル・システムの入出力処理を実行することができます。

IOExitLock.java インターフェース

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();
}

```

関連概念

[256 ページの『ユーザー出口での MFT のカスタマイズ』](#)

Managed File Transfer のフィーチャーは、ユーザー出口ルーチンと呼ばれる独自のプログラムを使用してカスタマイズできます。

関連資料

[260 ページの『MFT 転送入出力ユーザー出口の使用』](#)

Managed File Transfer 転送入出力ユーザー出口を使用してカスタム・コードを構成し、Managed File Transfer 転送用の基礎ファイル・システムの入出力処理を実行することができます。

IOExitPath.java インターフェース

IOExitPath.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2011, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

/**
 * Represents an abstract path that can be inspected and queried by WMQFTE for
 * transfer purposes.
 * <p>
 * There are two types of path supported:
 * <ul>
 * <li>{@link IOExitResourcePath} - Represents a path that denotes a data
 * resource. For example, a file, directory, or group of database records.</li>
 * <li>{@link IOExitWildcardPath} - Represents a wildcard path that can be
 * expanded to multiple {@link IOExitResourcePath} instances.</li>
 * </ul>
 */
public abstract interface IOExitPath {

    /**
     * Obtains the abstract path as a {@link String}.
     *
     * @return The abstract path as a {@link String}.
     */
}

```

```

*/
String getPath();

/**
 * Obtains the name portion of this abstract path as a {@link String}.
 * <p>
 * For example, a UNIX-style file system implementation evaluates the
 * path {@code /home/fteuser/file1.txt} as having a name of {@code
 * file1.txt}.
 *
 * @return the name portion of this abstract path as a {@link String}.
 */
String getName();

/**
 * Obtains the parent path for this abstract path as a {@link String}.
 * <p>
 * For example, a UNIX-style file system implementation evaluates the
 * path {@code /home/fteuser/file1.txt} as having a parent path of {@code
 * /home/fteuser}.
 *
 * @return The parent portion of the path as a {@link String}.
 */
String getParent();

/**
 * Obtains the abstract paths that match this abstract path.
 * <p>
 * If this abstract path denotes a directory resource, a list of paths
 * for all resources within the directory are returned.
 * <p>
 * If this abstract path denotes a wildcard, a list of all paths
 * matching the wildcard are returned.
 * <p>
 * Otherwise null is returned, because this abstract path probably denotes a
 * single file resource.
 *
 * @return An array of {@link IOExitResourcePath}s that
 *         match this path, or null if this method is not applicable.
 */
IOExitResourcePath[] listPaths();
}

```

関連概念

[256 ページの『ユーザー出口での MFT のカスタマイズ』](#)

Managed File Transfer のフィーチャーは、ユーザー出口ルーチンと呼ばれる独自のプログラムを使用してカスタマイズできます。

関連資料

[260 ページの『MFT 転送入出力ユーザー出口の使用』](#)

Managed File Transfer 転送入出力ユーザー出口を使用してカスタム・コードを構成し、Managed File Transfer 転送用の基礎ファイル・システムの入出力処理を実行することができます。

IOExitProperties.java インターフェース

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;
}
}

```

関連概念

256 ページの『[ユーザー出口での MFT のカスタマイズ](#)』

Managed File Transfer のフィーチャーは、ユーザー出口ルーチンと呼ばれる独自のプログラムを使用してカスタマイズできます。

関連資料

260 ページの『[MFT 転送入出力ユーザー出口の使用](#)』

Managed File Transfer 転送入出力ユーザー出口を使用してカスタム・コードを構成し、Managed File Transfer 転送用の基礎ファイル・システムの入出力処理を実行することができます。

IOExitRecordChannel.java インターフェース

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;
}

```

関連概念

256 ページの『ユーザー出口での MFT のカスタマイズ』

Managed File Transfer のフィーチャーは、ユーザー出口ルーチンと呼ばれる独自のプログラムを使用してカスタマイズできます。

関連資料

260 ページの『MFT 転送入出力ユーザー出口の使用』

Managed File Transfer 転送入出力ユーザー出口を使用してカスタム・コードを構成し、Managed File Transfer 転送用の基礎ファイル・システムの入出力処理を実行することができます。

 IOExitRecordResourcePath.java インターフェース

IOExitRecordResourcePath.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2011, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitroutine.api;

import java.io.IOException;

/**
 * Represents a path that denotes a record-oriented data resource (for example,
 * a z/OS data set). It allows the data to be located, the record format to be
 * understood, and {@link IOExitRecordChannel} instances to be created for read
 * or write operations.
 */
public interface IOExitRecordResourcePath extends IOExitResourcePath {

    /**
     * Record formats for record-oriented resources.
     */
    public enum RecordFormat {
        FIXED, VARIABLE
    }

    /**
     * Obtains the record length for records that are maintained by the resource
     * denoted by this abstract path.
     * <p>
     * For a resource with fixed-length records, the data for each record read
     * and written is assumed to be this length.
     * <p>
     * For a resource with variable-length records, this is the maximum length
     * for a record's data.
     * <p>
     * This method should return a value greater than zero, otherwise it can
     * result in the failure of a WMQFTE transfer that involves this abstract
     * path.
     *
     * @return The record length, in bytes, for records maintained by the
     *         resource.
     */
    int getRecordLength();
}

```

```

* Obtains record format, as a {@link RecordFormat} instance, for records
* that are maintained by the resource denoted by this abstract path.
*
* @return A {@link RecordFormat} instance for the record format for records
*         that are maintained by the resource denoted by this abstract
*         path.
*/
RecordFormat getRecordFormat();

/**
 * Opens a {@link IOExitRecordChannel} instance for reading data from the
 * resource denoted by this abstract path. The current data byte position
 * for the resource is expected to be the passed position value, such that
 * when {@link IOExitRecordChannel#read(java.nio.ByteBuffer)} is called,
 * data starting from that position is read.
 * <p>
 * Note that the data byte read position will be on a record boundary.
 *
 * @param position
 *         The required data byte read position.
 * @return A new {@link IOExitRecordChannel} instance allowing data to be
 *         read from the resource denoted by this abstract path.
 * @throws RecoverableIOException
 *         If a recoverable problem occurs while attempting to open the
 *         resource for reading. This means that WMQFTE can attempt to
 *         recover the transfer.
 * @throws IOException
 *         If some other I/O problem occurs.
 */
IOExitRecordChannel openForRead(long position)
    throws RecoverableIOException, IOException;

/**
 * Opens a {@link IOExitRecordChannel} instance for writing data to the
 * resource denoted by this abstract path. Writing of data, using the
 * {@link IOExitRecordChannel#write(java.nio.ByteBuffer)} method, starts at
 * either the beginning of the resource or end of the current data for the
 * resource, depending on the specified append parameter.
 *
 * @param append
 *         When {@code true} indicates that data written to the resource
 *         should be appended to the end of the current data. When
 *         {@code false} indicates that writing of data is to start at
 *         the beginning of the resource; any existing data is lost.
 * @return A new {@link IOExitRecordChannel} instance allowing data to be
 *         written to the resource denoted by this abstract path.
 * @throws RecoverableIOException
 *         If a recoverable problem occurs while attempting to open the
 *         resource for writing. This means that WMQFTE can attempt to
 *         recover the transfer.
 * @throws IOException
 *         If some other I/O problem occurs.
 */
IOExitRecordChannel openForWrite(boolean append)
    throws RecoverableIOException, IOException;
}

```

関連概念

256 ページの『[ユーザー出口での MFT のカスタマイズ](#)』

Managed File Transfer のフィーチャーは、ユーザー出口ルーチンと呼ばれる独自のプログラムを使用してカスタマイズできます。

関連資料

260 ページの『[MFT 転送入出力ユーザー出口の使用](#)』

Managed File Transfer 転送入出力ユーザー出口を使用してカスタム・コードを構成し、Managed File Transfer 転送用の基礎ファイル・システムの入出力処理を実行することができます。

IOExitResourcePath.java インターフェース

IOExitResourcePath.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"

```

```

*
* 5724-H72
*
* □ Copyright IBM Corp. 2011, 2023. All Rights Reserved.
*
* US Government Users Restricted Rights - Use, duplication or
* disclosure restricted by GSA ADP Schedule Contract with
* IBM Corp.
*/
package com.ibm.wmqfte.exitroutine.api;

import java.io.IOException;

/**
 * Represents a path that denotes a data resource (for example, a file,
 * directory, or group of database records). It allows the data to be located
 * and {@link IOExitChannel} instances to be created for read or write
 * operations.
 * <p>
 * There are two types of data resources as follows:
 * <ul>
 * <li>Directory - a container for other data resources. The
 * {@link #isDirectory()} method returns {@code true} for these.</li>
 * <li>File - a data container. This allows data to be read from or written to
 * it. The {@link #isFile()} method returns {@code true} for these.</li>
 * </ul>
 */
public interface IOExitResourcePath extends IOExitPath {

    /**
     * Creates a new {@link IOExitResourcePath} instance for a child path of the
     * resource denoted by this abstract path.
     * <p>
     * For example, with a UNIX-style path, {@code
     * IOExitResourcePath("/home/fteuser/test").newPath("subtest")} could be
     * equivalent to: {@code IOExitResourcePath("/home/fteuser/test/subtest")}
     *
     * @param child
     *         The child path name.
     * @return A new {@link IOExitResourcePath} instance that represents a child
     *         of this path.
     */
    IOExitResourcePath newPath(final String child);

    /**
     * Creates the directory path for the resource denoted by this abstract
     * path, including any necessary but nonexistent parent directories. If the
     * directory path already exists, this method has no effect.
     * <p>
     * If this operation fails, it might have succeeded in creating some of the
     * necessary parent directories.
     *
     * @throws IOException
     *         If the directory path cannot be fully created, when it does
     *         not already exist.
     */
    void makePath() throws IOException;

    /**
     * Obtains the canonical path of the abstract path as a {@link String}.
     * <p>
     * A canonical path is defined as being absolute and unique. For example,
     * the path can be represented as UNIX-style relative path: {@code
     * test/file.txt} but the absolute and unique canonical path representation
     * is: {@code /home/fteuser/test/file.txt}
     *
     * @return The canonical path as a {@link String}.
     * @throws IOException
     *         If the canonical path cannot be determined for any reason.
     */
    String getCanonicalPath() throws IOException;

    /**
     * Tests if this abstract path is an absolute path.
     * <p>
     * For example, a UNIX-style path, {@code /home/fteuser/test} is an absolute
     * path, whereas {@code fteuser/test} is not.
     *
     * @return {@code true} if this abstract path is an absolute path, {@code
     *         false} otherwise.
     */
    boolean isAbsolute();

```

```

/**
 * Tests if the resource denoted by this abstract path exists.
 *
 * @return {@code true} if the resource denoted by this abstract path
 *         exists, {@code false} otherwise.
 * @throws IOException
 *         If the existence of the resource cannot be determined for any
 *         reason.
 */
boolean exists() throws IOException;

/**
 * Tests whether the calling application can read the resource denoted by
 * this abstract path.
 *
 * @return {@code true} if the resource for this path exists and can be
 *         read, {@code false} otherwise.
 * @throws IOException
 *         If a problem occurs while attempting to determine if the
 *         resource can be read.
 */
boolean canRead() throws IOException;

/**
 * Tests whether the calling application can modify the resource denoted by
 * this abstract path.
 *
 * @return {@code true} if the resource for this path exists and can be
 *         modified, {@code false} otherwise.
 * @throws IOException
 *         If a problem occurs while attempting to determine if the
 *         resource can be modified.
 */
boolean canWrite() throws IOException;

/**
 * Tests whether the specified user is permitted to read the resource
 * denoted by this abstract path.
 * <p>
 * When WMQFTE invokes this method, the user identifier is the MQMD user
 * identifier for the requesting transfer.
 *
 * @param userId
 *         User identifier to test for access.
 * @return {@code true} if the resource for this abstract path exists and is
 *         permitted to be read by the specified user, {@code false}
 *         otherwise.
 * @throws IOException
 *         If a problem occurs while attempting to determine if the user
 *         is permitted to read the resource.
 */
boolean readPermitted(String userId) throws IOException;

/**
 * Tests whether the specified user is permitted to modify the resource
 * denoted by this abstract path.
 * <p>
 * When WMQFTE invokes this method, the user identifier is the MQMD user
 * identifier for the requesting transfer.
 *
 * @param userId
 *         User identifier to test for access.
 * @return {@code true} if the resource for this abstract path exists and is
 *         permitted to be modified by the specified user, {@code false}
 *         otherwise.
 * @throws IOException
 *         If a problem occurs while attempting to determine if the user
 *         is permitted to modify the resource.
 */
boolean writePermitted(String userId) throws IOException;

/**
 * Tests if the resource denoted by this abstract path is a directory-type
 * resource.
 *
 * @return {@code true} if the resource denoted by this abstract path is a
 *         directory type resource, {@code false} otherwise.
 */
boolean isDirectory();

```

```

* Creates the resource denoted by this abstract path, if it does not
* already exist.
*
* @return {@code true} if the resource does not exist and was successfully
*         created, {@code false} if the resource already existed.
* @throws RecoverableIOException
*         If a recoverable problem occurs while attempting to create
*         the resource. This means that WMQFTE can attempt to recover
*         the transfer.
* @throws IOException
*         If some other I/O problem occurs.
*/
boolean createNewPath() throws RecoverableIOException, IOException;

/**
* Tests if the resource denoted by this abstract path is a file-type
* resource.
*
* @return {@code true} if the resource denoted by this abstract path is a
*         file type resource, {@code false} otherwise.
*/
boolean isFile();

/**
* Obtains the last modified time for the resource denoted by this abstract
* path.
* <p>
* This time is measured in milliseconds since the epoch (00:00:00 GMT,
* January 1, 1970).
*
* @return The last modified time for the resource denoted by this abstract
*         path, or a value of 0L if the resource does not exist or a
*         problem occurs.
*/
long lastModified();

/**
* Deletes the resource denoted by this abstract path.
* <p>
* If the resource is a directory, it must be empty for the delete to work.
*
* @throws IOException
*         If the delete of the resource fails for any reason.
*/
void delete() throws IOException;

/**
* Renames the resource denoted by this abstract path to the specified
* destination abstract path.
* <p>
* The rename should still be successful if the resource for the specified
* destination abstract path already exists and it is possible to replace
* it.
*
* @param destination
*         The new abstract path for the resource denoted by this
*         abstract path.
* @throws IOException
*         If the rename of the resource fails for any reason.
*/
void renameTo(IOExceptionResourcePath destination) throws IOException;

/**
* Creates a new path to use for writing to a temporary resource that did
* not previously exist.
* <p>
* The implementation can choose the abstract path name for the temporary
* resource. However, for clarity and problem diagnosis, the abstract path
* name for the temporary resource should be based on this abstract path
* name with the specified suffix appended and additional characters to make
* the path unique (for example, sequence numbers), as required.
* <p>
* When WMQFTE transfers data to a destination it normally attempts to first
* write to a temporary resource then on transfer completion renames the
* temporary resource to the required destination. This method is called by
* WMQFTE to create a new temporary resource path. The returned path should
* be new and the resource should not previously exist.
*
* @param suffix
*         Recommended suffix to use for the generated temporary path.
*
* @return A new {@link IOExceptionResourcePath} instance for the temporary

```

```

*         resource path, that did not previously exist.
* @throws RecoverableIOException
*         If a recoverable problem occurs whilst attempting to create
*         the temporary resource. This means that WMQFTE can attempt to
*         recover the transfer.
* @throws IOException
*         If some other I/O problem occurs.
*/
IOExitResourcePath createTempPath(String suffix)
    throws RecoverableIOException, IOException;

/**
* Opens a {@link IOExitChannel} instance for reading data from the resource
* denoted by this abstract path. The current data byte position for the
* resource is expected to be the passed position value, such that when
* {@link IOExitChannel#read(java.nio.ByteBuffer)} is called, data starting
* from that position is read.
*
* @param position
*         The required data byte read position.
* @return A new {@link IOExitChannel} instance allowing data to be read
*         from the resource denoted by this abstract path.
* @throws RecoverableIOException
*         If a recoverable problem occurs while attempting to open the
*         resource for reading. This means that WMQFTE can attempt to
*         recover the transfer.
* @throws IOException
*         If some other I/O problem occurs.
*/
IOExitChannel openForRead(long position) throws RecoverableIOException,
    IOException;

/**
* Opens a {@link IOExitChannel} instance for writing data to the resource
* denoted by this abstract path. Writing of data, using the
* {@link IOExitChannel#write(java.nio.ByteBuffer)} method, starts at either
* the beginning of the resource or end of the current data for the
* resource, depending on the specified append parameter.
*
* @param append
*         When {@code true} indicates that data written to the resource
*         should be appended to the end of the current data. When
*         {@code false} indicates that writing of data is to start at
*         the beginning of the resource; any existing data is lost.
* @return A new {@link IOExitChannel} instance allowing data to be written
*         to the resource denoted by this abstract path.
* @throws RecoverableIOException
*         If a recoverable problem occurs whilst attempting to open the
*         resource for writing. This means that WMQFTE can attempt to
*         recover the transfer.
* @throws IOException
*         If some other I/O problem occurs.
*/
IOExitChannel openForWrite(boolean append) throws RecoverableIOException,
    IOException;

/**
* Tests if the resource denoted by this abstract path is in use by another
* application. Typically, this is because another application has a lock on
* the resource either for shared or exclusive access.
*
* @return {@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();
}

```

関連概念

256 ページの『ユーザー出口での MFT のカスタマイズ』

Managed File Transfer のフィーチャーは、ユーザー出口ルーチンと呼ばれる独自のプログラムを使用してカスタマイズできます。

関連資料

260 ページの『MFT 転送入出力ユーザー出口の使用』

Managed File Transfer 転送入出力ユーザー出口を使用してカスタム・コードを構成し、Managed File Transfer 転送用の基礎ファイル・システムの入出力処理を実行することができます。

IOExitWildcardPath.java インターフェース

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 {
```

関連概念

256 ページの『ユーザー出口での MFT のカスタマイズ』

Managed File Transfer のフィーチャーは、ユーザー出口ルーチンと呼ばれる独自のプログラムを使用してカスタマイズできます。

関連資料

260 ページの『MFT 転送入出力ユーザー出口の使用』

Managed File Transfer 転送入出力ユーザー出口を使用してカスタム・コードを構成し、Managed File Transfer 転送用の基礎ファイル・システムの入出力処理を実行することができます。

MonitorExit.java インターフェース

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 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);
}

```

関連概念

[256 ページの『ユーザー出口での MFT のカスタマイズ』](#)

Managed File Transfer のフィーチャーは、ユーザー出口ルーチンと呼ばれる独自のプログラムを使用してカスタマイズできます。

関連タスク

[170 ページの『MFT リソースのモニター』](#)

キューやディレクトリーなどの Managed File Transfer リソースをモニターできます。そのリソースで条件が満たされると、リソース・モニターがファイル転送などのタスクを開始します。**fteCreateMonitor** コマンドを使用するか、IBM MQ Explorer に対応した Managed File Transfer プラグインの「モニター」ビューを使用して、リソース・モニターを作成できます。

関連資料

[911 ページの『SourceTransferStartExit.java インターフェース』](#)

[910 ページの『SourceTransferEndExit.java インターフェース』](#)

[885 ページの『DestinationTransferStartExit.java インターフェース』](#)

[884 ページの『DestinationTransferEndExit.java インターフェース』](#)

[905 ページの『ProtocolBridgeCredentialExit.java インターフェース』](#)

[ProtocolBridgeCredentialExit.java インターフェース](#)

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);
}

```

関連概念

256 ページの『[ユーザー出口での MFT のカスタマイズ](#)』

Managed File Transfer のフィーチャーは、ユーザー出口ルーチンと呼ばれる独自のプログラムを使用してカスタマイズできます。

関連タスク

225 ページの『[出口クラスを使用したファイル・サーバーの資格情報のマップ](#)』

プロトコル・ブリッジ・エージェントのデフォルトの資格情報マッピング機能を使用しない場合は、独自のユーザー出口を作成して、Managed File Transfer のユーザー資格情報をファイル・サーバーのユーザー資格情報にマップできます。Managed File Transfer には、ユーザー資格情報マッピングを実行するサンプルのユーザー出口が用意されています。資格情報マッピング・ユーザー出口を構成すると、デフォルトの資格情報マッピング機能の代わりになります。

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);
}
}
```

関連概念

256 ページの『[ユーザー出口での MFT のカスタマイズ](#)』

Managed File Transfer のフィーチャーは、ユーザー出口ルーチンと呼ばれる独自のプログラムを使用してカスタマイズできます。

関連タスク

225 ページの『[出口クラスを使用したファイル・サーバーの資格情報のマップ](#)』

プロトコル・ブリッジ・エージェントのデフォルトの資格情報マッピング機能を使用しない場合は、独自のユーザー出口を作成して、Managed File Transfer のユーザー資格情報をファイル・サーバーのユーザー資格情報にマップできます。Managed File Transfer には、ユーザー資格情報マッピングを実行するサンプルのユーザー出口が用意されています。資格情報マッピング・ユーザー出口を構成すると、デフォルトの資格情報マッピング機能の代わりになります。

ProtocolBridgePropertiesExit2.java インターフェース

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);
}

```

関連概念

221 ページの『[ProtocolBridgePropertiesExit: プロトコル・ファイル・サーバー・プロパティの検索](#)』
 プロトコル・ファイル・サーバーが多数ある場合は、
 com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit インターフェースを実装

して、転送で参照されているプロトコル・ファイル・サーバー・プロパティを検索できます。ProtocolBridgeProperties.xml ファイルを保持するよりはむしろ、このインターフェースを実装することができます。ProtocolBridgePropertiesExit2.java インターフェースの使用が推奨されますが、ProtocolBridgePropertiesExit.java インターフェースもサポートされています。IBM WebSphere MQ File Transfer Edition から ProtocolBridgePropertiesExit.java インターフェースを既に実装していた場合は、IBM WebSphere MQ 7.5 以降でそれを使用できます。ProtocolBridgePropertiesExit2.java 内の新しい getCredentialLocation メソッドは、ProtocolBridgeCredentials.xml ファイルのデフォルトの場所（つまり、ホーム・ディレクトリー）を使用します。

256 ページの『ユーザー出口での MFT のカスタマイズ』

Managed File Transfer のフィーチャーは、ユーザー出口ルーチンと呼ばれる独自のプログラムを使用してカスタマイズできます。

関連タスク

225 ページの『出口クラスを使用したファイル・サーバーの資格情報のマップ』

プロトコル・ブリッジ・エージェントのデフォルトの資格情報マッピング機能を使用しない場合は、独自のユーザー出口を作成して、Managed File Transfer のユーザー資格情報をファイル・サーバーのユーザー資格情報にマップできます。Managed File Transfer には、ユーザー資格情報マッピングを実行するサンプルのユーザー出口が用意されています。資格情報マッピング・ユーザー出口を構成すると、デフォルトの資格情報マッピング機能の代わりになります。

SourceFileExitFileSpecification.java クラス

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;
}
}
}

```

関連概念

872 ページの『MFT ユーザー出口のメタデータ』

Managed File Transfer のユーザー出口ルーチンに提供できるメタデータには、環境、転送、およびファイル・メタデータの3つの異なるタイプがあります。このメタデータは、Java のキー/値ペアのマップとして示されます。

SourceTransferEndExit.java インターフェース

SourceTransferEndExit.java

```

/*
 * Licensed Materials - Property of IBM
 *
 * "Restricted Materials of IBM"
 *
 * 5724-H72
 *
 * © Copyright IBM Corp. 2008, 2023. All Rights Reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with
 * IBM Corp.
 */
package com.ibm.wmqfte.exitpoint.api;

/**

```

```

* An interface that is implemented by classes that want to be invoked as part of
* user exit routine processing. This interface defines a method that will be
* invoked immediately after completing a transfer on the agent acting as the
* source of the transfer.
*/
public interface SourceTransferEndExit {

    /**
     * Invoked immediately after the completion of a transfer on the agent acting as
     * the source of the transfer.
     *
     * @param transferExitResult
     *        a result object reflecting whether or not the transfer completed
     *        successfully.
     *
     * @param sourceAgentName
     *        the name of the agent acting as the source of the transfer.
     *        This is the name of the agent that the implementation of this
     *        method will be invoked from.
     *
     * @param destinationAgentName
     *        the name of the agent acting as the destination of the
     *        transfer.
     *
     * @param environmentMetaData
     *        meta data about the environment in which the implementation
     *        of this method is running. This information can only be read,
     *        it cannot be updated by the implementation. The constants
     *        defined in 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 onSourceTransferEnd(TransferExitResult transferExitResult,
                               String sourceAgentName,
                               String destinationAgentName,
                               Map<String, String>environmentMetaData,
                               Map<String, String>transferMetaData,
                               List<FileTransferResult>fileResults);

}

```

関連概念

[256 ページの『ユーザー出口での MFT のカスタマイズ』](#)

Managed File Transfer のフィーチャーは、ユーザー出口ルーチンと呼ばれる独自のプログラムを使用してカスタマイズできます。

関連資料

[911 ページの『SourceTransferStartExit.java インターフェース』](#)

[885 ページの『DestinationTransferStartExit.java インターフェース』](#)

[884 ページの『DestinationTransferEndExit.java インターフェース』](#)

[904 ページの『MonitorExit.java インターフェース』](#)

[905 ページの『ProtocolBridgeCredentialExit.java インターフェース』](#)

[SourceTransferStartExit.java インターフェース](#)

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 EnvironmentMetaDataConstants 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 will be
     *     reflected in subsequent exit routine invocations. 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 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);
}
```

関連概念

[256 ページの『ユーザー出口での MFT のカスタマイズ』](#)

Managed File Transfer のフィーチャーは、ユーザー出口ルーチンと呼ばれる独自のプログラムを使用してカスタマイズできます。

関連資料

[909 ページの『SourceFileExitFileSpecification.java クラス』](#)

[910 ページの『SourceTransferEndExit.java インターフェース』](#)

[885 ページの『DestinationTransferStartExit.java インターフェース』](#)

[884 ページの『DestinationTransferEndExit.java インターフェース』](#)

[904 ページの『MonitorExit.java インターフェース』](#)

[905 ページの『ProtocolBridgeCredentialExit.java インターフェース』](#)

TransferExitResult.java インターフェース

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;
    }
}

```

関連資料

[911 ページの『SourceTransferStartExit.java インターフェース』](#)

[885 ページの『DestinationTransferStartExit.java インターフェース』](#)

[884 ページの『DestinationTransferEndExit.java インターフェース』](#)

[904 ページの『MonitorExit.java インターフェース』](#)

[905 ページの『ProtocolBridgeCredentialExit.java インターフェース』](#)

関連情報

[ユーザー出口での MFT のカスタマイズ](#)

MFT エージェントのコマンド・キューに PUT できるメッセージ形式

次の XML スキーマは、エージェントによる操作の実行を要求するために、エージェントのコマンド・キューに PUT できるメッセージの形式を定義します。XML メッセージは、コマンド行のコマンドまたはアプリケーションを使用して、エージェントのコマンド・キューに置くことができます。

ファイル転送要求メッセージ・フォーマット

ファイル転送は、エージェントのコマンド・キューに到着した XML メッセージによって開始されます。通常、ユーザーがファイル転送コマンドを発行したか、IBM MQ Explorer を使用した結果、XML メッセージが到着します。転送要求 XML は、FileTransfer.xsd スキーマに準拠していて、<request> エlement をルート・Element として持っている必要があります。FileTransfer.xsd スキーマ文書は `MQ_INSTALLATION_PATH/mqft/samples/schema` ディレクトリーにあります。FileTransfer.xsd スキーマは、同じディレクトリーにある `fteutils.xsd` をインポートします。

ファイル転送メッセージは、以下の 3 つのルート・Element のいずれかを含むことができます。

- <request> - 新しい転送要求、管理対象呼び出し要求、または保留中のスケジュール済み転送の削除のためのもの
- <cancel> - 進行中のファイル転送を取り消すためのもの
- <transferSpecifications>- **fteCreateTransfer** コマンドで使用される複数の転送ファイル・グループを指定する場合

<transferSpecifications> Element を使用して複数の転送グループを指定する方法については、[転送定義ファイルの使用](#)を参照してください。

スキーマ

次のスキーマは、転送要求 XML メッセージで有効な Element について示します。

```

<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>

```

転送要求メッセージについて

転送要求メッセージで使用するエレメントと属性を以下のリストにまとめます。

エレメントの説明

<request>

ファイル転送要求を指定するために必要なすべてのエレメントが含まれるグループ・エレメント。

属性	説明
バージョン	Managed File Transfer によって指定された、このエレメントのバージョンを示します。

<managedTransfer>

1つのファイル転送または1つのファイル転送グループに必要なすべてのエレメントが含まれるグループ・エレメント。

<deleteScheduledTransfer>

スケジュール転送を取り消すための発信元と ID 情報が含まれるグループ・エレメント。

<managedCall>

プログラムまたは実行可能ファイルの1つの管理対象呼び出しに必要なすべてのエレメントが含まれるグループ・エレメント。

<ID>

保留中のスケジュール済み転送のリストから削除する転送要求を指定する固有 ID。

<originator>

要求の発信元を示すエレメントが含まれるグループ・エレメント。

<hostName>

ソース・ファイルが置かれているシステムのホスト名。

<userID>

ファイル転送を開始したユーザーの ID。

<mqmdUserID>

オプション。メッセージ記述子 (MQMD) に指定された IBM MQ ユーザー ID。

<schedule>

ファイル転送の予定時間、繰り返し動作、および次の実行予定を示すグループ・エレメント。

<submit>

スケジュール済み転送の開始日時を示します。

属性	説明
timebase	使用するタイム・ゾーンを示します。この属性には、以下のいずれかの値を使用できます。 <ul style="list-style-type: none"> • source - ソース・エージェントのタイム・ゾーンを使用します。 • admin - コマンドを発行する管理者のタイム・ゾーンを使用します。 • UTC - 協定世界時を使用します。
timezone	時間基準値に従ったタイム・ゾーンの記述

<repeat>

スケジュール済み転送の繰り返し頻度、スケジュール済み転送の繰り返し回数、およびスケジュール済み転送の繰り返しの停止日時の詳細を含むグループ・エレメント。

<frequency>

転送を繰り返す時間間隔。

属性	説明
interval	以下のいずれかで示される間隔の単位。 <ul style="list-style-type: none"> • minutes • hours • 日 • weeks • months • years

<expireTime>

スケジュール済み転送の繰り返子を停止する日時を指定するオプションのエレメント。このエレメントと <expireCount> エレメントは相互に排他的です。

<expireCount>

スケジュール済みファイル転送を終了するまでの実行回数を指定するオプションのエレメント。このエレメントと <expireTime> エレメントは相互に排他的です。

<sourceAgent>

ソース・ファイルがあるシステム上のエージェントの名前を示します。

属性	説明
エージェント	エージェントの名前を示します。
キュー・マネージャー	エージェントのキュー・マネージャーの名前。

<destinationAgent>

ファイルの転送先にするシステム上のエージェントの名前を示します。

属性	説明
エージェント	エージェントの名前を示します。
キュー・マネージャー	エージェントのキュー・マネージャーの名前。
hostName	エージェントのキュー・マネージャーのホスト名または IP アドレス。
portNumber	宛先エージェントのキュー・マネージャーとのクライアント接続で使用されるポート番号。
channel	宛先エージェントのキュー・マネージャーとの接続に使用されるチャンネル名。

<trigger>

ファイル転送の実行のために満たす必要がある条件を指定するオプションの要素。

属性	説明
log	トリガー障害がログに記録されるかどうかを示すフラグ。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • yes - 失敗したトリガー転送についてのログ項目が作成されます。 • no - 失敗したトリガー転送についてのログ項目が作成されません。

<fileExist>

ソース・エージェントと同じシステムにあるファイル名のコンマ区切りリストを指定します。この名前リスト内のファイルがトリガーの条件を満たす場合、転送が実行されます。この要素と <fileSize> エレメントは相互に排他的です。

属性	説明
comparison	名前リストに対してソース・ファイル名を評価する方法を示します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • = 名前リスト内の少なくとも 1 つのファイル名が一致しなければなりません。 • != 名前リストの少なくとも 1 つのファイルが存在しません。
値	比較タイプを示します。 <ul style="list-style-type: none"> • exist: ファイルが存在しなければなりません

<fileSize>

ソース・エージェントと同じシステムにあるファイル名のコンマ区切りリストを指定します。この名前リスト内のファイルがトリガーの条件を満たす場合、転送が実行されます。この要素と <fileExist> エレメントは相互に排他的です。

属性	説明
comparison	名前リストに対してソース・ファイル名を評価する方法を示します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • >= 名前リスト内のファイル名のいずれかが存在し、value 属性で指定した最小サイズを有しています。

属性	説明
値	次のいずれかの単位を持つ整数値で指定したファイル・サイズ <ul style="list-style-type: none"> • B - バイト • KB - キロバイト • MB - メガバイト • GB - ギガバイト (単位値の大/小文字は区別されません)

<reply>

同期ファイル転送用に生成された一時応答キューの名前を示します (コマンド行で **-w** パラメーターにより指定します)。キューの名前は、`command.properties` 構成ファイル内のキー **dynamicQueuePrefix**、またはデフォルトの `WMQFTE.*` によって定義されます。指定されていない場合、

属性	説明
detailed	応答メッセージで詳細な転送結果情報が必要かどうか。転送ごとに複数の応答メッセージを生成できます。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • true - 詳細な応答情報が必要です。情報の形式は、転送ログの進行メッセージ (つまり <transferSet> エレメント) に公開されるものと同じです。詳しくは、748 ページの『ファイル転送ログ・メッセージ・フォーマット』を参照してください。詳細な応答情報は、転送ソース・エージェントの <code>enableDetailedReplyMessages</code> プロパティーが true に設定されている場合にのみ存在します。 • false - 詳細な応答情報は不要です。 デフォルト値は false です。
QMGR	応答を受け取るために一時動的キューが生成されるコマンド・キュー・マネージャーの名前。
persistent	応答キューに書き込まれるメッセージが永続メッセージかどうか。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • true - メッセージは永続メッセージです • false - メッセージは永続メッセージではありません • qdef - メッセージの永続性を応答キューのプロパティーで定義します デフォルト値は false です。

<transferSet>

まとめて実行するファイル転送のグループ、またはまとめて実行する管理対象呼び出しのグループを示します。送信中は、<transferSet> は <item> エレメントが含まれるグループ・エレメントです。

属性	説明
priority	転送の優先順位。priority は 0 から 9 までの範囲の値で、0 が最低の優先順位です。デフォルトの優先順位は 0 で、デフォルトではソース・エージェントの優先順位が転送で使用されます。

<metaDataSet>

1つ以上のメタデータ項目を含むオプションのグループ・エレメント。

<metaData>

エージェントにより呼び出される出口点に渡される、ユーザー定義のメタデータを指定します。このエレメントは、ストリングとしてメタデータ値を含みます。

属性	説明
key	ストリングとしてのメタデータ名

<call>

呼び出すプログラムまたは実行可能ファイルを指定する <command> エレメントを含むグループ・エレメント。

<コマンド>

呼び出すプログラムまたは実行可能ファイルを指定します。このコマンドは、エージェント・コマンド・パスに存在していなければなりません。詳しくは、552 ページの表 45 を参照してください。このエレメントには、オプションの <argument> エレメントを含めることができます。

属性	説明
名前	コマンドの名前。
successRC	このコマンドによって返される、コマンドの成功を示す戻りコード。デフォルトは 0 です。
retryCount	コマンドが失敗したときに、コマンドを再試行する回数。
retryWait	コマンドを次に再試行するまで待機する時間 (秒数)。
タイプ	呼び出すプログラムのタイプ。有効な値は antscript、jcl、または executable です。

<argument>

コマンドに渡す引数を指定します。

<item>

ソースおよび宛先のファイル名と位置を示すエレメントが含まれるグループ・エレメント。

属性	説明
mode	転送モードがバイナリーであるかテキストであるかを示します。
checksumMethod	デジタル署名を作成するためにメッセージ・ダイジェストを生成するハッシュ・アルゴリズムのタイプを示します。有効な値は MD5 または none です。

<source>

ソース・システム上のファイルと、転送完了後にそれらのファイルを除去するかどうかを指定するグループ・エレメント

属性	説明
recursive	<source> エレメントがディレクトリーであるかそこにワイルドカード文字が含まれる場合にファイルが再帰的にサブディレクトリーに転送されることを指定します。
disposition	<source> がその宛先に正常に転送されたときに <source> エレメントに対して取るアクションを指定します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • leave - ソース・ファイルは変更されません。 • delete - ソース・ファイルは、それが正常に転送された後にソース・システムから削除されます。

<file>

転送ソースを指定します。 **Multi** Multiplatforms の場合、転送ソースは、ファイルまたはディレクトリー名にすることができます。z/OS プラットフォームでは、ファイル名、ディレクトリー名、データ・セット名、または PDS 名を転送のソースにすることができます。ご使用のオペレーティング・

システムと整合するフォーマットの完全修飾パスを使用します (例えば C:/from/here.txt)。ファイル URI は使用しないでください。

属性	説明
別名	ソース・ファイルの別名を指定します。この別名は、転送で指定されているディレクトリー・パスを除外したソース・ファイルの名前です。
EOL	テキスト転送の行末マーカを指定します。有効な値は次のとおりです。 <ul style="list-style-type: none"> • LF - 改行文字のみ • CRLF - 復帰と改行の文字シーケンス
encoding	テキスト・ファイル転送のソース・ファイルのエンコード。
 delimiter	レコード単位ソース・ファイル (z/OS データ・セットなど) でレコードとレコードの間に組み込まれる区切り文字を指定します。区切り文字の値は、00-FF の範囲にある 2 桁の 16 進数に接頭部 x を付けた形式で指定してください。例えば、x12 または x03,x7F などです。
delimiterType	宛先ファイルで個々のメッセージ・データの後に組み込む区切り文字のタイプを指定します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • binary - 16 進数区切り文字 この属性は、7.0.4.1 機能を有効にした場合にのみ使用できます。
delimiterPosition	レコード単位ソース・ファイル・レコードを通常のファイルに書き込むときに、区切り文字を挿入する位置を指定します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • prefix - 区切り文字は、宛先ファイルの各ソース・レコード単位ファイル・レコードのデータの前に挿入されます。 • postfix - 区切り文字は、宛先ファイルの、ソース・レコード単位の各ファイル・レコードからのデータの後に挿入されます。
includeDelimiterInFile	レコード単位ソース・ファイルでレコードとレコードの間に区切り文字を組み込むかどうかを指定します。
 keepTrailingSpaces	テキスト・モード転送の一部として固定長形式のデータ・セットから読み取られるソース・レコードの末尾スペースを保持するかどうかを指定します。デフォルトでは、末尾スペースは削除されます。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • true - テキスト・モード転送の一部として固定長形式のデータ・セットから読み取られるソース・レコードの末尾スペースを保持します • false - テキスト・モード転送の一部として固定長形式のデータ・セットから読み取られるソース・レコードから末尾スペースを削除します

<queue>

<source> エレメントと一緒に使用する場合には、転送元のキューの名前を指定します。そのキューは、ソース・エージェントのキュー・マネージャーに存在していなければなりません。QUEUE というフォーマットを使用してください。キュー・マネージャー名を組み込むことはできません。ソース・エージェントのキュー・マネージャーに存在しているキューを指定する必要があります。<queue> エレメントを <destination> エレメント内で使用した場合、<source> エレメント内では使用できません。

属性	説明
useGroups	完全に揃った最初のメッセージ・グループだけをソース・キューから転送するかどうかを指定します。有効な値は以下のとおりです。

属性	説明
	<ul style="list-style-type: none"> • true - 完全に揃った最初のメッセージ・グループだけを転送します • false - ソース・キューにあるすべてのメッセージを転送します
groupId	ソース・キューから読み取るメッセージ・グループを指定します。この属性は、useGroups 属性の値が true の場合のみ有効です。
delimiterType	宛先ファイルで個々のメッセージ・データの後に組み込む区切り文字のタイプを指定します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • text - テキスト区切り文字または Java リテラル区切り文字 • binary - 16 進数区切り文字
delimiter	宛先ファイルで個々のメッセージ・データ間に組み込む区切り文字を指定します。
delimiterPosition	宛先ファイルで、区切り文字を個々のメッセージ・データの前に組み込むか、後に組み込むかを指定します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • prefix - 区切り文字はデータの前に組み込まれます • postfix - 区切り文字はデータの後ろに組み込まれます。
encoding	ソース・キューのエンコード方式を指定します。
waitTime	<p>ソース・エージェントが以下のいずれかの状態の発生を待つ時間を秒単位で指定します。</p> <ul style="list-style-type: none"> • キューがもともと空だったか、後で空になった場合は、ソース・キューにメッセージが出現する、という状態 • useGroups 属性が true に設定されている場合は、完全に揃ったグループがソース・キューに出現する、という状態 <p>waitTime 値の設定については、682 ページの『メッセージからファイルへの転送の待機時間を指定する際のガイダンス』を参照してください。</p>

<destination>

宛先と、宛先エージェントにファイルが存在する場合の動作を指定するグループ・エレメント。

<file> と <queue> のいずれかを destination の子エレメントとして指定できます。

属性	説明
タイプ	<p>宛先のタイプ。有効な値は以下のとおりです。</p> <ul style="list-style-type: none"> • file - ファイルを宛先として指定します • directory - ディレクトリーを宛先として指定します •  dataset - z/OS データ・セットを宛先として指定します •  pds - z/OS 区分データ・セットを宛先として指定します • queue - 宛先として IBM MQ キューを指定します • filespace - 宛先としてファイル・スペースを指定します <p>値 queue が有効なのは、<destination> エレメントに子エレメント <queue> がある場合に限られます。</p> <p>値 filespace が有効なのは、<destination> エレメントに子エレメント <filespace> がある場合に限られます。</p>

属性	説明
	その他の値は、 <destination> エlementに子Element <file> がある場合に限り有効です。
exist	宛先システムに宛先ファイルが存在する場合に取る処置を示します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • error - エラーを報告し、ファイルは転送されません。 • overwrite - 既存の宛先ファイルを上書きします。 <destination> Elementに子Element <queue> または <filespace> がある場合は、この属性が無効になります。

<file>

前述の **<destination>** Elementの追加設定を指定します。ご使用のオペレーティング・システムと整合するフォーマットの完全修飾パスを使用します (例えば C:/from/here.txt)。ファイル URI は使用しないでください。

属性	説明
別名	<destination> ファイルの別名を指定します。この別名は、転送で指定されているディレクトリー・パスを除外したソース・ファイルの名前です。
encoding	テキスト・ファイル転送の <destination> ファイルのエンコード。
EOL	テキスト転送の行末マーカを指定します。有効な値は次のとおりです。 <ul style="list-style-type: none"> • LF - 改行文字のみ • CRLF - 復帰と改行の文字シーケンス
truncateRecords	オプション。LRECL データ・セット属性よりも長い <destination> レコードが切り捨てられることを指定します。 <ul style="list-style-type: none"> • true - それらのレコードは切り捨てられます • false - それらのレコードは折り返されます デフォルト設定は false です。

<queue>

<destination> Elementと一緒に使用する場合には、転送先のキューの名前を指定します。そのキューは、宛先エージェントのキュー・マネージャーに接続しているいずれかのキュー・マネージャーに存在していなければなりません。QUEUE@QM というフォーマットを使用してください (QUEUE は、メッセージの書き込み先のキューの名前、QM は、そのキューが存在しているキュー・マネージャーです)。**<queue>** Elementを **<source>** Element内で使用した場合、**<destination>** Element内では使用できません。

属性	説明
delimiter	ファイルを複数のメッセージに分割するための区切り文字。
delimiterType	区切り文字のタイプを指定します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • text - Java 正規表現 • binary - 16 進数バイトのシーケンス • size - バイト数、キビバイト数、メビバイト数のいずれか。例えば、1 B、1 K、または 1 M のようにします。

属性	説明
delimiterPosition	区切り文字が個々のメッセージ内で、データの前に組み込まれると想定されているか、データの後ろに組み込まれると想定されているかを指定します。有効なオプションは以下のとおりです。 <ul style="list-style-type: none"> • prefix - 区切り文字はデータの前にあると想定されています • postfix - 区切り文字はデータの後ろにあると想定されています
includeDelimiterInMessage	ファイルを複数のメッセージに分割するために使用した区切り文字をそれらのメッセージの末尾に組み込むかどうかを指定するブール値。
encoding	宛先キューのエンコード方式を指定します。
persistent	メッセージが永続メッセージであるかどうかを指定します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • true - メッセージは永続メッセージです • false - メッセージは永続メッセージではありません • qdef - メッセージの永続性の値を宛先キューの設定で定義します
setMqProps	ファイルの最初のメッセージで IBM MQ メッセージ・プロパティを設定するかどうか、およびエラーの発生時にキューにメッセージを書き込むかどうかを指定するブール値。
unrecognisedCodePage	宛先キュー・マネージャーがデータのコード・ページを認識できない場合に、テキスト・モードの転送が失敗するか、それとも変換が実行されるかを指定します。有効な値は以下のとおりです。 <ul style="list-style-type: none"> • fail - 転送は失敗を報告します。 • binary - データは宛先コード・ページに変換され、データの形式を記述する IBM MQ メッセージ・ヘッダーは MQFMT_NONE に設定されます。 デフォルトの動作は fail です。

<filespace>

転送先のファイル・スペースの名前を指定するグループ・エレメント。

<名前>

<filespace> エレメントと一緒に使用する場合には、ファイル・スペースの名前を値として指定します。

<preSourceCall>

転送のソースで転送の開始前に呼び出すコマンドを指定するグループ・エレメント。

<postSourceCall>

転送のソースで転送の完了後に呼び出すコマンドを指定するグループ・エレメント。

<preDestinationCall>

転送の宛先で転送の開始前に呼び出すコマンドを指定するグループ・エレメント。

<postDestinationCall>

転送の宛先で転送の完了後に呼び出すコマンドを指定するグループ・エレメント。

<コマンド>

<preSourceCall>、<postSourceCall>、<preDestinationCall>、<postDestinationCall> のいずれかのエレメントと一緒に使用する場合には、呼び出すコマンドを指定します。このコマンドは、エージェント・コマンド・パスに存在していなければなりません。詳しくは、[552 ページの表 45](#) を参照してください。

属性	説明
名前	実行するコマンドの名前。

属性	説明
successRC	コマンドが正常に実行された場合に返される戻りコード。

<argument>

<command> エlementと一緒に使用する場合には、コマンドに渡す引数を指定します。 <command> エlementの中に <argument> エlementをいくつでも記述できます。

<job>

転送仕様全体のジョブ情報を含むオプションのグループ・Element。 <job> は、転送が開始されたときに、ログ・メッセージに追加されるユーザー定義ジョブ名 ID です。 この<job> Elementは、転送要求メッセージに含まれる <job> Elementと同じです。 [748 ページの『ファイル転送ログ・メッセージ・フォーマット』](#) のトピックを参照してください。

<名前>

<job> Elementと一緒に使用する場合には、ジョブの名前を値として指定します。

<transferSpecifications>

複数の転送グループの <item> Elementを含むグループ・Element。 このElementの使用方法の詳細については、『[転送定義ファイルの使用](#)』を参照してください。

<cancel>

進行中のファイル転送を取り消すために必要なすべてのElementが含まれるグループ・Element。

属性	説明
バージョン	Managed File Transfer によって指定された、このElementのバージョンを示します。

<transfer>

<cancel> Elementと一緒に使用する場合、このElementの値は取り消す転送要求 ID を示します。

<job>

ジョブ情報が入ったグループ・Element。

<jobName>

論理ジョブ ID を指定します。

ファイル転送取り消しメッセージ・フォーマット

ファイル転送要求は、特定のエージェントに対する転送を識別する 48 文字の ID を返します。 この ID は転送を取り消すために使用されます。

転送取り消しメッセージについて

転送取り消しメッセージで使用されるElementと属性について説明します。

<cancel>

進行中のファイル転送を取り消すために必要なすべてのElementが含まれるグループ・Element。

属性	説明
バージョン	Managed File Transfer によって指定された、このElementのバージョンを示します。

<originator>

要求の発信元を示すElementが含まれるグループ・Element。

<hostName>

ソース・ファイルが置かれているシステムのホスト名。

<userID>

ファイル転送を開始したユーザーの ID。

<mqmdUserID>

オプション。メッセージ記述子 (MQMD) に指定された IBM MQ ユーザー ID。

<transfer>

<cancel> エlementと一緒に使用する場合は、このElementの値は取り消す転送要求 ID を示します。

<job>

オプション。ジョブ情報が入ったグループ・Element。

<jobName>

論理ジョブ ID を指定します。

例

このスキーマに準拠する XML メッセージの例が、以下の要求用に提供されています。

- [ファイル転送の作成](#)
- [非同期ファイル転送の作成要求](#)
- [ファイル転送を取り消す](#)
- [スケジュール済み転送の作成](#)
- [スケジュール済み転送の削除](#)
- [管理対象呼び出しの作成](#)
- [管理対象呼び出しを含むファイル転送の作成](#)

ファイル転送要求メッセージの例

エージェントによる転送の作成または取り消しを要求するために、エージェントのコマンド・キューに PUT できるメッセージの例。

転送作成要求

```
<?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>
```

転送作成要求 - 同期

ユーザーがブロック化同期要求を行う場合、すなわち、転送が完了するのを待って状況メッセージを受け取る場合、コマンド・キューに置かれたメッセージには、応答メッセージの送信先キューを指定した応答Elementが含まれています。以下の例は、FTEAGENT が使用するコマンド・キューに置かれたメッセージを示しています。

```
<?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>

```

<reply> エレメントには、転送の正常終了（または異常終了）についての応答を受け取るための一時動的キューが作成されたコマンド・キュー・マネージャーの名前が指定されています。一時動的キューの名前は2つの部分で構成されています。

- `command.properties` 構成ファイル (WMQFTE) 内のキー **dynamicQueuePrefix** によって定義される接頭部。デフォルト)
- IBM MQ が生成したキューの ID

転送取り消し要求

```

<?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>

```

関連資料

780 ページの『[ファイル転送要求メッセージ・フォーマット](#)』

ファイル転送は、エージェントのコマンド・キューに到着した XML メッセージによって開始されます。通常、ユーザーがファイル転送コマンドを発行したか、IBM MQ Explorer を使用した結果、XML メッセージが到着します。転送要求 XML は、FileTransfer.xsd スキーマに準拠していて、<request> エレメントをルート・エレメントとして持っている必要があります。FileTransfer.xsd スキーマ文書は `MQ_INSTALLATION_PATH/mqft/samples/schema` ディレクトリーにあります。FileTransfer.xsd スキーマは、同じディレクトリーにある `fteutils.xsd` をインポートします。

スケジュール済みファイル転送メッセージの例

エージェントによるスケジュールの作成または削除を要求するために、エージェントのコマンド・キューに PUT できるメッセージの例。

スケジュール済み転送の作成

```

<?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>

```

スケジュール済み転送の削除

```

<?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>

```

関連資料

780 ページの『[ファイル転送要求メッセージ・フォーマット](#)』

ファイル転送は、エージェントのコマンド・キューに到着した XML メッセージによって開始されます。通常、ユーザーがファイル転送コマンドを発行したか、IBM MQ Explorer を使用した結果、XML メッセージが到着します。転送要求 XML は、FileTransfer.xsd スキーマに準拠していて、<request> エレメントをルート・エレメントとして持っている必要があります。FileTransfer.xsd スキーマ文書は MQ_INSTALLATION_PATH/mqft/samples/schema ディレクトリーにあります。FileTransfer.xsd スキーマは、同じディレクトリーにある fteutils.xsd をインポートします。

MFT エージェント呼び出し要求メッセージの例

エージェント・コマンド・キューに置いて、そのエージェントが管理呼び出しを作成するか、プログラムを呼び出す転送を作成するよう要求するメッセージの例。

管理対象呼び出し要求の例

```

<?xml version="1.0" encoding="UTF-8"?>
<request xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="1.00"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedCall>
    <originator>
      <hostName>example.com.</hostName>
      <userID>mqm</userID>
    </originator>
    <agent agent="DNWE" QMgr="QM1"/>
    <transferSet>
      <call>
        <command name="echo" successRC="0">
          <argument>call</argument>
          <argument>test</argument>
        </command>
      </call>
    </transferSet>
  </managedCall>
</request>

```

```

        </command>
      </call>
    </transferSet>
  </job>
  <name>managedCallCalls.xml</name>
</job>
</managedCall>
</request>

```

呼び出しを使用したマネージド転送要求の例

```

<?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>

```

関連概念

[250 ページの『MFT で実行するプログラムの指定』](#)

Managed File Transfer Agent が実行されているシステムで、プログラムを実行することができます。ファイル転送要求の一部として、転送の開始前または終了後のいずれかにプログラムを実行するように指定することができます。また、管理対象呼び出し要求を実行依頼することで、ファイル転送要求に含まれないプログラムを開始することも可能です。

関連資料

[780 ページの『ファイル転送要求メッセージ・フォーマット』](#)

ファイル転送は、エージェントのコマンド・キューに到着した XML メッセージによって開始されます。通常、ユーザーがファイル転送コマンドを発行したか、IBM MQ Explorer を使用した結果、XML メッセージが到着します。転送要求 XML は、FileTransfer.xsd スキーマに準拠していて、<request> エレメントをルート・エレメントとして持っている必要があります。FileTransfer.xsd スキーマ文書は MQ_INSTALLATION_PATH/mqft/samples/schema ディレクトリーにあります。FileTransfer.xsd スキーマは、同じディレクトリーにある fteutils.xsd をインポートします。

MFT モニター要求メッセージ・フォーマット

リソース・モニターは、通常はユーザーによる `fteCreateMonitor` コマンドの発行または IBM MQ Explorer・インターフェースの使用の結果、適切な XML メッセージがエージェントのコマンド・キューに到達すると作成されます。

モニター XML は、ルート・エレメントとして `<monitor>` エレメントを使用する `Monitor.xsd` スキーマに準拠している必要があります。

モニター・メッセージには、以下の 3 つのルート・エレメントのいずれかを含むことができます。

- `<monitor>` - 新規リソース・モニターの作成および開始用
- `<deleteMonitor>` - 既存モニターの停止および削除用

`fteListMonitors` コマンドは、直接 `SYSTEM.FTE` トピックから一致するモニター定義を取り出すため、このコマンドのコマンド・メッセージはありません。

スキーマ

次のスキーマは、モニター要求 XML メッセージで有効なエレメントについて示します。

```
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema"
            targetNamespace="https://www.ibm.com/xmlns/wmqfte/7.0.1/
MonitorDefinition"
            xmlns="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition">
  <xsd:include schemaLocation="FileTransfer.xsd"/>
  <xsd:element name="monitor">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="name" type="monitorNameType"
                    minOccurs="1" maxOccurs="1"/>
        <xsd:element name="description" type="xsd:string"
                    minOccurs="0" maxOccurs="1"/>
        <xsd:element name="pollInterval" type="pollIntervalType"
                    minOccurs="1" maxOccurs="1"
                    default="10"/>
        <xsd:element name="batch" type="batchType"
                    minOccurs="0" maxOccurs="1"/>
        <xsd:element name="agent" type="agentNameType"
                    minOccurs="1" maxOccurs="1"/>
        <xsd:element name="resources" type="monitorResourcesType"
                    minOccurs="0" maxOccurs="1"/>
        <xsd:element name="triggerMatch" type="triggerMatchType"
                    maxOccurs="1" minOccurs="1"/>
        <xsd:element name="reply" type="replyType"
                    maxOccurs="1" minOccurs="0"/>
        <xsd:element name="tasks" type="monitorTasksType"
                    maxOccurs="1" minOccurs="1"/>
        <xsd:element name="originator" type="origRequestType"
                    maxOccurs="1" minOccurs="1"/>
        <xsd:element name="job" type="jobType"
                    maxOccurs="1" minOccurs="0"/>
        <xsd:element name="defaultVariables" type="defaultVariablesType"
                    maxOccurs="1" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="version" type="versionType" use="required"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="deleteMonitor">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="name" type="monitorNameType"
                    minOccurs="1" maxOccurs="1"/>
        <xsd:element name="originator" type="origRequestType"
                    maxOccurs="1" minOccurs="1"/>
        <xsd:element name="reply" type="replyType"
                    maxOccurs="1" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="version" type="versionType" use="required"/>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

```

<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:choice>
  </xsd:sequence>
</xsd:complexType>

```

```

        <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: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>

```

モニター作成メッセージの理解

モニター作成メッセージで使用されるエレメントと属性について説明します。

エレメントの説明

<monitor>

進行中のファイル転送を取り消すために必要なすべてのエレメントが含まれるグループ・エレメント。

属性	説明
バージョン	Managed File Transfer によって指定された、このエレメントのバージョンを示します。

<名前>

モニターのエージェント内で固有のモニター名。

<description>

モニターの説明 (現在は使用されていません)。

<pollInterval>

トリガー条件に照らし合わせるリソースの各検査間の時間間隔。

属性	説明
単位	ポーリング間隔の時間単位を指定します。有効な値は次のとおりです。 <ul style="list-style-type: none"> • seconds • minutes • hours • 日 • weeks • months • years

<agent>

モニターが関連付けられているエージェントの名前。

<resources>

モニター対象リソースを指定するエレメントが含まれているグループ・エレメント。

<directory>

モニター対象であるモニターのエージェント・マシン上のディレクトリーを指定する完全修飾パス。

属性	説明
recursionLevel	指定したディレクトリーに加え、モニター対象となるサブディレクトリーの数。
ID	リソースの固有 ID。

<queue>

モニター・エージェントのキュー・マネージャーでモニターするキューを指定するキュー名。

<triggerMatch>

モニター対象リソースと比較するトリガー条件を指定するエレメントが含まれているグループ・エレメント。

<conditions>

モニター対象リソースと比較する条件のタイプを指定するエレメントが含まれているグループ・エレメント。

<allOf>

含まれているすべての条件が満たされる必要があることを指定する述部。

<anyOf>

含まれている任意の条件が満たされる必要があることを指定する述部。

<condition>

モニター・トリガー条件全体に与えられる比較条件を定義します。

<名前>

条件の名前。

<resource>

条件を比較する際の基準となるリソース定義を指定します。

属性	説明
ID	リソースの固有 ID。

モニター対象のリソースがディレクトリーである場合は、以下の3つのエレメントのいずれかを条件で指定する必要があります。

- fileMatch
- fileNoMatch
- fileSize

モニター対象のリソースがキューである場合は、以下の2つのエレメントのいずれかを条件で指定する必要があります。

- queueNotEmpty
- completeGroups

<fileMatch>

ファイル名マッチング条件のグループ・エレメント。

<pattern>

ファイル名マッチング・パターンを指定します。条件を満たすには、リソース上のファイルがこのパターンと一致している必要があります。デフォルト・パターンは、* (任意のファイルが一致) です。

<fileNoMatch>

ファイル名逆マッチング条件のグループ・エレメント

<pattern>

ファイル名逆マッチング・パターンを指定します。モニター対象リソースに一致するファイルがない場合に、この条件が満たされます。デフォルト・パターンは、* (ファイルがまったくない場合に一致) です。

<fileSize>

ファイル・サイズ比較のグループ・エレメント。

<compare>

ファイル・サイズ比較を指定します。値は負以外の整数でなければなりません。

属性	説明
operator	使用する比較演算子。「>=」のみがサポートされています。
単位	ファイル・サイズ単位を指定します。以下のいずれかです。 <ul style="list-style-type: none"> • B - バイト • KB - キロバイト • MB - メガバイト • GB - ギガバイト 単位の値では、大/小文字は区別されません。したがって、「mb」は「MB」としても機能します。

<pattern>

一致するファイル名パターン。デフォルトは、*(任意のファイルが一致)です。

<queueNotEmpty>

リソースがキューの場合にのみ指定できます。モニターを起動するためにはキューにメッセージが入っていないなければならない、という動作を指定します。

<completeGroups>

リソースがキューの場合にのみ指定できます。モニターを起動するためには完全に揃ったメッセージ・グループがキューに存在していなければならない、という動作を指定します。キューに存在する完全に揃ったグループごとに1つの転送タスクが実行されます。

<reply>

非同期要求の応答キューを指定するために使用されるエレメント(オプション)。

属性	説明
QMGR	キュー・マネージャー名。

<tasks>

モニター・トリガー条件が満たされた場合に呼び出すタスクを指定するエレメントが含まれているグループ・エレメント。

<task>

トリガー条件が満たされた場合にモニターが呼び出す個々のタスクを定義するグループ・エレメント。現時点で、指定できるのは1つのタスクのみです。

<名前>

タスクの名前。任意の英数字を受け入れます。

<description>

タスクの説明。任意のテキスト値を使用できます。

<transfer>

転送タスクを定義するグループ・エレメント。

<request>

タスクのタイプを定義するグループ・エレメント。これには、FileTransfer.xsdスキーマ定義から継承される次のエレメントのいずれかが含まれていなければなりません。

- [managedTransfer](#)
- managedCall

属性	説明
バージョン	Managed File Transfer から提供される、この要求のバージョン。これは、フォーマット n.mm で表されます。ここで、n はメジャー・リリース・バージョン、mm はマイナー・バージョンです。例: 1.00。

<originator>

要求の発信元を示すエレメントが含まれるグループ・エレメント。

<hostName>

ソース・ファイルが置かれているシステムのホスト名。

<userID>

ファイル転送を開始したユーザーの ID。

<mqmdUserID>

オプション。メッセージ記述子 (MQMD) に指定された IBM MQ ユーザー ID。

<job>

ジョブ情報が入ったグループ・エレメント。

<jobName>

論理ジョブ ID を指定します。

<defaultVariables>

1つ以上の variable エレメントを含むグループ・エレメント。これらの変数は、キューをモニターする際に変数置換で使用されます。変数置換について詳しくは、[180 ページの『変数置換を使用した MFT リソース・モニター・タスクのカスタマイズ』](#)を参照してください。

<variable>

key 属性によって与えられるキーに関連付けられた値を含むエレメント。

属性	説明
key	デフォルトの変数の名前。

モニター削除メッセージの理解

モニター削除メッセージで使用されるエレメントと属性について説明します。

エレメントの説明

<deleteMonitor>

モニターの停止および削除に必要なすべてのエレメントが含まれるグループ・エレメント。

属性	説明
バージョン	Managed File Transfer によって指定された、このエレメントのバージョンを示します。

<名前>

削除するモニターの名前。

<originator>

要求の発信元を示すエレメントが含まれるグループ・エレメント。

<hostName>

ソース・ファイルが置かれているシステムのホスト名。

<userID>

ファイル転送を開始したユーザーの ID。

<mqmdUserID>

オプション。メッセージ記述子 (MQMD) に指定された IBM MQ ユーザー ID。

<reply>

要求用に生成された一時応答キューの名前を示します。キューの名前は、`command.properties` 構成ファイル内のキー `dynamicQueuePrefix` によって定義されているとおりです。これが指定されていない場合には、キュー名はデフォルト値の `WMQFTE` になります。

属性	説明
QMGR	応答を受け取るために一時動的キューが生成されるコマンド・キュー・マネージャーの名前。

例

このスキーマに準拠する XML メッセージの例が、以下のモニター要求用に提供されています。

- [モニターの作成](#)
- [モニターの削除](#)

MFT モニター要求メッセージの例

エージェントによるモニターの作成または削除を要求するために、エージェントのコマンド・キューに PUT できるメッセージの例。

モニター作成要求

```
<?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>
</monitor>
```

```

    </task>
  </tasks>
  <originator>
    <hostName>example.com.</hostName>
    <userID>mqm</userID>
  </originator>
</monitor:monitor>

```

モニター削除要求

```

<?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>

```

関連資料

797 ページの『MFT モニター要求メッセージ・フォーマット』

リソース・モニターは、通常はユーザーによる `fteCreateMonitor` コマンドの発行または IBM MQ Explorer・インターフェースの使用の結果、適切な XML メッセージがエージェントのコマンド・キューに到達すると作成されます。

Ping MFT エージェント要求メッセージ・フォーマット

`ftePingAgent` コマンドを発行するか、XML メッセージをエージェントのコマンド・キューに PUT することにより、エージェントを ping できます。ping エージェント要求の XML は、`PingAgent.xsd` スキーマに準拠する必要があります。Managed File Transfer のインストール後には、`PingAgent.xsd` スキーマ・ファイルが `MQ_INSTALLATION_PATH/mqft/samples/schema` ディレクトリーに配置されます。`PingAgent.xsd` スキーマは、同じディレクトリーにある `fteutils.xsd` をインポートします。

エージェントがアクティブの場合に、コマンド・キュー上の ping エージェント要求メッセージを受け取ると、そのエージェントは、ping エージェント要求メッセージをコマンド行に PUT したコマンドまたはアプリケーションに XML 応答メッセージを返します。エージェントからの応答メッセージは、`Reply.xsd` で定義されたフォーマットです。このフォーマットに関して詳しくは、808 ページの『MFT エージェント応答メッセージ・フォーマット』を参照してください。

スキーマ

次のスキーマは、ping エージェント要求 XML メッセージで有効なエレメントについて示します。

```

<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>

```

ping エージェント要求メッセージについて

ping エージェント要求メッセージで使用するエレメントと属性を以下のリストにまとめます。

<pingAgent>

ping エージェント要求を指定するために必要なすべてのエレメントが含まれるグループ・エレメント。

<originator>

ping 要求の送信元を指定するために必要なすべてのエレメントが含まれるグループ・エレメント。

<hostName>

要求の送信元であるマシンのホスト名。

<userID>

要求の送信元のユーザー名。

<mqmdUserID>

要求の送信元の MQMD ユーザー名。

<agent>

ping 対象のエージェント。

属性	説明
エージェント	必須。 エージェントの名前。
キュー・マネージャー	オプション。 エージェントの接続先キュー・マネージャー。

<reply>

応答メッセージの送信先エージェントのキューの名前。

属性	説明
QMGR	必須。 応答キューがあるキュー・マネージャーの名前。

例

この例は、エージェント AGENT_JUPITER に送信された ping エージェント・メッセージを示しています。AGENT_JUPITER がアクティブで、エージェント要求を処理できる場合、このエージェントは応答メッセージを QM_JUPITER 上にあるキュー WMQFTE.4D400F8B20003708 に送信します。

```
<?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 エージェント応答メッセージ・フォーマット

エージェントがエージェント・コマンド・キューで XML メッセージを受け取ったとき、応答が必要な場合は、オリジナル・メッセージで定義された応答キューにエージェントが XML 応答メッセージを送信します。応答 XML は Reply.xsd スキーマに準拠しています。Reply.xsd スキーマ文書は MQ_INSTALLATION_PATH/mqft/samples/schema ディレクトリーにあります。Reply.xsd スキーマは、同じディレクトリーにある fteutils.xsd をインポートします。

スキーマ

次のスキーマは、応答 XML メッセージで有効なエレメントについて示します。

```
<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
```


これらのメッセージについて詳しくは、IBM Documentation: https://www.ibm.com/docs/SSFKSJ_9.0.0/com.ibm.wmqfte.doc/messages_main.html を参照してください。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒 103-8510

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ビー・エム株式会社

日本アイ・ビー・エム株式会社

法務・知的財産

U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law

〒 103-8510

103-8510

東京 103-8510、日本

以下の保証は、国または地域の法律に沿わない場合は、適用されません。 INTERNATIONAL BUSINESS MACHINES CORPORATION は、法律上の瑕疵担保責任、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。"" 国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ビー・エム株式会社

Software Interoperability Coordinator, Department 49XA

3605 Highway 52 N

Rochester, MN 55901

U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っていません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名前はすべて架空のものであり、名前や住所が類似する個人や企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほめかしたり、保証することはできません。

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

プログラミング・インターフェース情報

プログラミング・インターフェース情報 (提供されている場合) は、このプログラムで使用するアプリケーション・ソフトウェアの作成を支援することを目的としています。

本書には、プログラムを作成するユーザーが WebSphere MQ のサービスを使用するためのプログラミング・インターフェースに関する情報が記載されています。

ただし、この情報には、診断、修正、および調整情報が含まれている場合があります。診断、修正、調整情報は、お客様のアプリケーション・ソフトウェアのデバッグ支援のために提供されています。

重要: この診断、修正、およびチューニング情報は、変更される可能性があるため、プログラミング・インターフェースとして使用しないでください。

商標

IBM、IBM ロゴ、ibm.com[®]は、世界の多くの国で登録された IBM Corporation の商標です。現時点での IBM の商標リストについては、"Copyright and trademark information" www.ibm.com/legal/copytrade.shtml をご覧ください。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。

Microsoft および Windows は、Microsoft Corporation の米国およびその他の国における商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

この製品には、Eclipse Project (<http://www.eclipse.org/>) により開発されたソフトウェアが含まれています。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。



部品番号:

(1P) P/N: