







お願い

本書および本書で紹介する製品をご使用になる前に、201 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 WebSphere Adapter for FTP バージョン 6、リリース 1、モディフィケーション 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。

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します。

IBM 発行のマニュアルに関する情報のページ

<http://www.ibm.com/jp/manuals/>

こちらから、日本語版および英語版のオンライン・ライブラリーをご利用いただけます。また、マニュアルに関するご意見やご感想を、上記ページよりお送りください。今後の参考にさせていただきます。

(URL は、変更になる場合があります)

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原 典： WebSphere® Adapters  
WebSphere Adapter for FTP User Guide Version 6 Release 1  
Version 6 Release 1

発 行： 日本アイ・ビー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

# 目次

<b>第 1 章 WebSphere Adapter for FTP の概要</b>	<b>1</b>
このリリースの新機能	2
ハードウェア要件とソフトウェア要件	3
Adapter for FTP の技術概要	4
Outbound 処理	4
Inbound 処理	11
ビジネス・オブジェクト	19
外部サービス・ウィザード	21
標準規格の準拠	21
アクセシビリティ	21
インターネット・プロトコル・バージョン 6 (IPv6)	22
<b>第 2 章 アダプター実装の計画</b>	<b>23</b>
始める前に	23
セキュリティ	23
Secure Sockets Layer の構成	23
連邦情報処理標準 140 に対応するアダプターの構成	26
ユーザー認証	27
デプロイメント・オプション	28
クラスター環境での WebSphere Adapters	31
バージョン 6.1.0 へのマイグレーション	33
マイグレーションに関する考慮事項	33
マイグレーションの実行	34
バージョン 6.0.2 プロジェクトをマイグレーションせずに更新する	36
<b>第 3 章 サンプルおよびチュートリアル</b>	<b>37</b>
<b>第 4 章 デプロイメント用のモジュールの構成</b>	<b>39</b>
モジュールの構成のためのロードマップ	39
認証別名の作成	41
モジュールの作成	43
ビジネス・オブジェクトの定義	45
アダプター・パターン・ウィザードによる単純サービスの作成	47
プロジェクトの作成	54
Outbound 処理のモジュールの構成	57
デプロイメント・プロパティおよびランタイム・プロパティの設定	57
データ・タイプおよび操作名の選択	60
データ・バインディングの構成	63
データ・ハンドラーの構成	67
対話仕様プロパティの設定およびサービスの生成	72
Inbound 処理のモジュールの構成	75

デプロイメント・プロパティおよびランタイム・プロパティの設定	75
データ・タイプおよび操作名の選択	79
データ・バインディングの構成	81
データ・ハンドラーの構成	85
サービスの生成	88

## 第 5 章 アセンブリー・エディターを使用した対話仕様プロパティの変更 . . . . . 91

## 第 6 章 モジュールのデプロイ . . . . . 93

デプロイメント環境	93
テスト用モジュールのデプロイ	93
Inbound 処理テスト用のターゲット・コンポーネントの生成および配線	93
サーバーへのモジュールの追加	96
テスト・クライアントを使用したモジュールの Outbound 処理テスト	97
実動用モジュールのデプロイ	97
RAR ファイルのインストール (スタンドアロン・アダプターを使用するモジュールの場合のみ)	97
EAR ファイルとしてのモジュールのエクスポート	99
EAR ファイルのインストール	101

## 第 7 章 アダプター・モジュールの管理 103

組み込みアダプターの構成プロパティの変更	103
組み込みアダプターのリソース・アダプター・プロパティの設定	103
組み込みアダプターの管理 (J2C) 接続ファクトリー・プロパティの設定	105
組み込みアダプターのアクティベーション・スペック・プロパティの設定	107
スタンドアロン・アダプターの構成プロパティの変更	109
スタンドアロン・アダプターのリソース・アダプター・プロパティの設定	109
スタンドアロン・アダプターの管理 (J2C) 接続ファクトリー・プロパティの設定	110
スタンドアロン・アダプターのアクティベーション・スペック・プロパティの設定	112
アダプターを使用するアプリケーションの開始	114
アダプターを使用するアプリケーションの停止	114
Performance Monitoring Infrastructure を使用したパフォーマンスのモニター	115
Performance Monitoring Infrastructure の構成	115
Common Event Infrastructure (CEI) によるトレースの使用可能化	118
パフォーマンスに関する統計の表示	118
トラブルシューティングおよびサポート	120
ロギングおよびトレースの構成	120
First Failure Data Capture (FFDC) のサポート	123

ビジネス・フォールト . . . . .	123	対話仕様プロパティ . . . . .	154
XAResourceNotAvailableException . . . . .	128	Inbound 構成プロパティ . . . . .	164
org.xml.sax.SAXParseException . . . . .	129	アダプター・タイプ・プロパティ . . . . .	166
セルフ・ヘルプ・リソース . . . . .	129	リソース・アダプター・プロパティ . . . . .	168
<b>第 8 章 参照情報 . . . . .</b>	<b>131</b>	アクティベーション・スペック・プロパティ . . . . .	171
ビジネス・オブジェクトの情報 . . . . .	131	グローバルゼーション . . . . .	193
ビジネス・オブジェクトの構造 . . . . .	131	グローバルゼーションおよび双方向変換 . . . . .	193
命名規則 . . . . .	135	双方向データ変換で使用可能なプロパティ . . . . .	196
ビジネス・オブジェクト属性プロパティ . . . . .	136	アダプター・メッセージ . . . . .	197
ビジネス・オブジェクトの操作サポート . . . . .	136	関連情報 . . . . .	198
カスタム・ビジネス・オブジェクト . . . . .	137	<b>特記事項 . . . . .</b>	<b>201</b>
カスタム・ファイル分割 . . . . .	137	プログラミング・インターフェース情報 . . . . .	203
Outbound 構成プロパティ . . . . .	138	商標 . . . . .	203
アダプター・タイプ・プロパティ . . . . .	140	<b>索引 . . . . .</b>	<b>205</b>
リソース・アダプター・プロパティ . . . . .	142		
管理 (J2C) 接続ファクトリー・プロパティ . . . . .	145		

## 第 1 章 WebSphere Adapter for FTP の概要

WebSphere Adapter for FTP を使うと、FTP サーバーが管理するファイルにアクセスするために WebSphere Process Server および WebSphere Enterprise Service Bus を使用する統合プロセスを、FTP 通信またはプロトコルの詳細を認識せずに作成することができます。

構成が完了すると、サービス指向アーキテクチャー (SOA) 実装におけるサービス・プロバイダーのようにアダプターが機能し、ファイルの送信および取得の操作を提供します。アダプターは、WebSphere Process Server または WebSphere Enterprise Service Bus にデプロイされるモジュールの一部です。

アダプターは、サービス・インターフェースを公開することによって、データの取得方法または操作の実行方法を隠蔽します。モジュールの外部のサービスは、FTP サーバーと直接対話せずにアダプターと対話するため、モジュールのセットアップ時に指定された認証の詳細 (ユーザー名、パスワードなど) は、モジュール外部のサービスから保護されます。

これにはどのような利点があるでしょうか。モジュールは、特定の Inbound サービスまたは Outbound サービスを実行するように設計された再利用可能な単位であり、WebSphere Integration Developer の外部サービス・ウィザードを使用して作成されます。各モジュールは一貫性のあるインターフェースおよび標準のビジネス・オブジェクトを使用するので、サービスを利用するアプリケーションは FTP サーバーの詳細を深いレベルまで理解する必要はありません。

次の図は、SOA 実装の一部としてアダプターがどのように機能するのかを示しています。

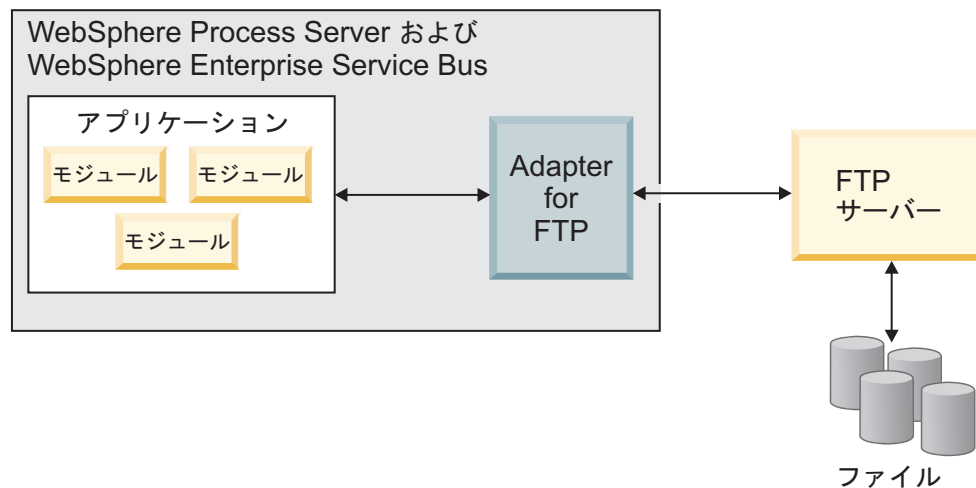


図 1. アダプターの概要

---

## このリリースの新機能

製品のフィーチャーと機能について、新規のものや拡張されたものについて説明します。

この情報に関する最新情報は、WebSphere Adapters 製品サポート Web サイトから入手できます。最新情報または追加情報を確認するには、<http://www.ibm.com/software/integration/wbiadapters/support/> を参照してください。

バージョン 6.1.0 の新機能:

- エンタープライズ・サービス・ディスカバリー・ウィザードの変更

ウィザードの名前が、外部サービス・ウィザードに変更されました。また、各種機能と使いやすさが向上し、アダプターで使用するサービスの構築が簡単に行えるようになりました。このウィザードを使うと、定義済みのデータ・バインディング、データ・ハンドラー、および関数セクターにアクセスして、ファイルとビジネス・オブジェクト間の変換を自動的に行わせることができます。

このウィザードでは、さまざまなプロパティにおいてデフォルト値が設定されています。これによって、特定の情報を簡単に入力できるようになり、どのプロパティが必須であるかを指定して、拡張プロパティを気にすることなくモジュールを構成することができます。

- ビジネス・グラフは、オプションになりました。バージョン 6.0.2 での各ビジネス・オブジェクトが含まれたビジネス・グラフが、オプションになりました。ビジネス・グラフは、モジュールのビジネス・オブジェクトがバージョン 6.0.2 で作成された場合のみ必要になります。
- ビジネス・フォールトのサポート

ビジネス例外用のビジネス・フォールトを、アダプターで生成できるようになりました。これにより、それらのエラー条件に修正アクションを容易に割り当てることができます。

- オペレーティング・システム・サポートが拡張されました。バージョン 6.1.0 でサポートされるオペレーティング・システムについて詳しくは、IBM Web サイト (<http://www.ibm.com/support/docview.wss?uid=swg27006249>) にある WebSphere Adapter for FTP のハードウェアおよびソフトウェアの要件を参照してください。
- WebSphere Adapter for FTP および関連する外部サービス・ウィザードの成果物をバージョン 6.0.2 からバージョン 6.1.0 にマイグレーションする作業が自動化されました。
- 単純サービスを作成する時間を短縮するために、アダプター・パターン・ウィザードが追加されました。詳しくは、47 ページの『アダプター・パターン・ウィザードによる単純サービスの作成』を参照してください。
- 6.0.2 ビジネス・オブジェクトとの後方互換性。
- Outbound Exists 操作で使用可能なサブフォルダー検索。
- Create、Append、および Overwrite 操作における固有ファイル名生成をサポート。
- Append および Overwrite 操作における FTP サーバー・ディレクトリーへのファイル作成をサポート。



- Retrieve 操作における FTP サーバー・ディレクトリーからのファイル削除をサポート。
- Outbound の Create 操作時のイベント・ファイル順序付けをサポート。
- Outbound の Retrieve 操作のファイル構文解析およびデータ変換をサポート。
- Secure Sockets Layer (SSL) および連邦情報処理標準 (FIPS) 140 をサポート。
- First Failure Data Capture (FFDC) の構成がサポートされています。これは、WebSphere Application Server 症状データベースに組み込むことができ、診断モジュールにおいて記録されているデータをカスタマイズする際の手助けとなる、情報と推奨アクションを提供します。
- ノード・レベルまたはスタンドアロンのアダプターのデプロイメントのサポート
- 双方向スクリプト処理のサポートの簡素化
- FTP スクリプト・ファイルにおけるパラメーター置換およびエラー処理をサポート。
- ロギング、トレース、およびモニターの機能はアダプターから外されました。
- アダプター RAR ファイルは WebSphere Integration Developer 内で使用可能です。つまり、アダプター RAR ファイルを別にインストールする必要はありません。このウィザードによって、アダプター・ファイルが自動的にプロジェクトにコピーされます。
- アダプターに関する資料は、WebSphere Integration Developer インフォメーション・センターのアダプターの構成および使用 (Configuring and using adapters) セクションにあります。

---

## ハードウェア要件とソフトウェア要件

WebSphere Adapters のハードウェア要件とソフトウェア要件は、以下の IBM Web サイトに記載されています。

WebSphere Adapters のハードウェア要件とソフトウェア要件: <http://www.ibm.com/support/docview.wss?uid=swg27006249>

### 追加情報

以下のリンク先には、アダプターの構成およびデプロイを行う際に必要となる追加情報が記載されています。

- WebSphere Business Integration Adapters および WebSphere Adapters の互換性マトリックスによって、ご使用のアダプターで必要となるソフトウェアのサポート対象バージョンが識別されます。この資料を表示するには、WebSphere Adapters のサポート・ページ (<http://www.ibm.com/software/integration/wbiadapters/support/>) にアクセスして、「**Planning upgrades**」の下にある互換性マトリックスのリンクをクリックしてください。
- WebSphere Adaptersの技術情報には、製品資料には記載されていない回避策および追加情報が書かれています。ご使用のアダプターの技術情報を表示するには、<http://www.ibm.com/support/search.wss?tc=SSMKUK&rs=695&rank=8&dc=DB520+D800+D900+DA900+DA800+DB560&dtm> にアクセスして、「**Product category**」リストの中からアダプター名を選択し、検索アイコンをクリックしてください。

## Adapter for FTP の技術概要

WebSphere Adapter for FTP は、WebSphere Process Server または WebSphere Enterprise Service Bus 上で実行されるサービスに、1 つ以上の FTP サーバーと通信するための方法を提供します。

サービスはモジュールに含まれ、モジュールは WebSphere Integration Developer のプロジェクト、および WebSphere Process Server に対するデプロイメントの単位の両方から構成されます。モジュールはパッケージ化され、エンタープライズ・アーカイブ (EAR) ファイルとして WebSphere Process Server にデプロイされます。

モジュールには、コンポーネント (実際のサービス、インポートおよびエクスポート) が含まれます。インポートがモジュールの外部にあるサービスを識別することにより、サービスはモジュール内部から呼び出せるようになります。エクスポートにより、モジュール内のコンポーネントは外部クライアントにサービスを提供できます。インポートおよびエクスポートを使用するには、モジュールからデータをトランスポートする方法を指定したバインディング情報が必要です。WebSphere Integration Developer のアセンブリ・エディターでは、インポートおよびエクスポートをセットアップし、サポートされるバインディングをリストして、インポートおよびエクスポートを容易に作成することができます。

## Outbound 処理

Adapter for FTP では、Outbound 要求処理をサポートしています。モジュールからビジネス・オブジェクトの形で送信された要求をアダプターが受信すると、要求を処理して、可能な場合はビジネス・オブジェクトの形で結果を返します。

以下の図は、WebSphere Adapter for FTP の Outbound 処理フローを示しています。

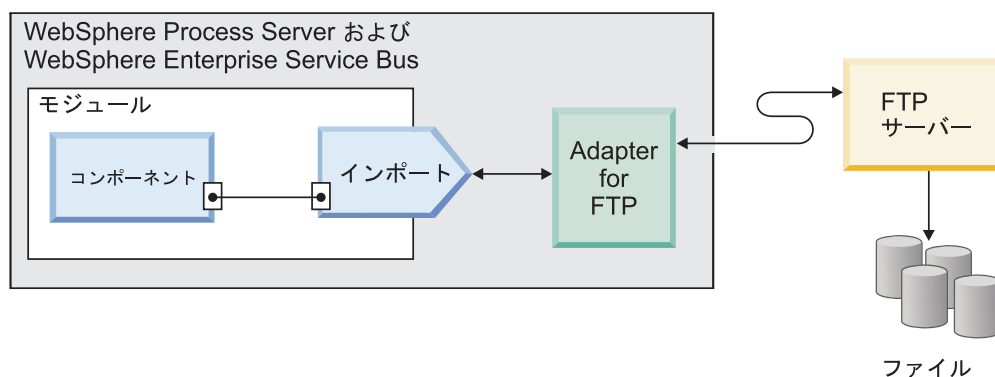


図2. Outbound 処理フロー

## Outbound データ変換

Outbound 通信時のデータ変換とは、アダプターがビジネス・オブジェクトをネイティブ・フォーマットで作成されたイベント・レコード (バイトまたはストリングなど) に変換するために使用するプロセスを指します。アダプターは、アダプター固有のデータ・バインディングおよびデータ・ハンドラーを使用してこの変換を行います。

データ変換によって、外部アプリケーションは、容易に理解および処理できる形式でデータを送受信できます。アダプターがビジネス・オブジェクト内の対応する属性からイベント・レコードを作成するために使用するデータ・バインディングおよびデータ・ハンドラーは、WebSphere Integration Developer の外部サービス・ウィザードを使用して構成されます。

## データ・バインディング

データ・バインディングは、ビジネス・オブジェクトのフィールドを読み取って、イベント・レコード内の対応するフィールドに書き込む役割があります。各データ・バインディングは、ビジネス・オブジェクトのフォーマット設定方法を定義したマップです。Adapter for FTPは、Outbound 通信時に FTPFileBaseDataBinding データ・バインディングを使用します。

Outbound 通信時に、データ・バインディングはビジネス・オブジェクトから以下のフィールドを取得して、イベント・レコード内の同等のフィールドに値を取り込みます。

- DirectoryPath
- Filename
- DataConnectionMode
- FileTransferType
- SecondServerDirectory
- SecondServerUsername
- SecondServerPassword
- IncludeEndBODElimiter
- FileInLocalDirectory
- LocalDirectoryPath
- LocalArchivingEnabledForCreate
- LocalArchiveDirForCreate
- StagingDirectory
- GenerateUniqueFile
- SplittingFunctionClassName
- SplitCriteria
- DeleteOnRetrieve
- ArchiveDirectoryForRetrieve
- FileContentEncoding

変換する必要のないデータの場合、データは変更されずにシステムを通過するため、アダプターはいわゆる「パススルー処理」を実行します。

## データ・ハンドラー

データ変換では、データ・バインディングのほかにデータ・ハンドラーも使用する必要があります。データ・ハンドラーは、ビジネス・オブジェクトとネイティブ・フォーマットの間の変換を実行します。WebSphere Adapter for FTP のバージョン 6.1.0 では、アダプターは以下のデータ・ハンドラーを提供します。

- XMLDataHandler
- WTXInvokerDataHandler
- WTXMapSelectionDataHandler

## サポートされる操作

操作は、Outbound 処理時に FTP サーバーを介してアクセス可能なリモート・ファイル・システムに対してアダプターが実行できるアクションの名前です。通常、操作の名前は、*Create* または *Append* など、アダプターが実行するアクションのタイプを示します。

WebSphere Adapter for FTP は、Outbound 処理時に以下の操作をサポートします。

表 1. サポートされる Outbound 操作

操作	結果
Create	<p>指定された名前のファイルが、FTP サーバー内の指定ディレクトリーに作成される。ファイルの内容は、要求の一部として提供されることもありますし、ローカル・ファイル・システムから取得されることもあります。</p> <p>作成するファイルが存在しない場合は、ファイルが作成された後に、ファイル名が呼び出し側コンポーネントに返されて、ファイル作成が成功したことが示されます。</p> <p>要求の一部としてファイル内容を受け取った場合、アダプター・マシンにファイルをアーカイブしてからファイルを作成するオプションが、アダプターによって提供されます。</p> <p>ファイルをステージング・ディレクトリーに作成してから、実際のディレクトリーに送信することができます。ステージング・ディレクトリーを指定しない場合、ファイルは実際のディレクトリーに直接作成されます。</p> <p>アダプターには、固有のファイル名を生成する機能があります。10 ページの『固有ファイル名の生成』を参照してください。</p> <p>アダプターには、作成された出力ファイルにファイル・シーケンスを作成する機能があります。9 ページの『Create 操作中のファイル・シーケンスの生成』を参照してください。</p> <p>作成するファイルがすでに存在する場合は、DuplicateRecord 例外が送信されて、ファイルは作成されません。既存のファイルも上書きされません。</p>

表 1. サポートされる Outbound 操作 (続き)

操作	結果
Append	<p>FTP サーバーの指定されたディレクトリーにある指定された名前のファイルが、要求の中で送信された内容で付加される。</p> <p>付加するファイルが存在する場合は、内容が付加され、ファイル名が呼び出し側コンポーネントに返されて、正常に対応したことが示されます。</p> <p>付加するファイルは、指定のディレクトリーから、実際のディレクトリー (存在する場合) にコピーされ、ステージング・ディレクトリー内のファイルに内容が付加されます。その後で、このファイルは実際のディレクトリーに戻されます。</p> <p>CreateIfFileNotExist プロパティーが true に設定されている場合、アダプターは新規ファイルを作成します。</p> <p>アダプターには、固有のファイル名を生成する機能があります。10 ページの『固有ファイル名の生成』を参照してください。</p> <p>付加するファイルが存在しない場合は、呼び出し側コンポーネントに RecordNotFound 例外が送信されます。</p>
Delete	<p>指定されたディレクトリー内のファイルが FTP サーバーで削除され、削除の成功を示す null をアダプターが呼び出し側コンポーネントに返す。</p> <p>削除するファイルが存在しない場合は、呼び出し側コンポーネントに RecordNotFound 例外が送信されます。</p>
Retrieve	<p>指定された要求のファイルの内容が返される。</p> <p>SplittingFunctionClassName プロパティーおよび SplitCriteria プロパティーに基づいて、ファイル内容が分割されます。構成済みのデータ・ハンドラーに基づいて、ファイル内容がビジネス・オブジェクトに変換されます。</p> <p>要求で指定されたファイルが存在する場合、そのファイルの内容が取得され、応答として送信されます。ファイル内容は、呼び出し側コンポーネントに返すこともできますし、ローカル・ファイル・システムに保存することもできます。</p> <p>アダプターは、ファイルの取得後に FTP サーバー・ディレクトリーからファイルを削除するオプションを提供します。これを行うには、DeleteOnRetrieve プロパティーを使用します。</p> <p>アダプターでは、ArchiveDirectoryForDeleteOnRetrieve プロパティーを使用して、FTP サーバーにファイルをアーカイブしてからファイルを削除するオプションをサポートしています。</p> <p>取得するファイルが存在しない場合は、呼び出し側コンポーネントに RecordNotFound 例外が送信されます。</p>

表 1. サポートされる Outbound 操作 (続き)

操作	結果
Overwrite	<p>ディレクトリーのファイルに要求で指定された内容を上書きする。</p> <p>上書きするファイルが存在する場合は、内容が上書きされ、ファイル名が呼び出し側コンポーネントに返されて、正常に対応したことが示されます。</p> <p>上書きするファイルは、指定されたディレクトリーから、ステージング・ディレクトリー (存在する場合) にコピーされ、ステージング・ディレクトリー内のファイルの内容が上書きされます。その後で、このファイルは指定されたディレクトリーに戻されます。ステージング・ディレクトリーが指定されていない場合は、指定されたディレクトリー内のファイルの内容が上書きされます。</p> <p>CreateIfFileNotExist プロパティが true に設定されている場合、アダプターは新規ファイルを作成します。</p> <p>アダプターには、固有のファイル名を生成する機能があります。10 ページの『固有ファイル名の生成』を参照してください。</p> <p>上書きするファイルが存在しない場合は、呼び出し側コンポーネントに RecordNotFound 例外が送信されます。</p>
Exists	<p>要求内のファイル名が、指定のディレクトリー、またはそのサブフォルダーに存在する場合、アダプターは true を返すと共に、ファイルの絶対パスを呼び出し側コンポーネントに返す。同じファイル名が複数のディレクトリーに存在する場合、アダプターは true を返すと共に、最初に検出したファイルの絶対パスを呼び出し側コンポーネントに返します。</p> <p>ファイル名が存在しない場合、またはディレクトリーが存在しない場合、アダプターは呼び出し側コンポーネントに false を返します。</p>
List	<p>要求で指定されたすべてのファイル名およびディレクトリーを、呼び出し側コンポーネントに返す。</p> <p>ディレクトリーのみが指定されている場合、ディレクトリー内のすべてのファイル名が取得され、応答として呼び出し側コンポーネントに送信されます。</p> <p>指定したディレクトリーが存在しない場合は、呼び出し側コンポーネントに RecordNotFound 例外が送信されます。</p>
ServerToServer FileTransfer	<p>1 つの FTP サーバー・ディレクトリーから別の FTP サーバー・ディレクトリーへ指定されたファイルを転送する。ファイルの転送が正常に終了すると、呼び出し側コンポーネントに null が返されます。</p> <p>2 つのサーバーに関する必須情報のすべてが要求に含まれていない場合、アダプターは呼び出し側コンポーネントに FTPFileServerToServerFileTransfer 例外を送信します。</p>

表 1. サポートされる Outbound 操作 (続き)

操作	結果
ExecuteFTPScript	<p>アダプター・マシンの FTP スクリプト・ファイルに組み込まれているコマンドを実行する。この操作では、FTP サーバーがサポートするコマンドのみが実行されます。操作が失敗すると、アダプターは FTPFileExecuteFTPScript 例外を呼び出し側コンポーネントに送信します。</p> <p>アダプターは、すでに確立されている接続を使用してコマンドを実行するため、接続関連のコマンド (open など) をスクリプト・ファイルに組み込まないでください。</p> <p>スクリプト・ファイルのロケーションは、DirectoryPath プロパティーおよび Filename プロパティーで指定されています。</p> <p>スクリプト・ファイル内のコマンドを FTP サーバー上の特定のディレクトリーで実行する必要がある場合は、スクリプト・ファイルにそのディレクトリーに移動する最初のコマンドを指定する必要があります。</p> <p>コマンド・リストが実行されると、応答ストリングが呼び出し側コンポーネントに返されます。また、アダプターは FTP スクリプト・ファイルにおけるパラメーター置換をサポートします (パラメーター %1、%2 を実際の値に置換)。値は要求の一部として送信されません。</p>

## Create 操作中のファイル・シーケンスの生成

Adapter for FTP では、Outbound の Create 操作中におけるファイル・シーケンスの生成をサポートしています。シーケンスの保管先ファイルの絶対パスを指定する FileSequenceLog プロパティーが導入されました。FileSequenceLog プロパティー値が指定された場合、アダプターはファイル・シーケンス番号を生成し、作成するファイルのファイル名に付加します。シーケンス番号は、\$FILENAME.\$SEQUENCE\_NUMBER.\$FILE\_EXT の形式でファイル名に付加されます。例えば、HostName = localhost および Filename = Customer.txt の場合、出力ファイルは Customer.1.txt、Customer.2.txt、Customer.3.txt のようになります。この形式は z/OS® および i5/OS® も含め、すべてのプラットフォームで同じです。シーケンス番号は、複数のアダプターが再始動するたびに、増え続けます。

アダプターがスタンドアロン・モードで作動している場合、FileSequenceLog プロパティー値はローカル・ファイル・システム上のファイルでなければなりません。アダプターがクラスター環境で作動している場合、FileSequenceLog プロパティー値は、すべてのクラスターからアクセス可能なマップ済みドライブ上のファイルでなければなりません。アダプターがシーケンス・ログ・ファイルの書き込み許可を保有していることを確認する必要があります。保有していない場合、IOException が送信されます。

シーケンス番号は、複数のアダプターが再始動するたびに、増え続ける必要があります。

ディレクトリー・パスおよびファイル名を指定する方法は 2 つあります。

1. HostName プロパティーおよび Filename プロパティーを設定し、ビジネス・オブジェクト要求の DirectoryPath フィールドおよび Filename フィールドは指定しない。

2. ビジネス・オブジェクト要求の **DirectoryPath** および **Filename** フィールドを設定する。

**HostName** プロパティおよび **Filename** プロパティを設定すると、ビジネス・オブジェクト要求ごとに値を設定する必要がなくなるため、時間を節約できます。しかし、ビジネス・オブジェクト要求に指定された値は、管理接続ファクトリー・プロパティに設定された値よりも優先します。

ファイル・シーケンスをリセットする必要がある場合、ファイル・シーケンス・ログ・ファイルにアクセスして、ファイル名のエントリーを削除できます。新しいシーケンスは 1 から始まります。 **FileSequenceLog** プロパティおよび

**GenerateUniqueFilename** プロパティの両方が有効な場合、**GenerateUniqueFilename** プロパティ値が優先され、ログ・シーケンスは生成されません。

## 固有ファイル名の生成

ファイル名は、次の 2 つの方法で生成できます。

1. シーケンス番号を維持するためのシーケンス・ファイルを、アダプターで保持し、使用することができます。この番号は、デフォルトのファイル名に付加されます。
2. 一部の FTP サーバーには、固有ファイル名を生成するサポートが組み込まれています。これを使用するには、RFC 1123 に指定された **STOU** コマンドが、サーバーでサポートされている必要があります。

最初の方法のシーケンス・ファイルを使用するには、シーケンス・ファイルのロケーションとターゲット・ファイル名を指定しなければなりません。生成されるファイル名は、ターゲット・ファイル名にシーケンス番号が付加されたものになります。

**注:** ここでは、ウィザードからの名前を正確に使用する必要があります。これらのプロパティは、管理接続ファクトリー、対話仕様、およびビジネス・オブジェクトの 3 つの場所にあります。ビジネス・オブジェクトのプロパティは対話仕様のプロパティよりも優先され、同様に管理接続ファクトリーよりも優先されます。単純にするために、特定のオブジェクトを個別に処理したい場合を除いて、管理接続ファクトリーのプロパティを使用してください。

デフォルト・ファイル名に拡張子がある場合には、その拡張子の前にシーケンス番号が付加されます。例えば、管理接続ファクトリーでのデフォルト・ファイル名が **Customer.txt** の場合、作成される出力ファイル名は **Customer.1.txt**、**Customer.2.txt** のようになります。シーケンス番号は、ビジネス・オブジェクト・タイプごとに個別に保持されます。

2 番目の方法の FTP サーバー・サポートを使用するには、対話仕様またはビジネス・オブジェクトの **GenerateUniqueFile** プロパティを設定します。**GenerateUniqueFilename** プロパティを **true** に設定すると、ファイル名生成メカニズムとして、アダプターのものではなく、FTP サーバーのものが使用されます。

**注:** アダプターでは、**GenerateUniqueFile** と **StagingDirectory** の両方のオプションを同時にはサポートしていません。



## Inbound 処理

Adapter for FTP では、イベントの Inbound 処理をサポートしています。アダプターは、イベントに対して指定された間隔で、FTP サーバーに関連付けられたファイル・システムをポーリングします。ファイルが作成、または更新されるたびに、アダプターはイベントとして追跡します。アダプターは、イベントを検出すると、ファイルのコピーを要求し、ファイル・データをビジネス・オブジェクトに変換して、利用するサービスに送信します。

以下の図は、WebSphere Adapter for FTP の Inbound 処理フローを示しています。

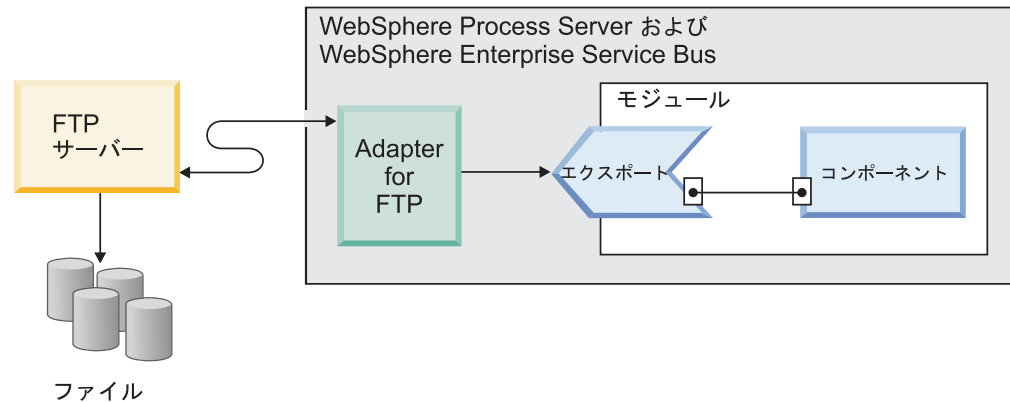


図3. Inbound 処理フロー

アダプターは、FTPPollFrequency プロパティーに基づいて定期的に、FTP サーバーのイベント・ディレクトリーのファイルをポーリングします。ファイルがイベント・ディレクトリーに到達すると、アダプターは、ファイル全体を読み取って、このファイルをアダプター・サーバー上のローカル・イベント・ディレクトリーにダウンロードします。ファイルがダウンロードされた後、アダプターは、FTPArchiveDirectory プロパティーで指定された FTP サーバーのアーカイブ・ディレクトリーにファイルをアーカイブするか、またはユーザー構成に基づいてファイルを削除します。イベント・ディレクトリー、アーカイブ・ディレクトリー、ポーリング頻度、およびポーリング数量 (1 回のポーリング周期でポーリングするファイルの数) は、すべて構成可能なプロパティーです。

ビジネス・オブジェクトが正常にエクスポートに通知された後、ローカル・ステージング・ディレクトリーのイベントは、ローカル・ファイル・システム上のアーカイブ・ディレクトリー内にアーカイブされるか、または構成に基づいて削除されます。アダプターは、イベントをアーカイブするか削除する必要があります。そうしないと、イベントは再びポーリングされます。

Inbound イベント処理は、次のステップで構成されます。

1. FTP サーバーが、ファイルの形式でイベントを生成します。
2. Adapter for FTP が、イベント・ディレクトリーをポーリングします。
3. ファイルがアダプターにダウンロードされます。
4. SplittingFunctionClassName プロパティーおよび SplitCriteria プロパティーに基づいて、ファイルが分割されます。イベント・ファイルは数個のチャンクに分割さ

れ、それぞれのチャンクはエクスポートに対して別々に通知されます。これにより、イベント処理時のメモリー・ロードを削減します。

- 区切り文字に基づいて分割を行う場合は、この機能を実行するクラスおよび分割基準が提供されます。
  - ファイル・サイズに基づいて分割を行う場合は、この機能を実行するクラス名が提供されます。
  - 他の基準に基づいて分割を行う場合は、専用のファイル分割クラスを提供する必要があります。
5. アダプターは、関数セレクターを介して、ポーリングされた文書の場所およびファイルの取得先であるマシンのホスト名を含むデータを、エクスポートに送信します。関数セレクターでは、構成済みのデータ・バインディングを呼び出して、テキスト・レコードをビジネス・オブジェクトに変換します。

## サポートされる Inbound 操作

アダプターでは、emitFTPFile 操作をサポートしています。これは、Inbound 構成時のデフォルト操作として実行されます。

## イベント・ファイル・ロック

ファイル・ロック動作は、オペレーティング・システムによって異なります。Windows<sup>®</sup> では、アダプターによってイベント・ディレクトリーからポーリングされているファイルのいずれかが、別のアプリケーションによって使用されていて、イベント・ディレクトリーへのコピー中である場合、アダプターで処理することはできません。

しかし、AIX<sup>®</sup> などの UNIX<sup>®</sup> 環境には、書き込み中のファイルにアプリケーションがアクセスできないようにするファイル・ロック機構はありません。別のアプリケーションによってイベント・ディレクトリーにコピーされているファイルをアダプターから処理できるため、エラーが発生するおそれがあります。Java<sup>™</sup> には、ファイルが書き込み中であるかどうかを確認するプラットフォーム固有の方法は存在しません。

この状態を発生させないためには、先にイベント・ファイルをステージング・ディレクトリーにコピーしてから、移動コマンドを使用してイベント・ディレクトリーに移動させます。UNIX スクリプトのサンプルが、アダプターの一部としていくつか用意されています。アダプター・インストーラーの Unix-script-file フォルダーから、CheckIfFileIsOpen.sh というスクリプト・ファイルを入手できます。

## 関数セレクター

Inbound 処理中に、関数セレクターは、サービス上で呼び出す必要がある操作を返します。外部サービス・ウィザードで Inbound 処理用のアダプターを構成する際に、関数セレクターを選択します。アダプターには、2 つの関数セレクター `FilenameFunctionSelector` と `EmbeddedNameFunctionSelector` が用意されています。

## FilenameFunctionSelector

FilenameFunctionSelector は、正規表現に基づいてオブジェクト名を解決してファイル名にマッピングする、ルール・ベースの関数セクターです。正規表現とは、特定の構文ルールに従って一連のストリングを記述したり、突き合わせたりする際に使用されるストリングです。

突き合わせルールの例について、以下の表で説明します。ここでのルールは、ObjectName および Rule の各フィールドで構成されています。

表 2. FilenameFunctionSelector の突き合わせルールの例

ファイル名	オブジェクト名	ルール
Customer0001.txt	Customer	CUST.*TXT
22310RZ93.z21	Order	[0-9]*OR[A-Z][0-9]{2}.*
22310RZ93.z21	Order	*OR.*

2 行目と 3 行目のルールは、ともに解決される名前は同じですが、2 行目のルールのほうが一致率は低くなります。これは、2 行目のルールではファイル名を突き合わせる際に特定の数字と文字の順序が必要になるのに対し、3 行目のルールではファイル名に「OR」の文字があるものをすべて解決するからです。「.\*」の文字の組み合わせは、任意の文字が任意の回数現れることを示します。

関数セクターでは、ネイティブ関数名を生成するために、ユーザーが指定したオブジェクト名の前に emit を付加します。例えば、オブジェクト名が Customer である場合、関数セクターは関数名 emitCustomer を戻します。オブジェクト名は、ラッパー名やビジネス・グラフ名ではなく、Customer や Order などのペイロード・オブジェクト名でなければなりません。パススルー・シナリオの場合、オブジェクト名として FTPFile を使用します。

複数のルールによって FilenameFunctionSelector を構成できます。各ルールには、ファイル名に対して突き合わせるオブジェクト名と正規表現が含まれています。複数のルールが一致した場合、関数セクターは、最初に一致したルールに基づいてオブジェクト名を戻します。ルールが一致しなかった場合、アダプターはエラーを生成します。構成内にルールが存在しなければ、関数セクターは関数名 emitFTPFile を使用します。

正規表現の使用法を決定するルールの詳細な説明については、

<https://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html> にある Java クラス・パターンの資料を参照してください。

## EmbeddedNameFunctionSelector

EmbeddedNameFunctionSelector は、コンテンツ固有のビジネス・オブジェクトに使用され、オブジェクト名がイベント・ファイルに組み込まれます。ここでは、要求されたコンテンツ・データ（ラッパーではなく）に基づいた関数名が戻されます。例えば、コンテンツ固有のビジネス・オブジェクトが CustomerWrapperBG の場合、関数セクターが戻す関数は emitCustomer になります。

EmbeddedNameFunctionSelector は、データ・ハンドラーを使用して構成する必要があります。データ・バインディングは、アダプター固有の WrapperDataBinding でな

ければならず、関数セクターで構成されたデータ・ハンドラーと同じものを使用するように構成する必要があります。

## Inbound データ変換

Inbound 通信時のデータ変換とは、アダプターがバイトまたはストリングなどのネイティブ・フォーマットで作成されたイベント・レコードを、ビジネス・オブジェクトに変換するために使用するプロセスを指します。アダプターは、アダプター固有のデータ・バインディングおよびデータ・ハンドラーを使用してこの変換を行います。

アダプターがイベント・レコードの内容を読み取って、ビジネス・オブジェクト内の対応する属性に値を書き込むために使用するデータ・バインディングおよびデータ・ハンドラーは、WebSphere Integration Developer の外部サービス・ウィザードを使用して構成されます。

## データ・バインディング

アダプターが、ネイティブ・フォーマットで作成されたイベント・レコードからフィールドを取得してビジネス・オブジェクトに取り込むには、データ・バインディングが必要です。データ・バインディングは、イベント・レコードのフィールドを読み取って、ビジネス・オブジェクト内の対応するフィールドに取り込む役割があります。Adapter for FTPは、Inbound 通信時に FTPFileBaseDataBinding データ・バインディングを使用します。

Inbound 通信時に、データ・バインディングはイベント・レコードから以下のフィールドを取得して、その値を以下のビジネス・オブジェクト属性に取り込みます。

- Filename
- ChunkInfo
- DirectoryPath
- FileContentEncoding
- FtpServerHostName
- FtpServerEventDirectory

変換する必要のないデータの場合、データは変更されずにシステムを通過するため、アダプターはいわゆる「パススルー処理」を実行します。

## データ・ハンドラー

データ変換では、データ・バインディングのほかにデータ・ハンドラーも使用する必要があります。データ・ハンドラーは、データをネイティブ・フォーマットからビジネス・オブジェクトに変換します。WebSphere Adapter for FTP のバージョン 6.1.0 では、アダプターは以下のデータ・ハンドラーを提供します。

- XMLDataHandler
- WTXInvokerDataHandler
- WTXMapSelectionDataHandler

## 参照によるファイルの引き渡し

また、アダプターはパススルー機能もサポートします。この機能ではイベント・ファイル名のみがエクスポートに送信されます。イベント・ファイルにはタイム・スタンプが付加され、ローカル・アーカイブ・ディレクトリーに置かれます。この機能は、データ変換が必要ないときに使用されます。

## ファイルの分割

Inbound イベント処理モードでは、オプション・ファイル分割機能をサポートしており、イベント・ファイルが数個のビジネス・オブジェクト (チャンクとも呼ばれます) に分割され、それぞれのビジネス・オブジェクトはエクスポートに対して別々に通知されます。これにより、イベント処理時のメモリー・ロードを削減します。ファイル分割は、区切り文字、または `SplitCriteria` プロパティーに指定されているファイル・サイズに基づいて実行されます。

アダプターでは、ファイル分割のために `SplitBySize` クラスおよび `SplitByDelimiter` クラスを提供しています。オプションで、カスタムのファイル分割クラスを指定し、そのクラス名を `SplittingFunctionClassName` プロパティーに入力することによって使用することができます。

## サイズによるファイルの分割

サイズ値は、`SplittingFunctionClassName` プロパティーで設定されます。

ファイル分割を実行した後の結果ファイルのことを、チャンクといいます。チャンク機能が使用可能になっている場合、ファイルの各チャンクはエクスポートに別々に送られます。`PollQuantity` プロパティーに指定されている数のビジネス・オブジェクトがエクスポートに送られます。例えば、`PollQuantity` の値が 3 の場合、以下のようになります。

ポーリングされるビジネス・オブジェクトの数は 3 です。

エクスポートが受け取るビジネス・オブジェクトの数は 3 です。

アダプターは、チャンク・データを再アセンブルすることはありません。アダプターは、チャンクを組み合わせる外部アプリケーションに、チャンク・データについての情報を提供します。チャンク情報は `chunkInfo` プロパティー (ビジネス・オブジェクトに含まれている) に設定されています。この情報には、バイト単位のチャンク・サイズやイベント ID などが含まれます。イベント ID の例は、次のとおりです。

```
AbsolutePathOfTheEventFileNameInLocalEventDirectory/_yyyy_MM_dd_HH_mm_ss_SSS.  
currentBONumber/_totalBOS
```

## 区切り文字によるファイルの分割

区切り文字は、イベント・ファイルを分割するために使用される指定値です。区切り文字は、`SplitCriteria` プロパティーで指定されます。

区切り文字の使用には、以下の規則が適用されます。

- 指定された区切り文字は、そのビジネス・オブジェクト内に含まれるデータと同じであってははいけません。同じである場合、ファイル分割が異常な結果になる可能性があります。

- 区切り文字には、イベント・ファイルの改行を表す正確な値が含まれている必要があります。イベント・ファイルを MAC マシンで作成する場合、改行文字は `¥r` になります。UNIX マシンの場合は `¥n`、Windows マシンの場合は `¥r¥n` になります。
- 複数の区切り文字がある場合、各区切り文字をセミコロン (;) で区切る必要があります。セミコロンが区切り文字の一部である場合、セミコロンを `¥;` のようにエスケープする必要があります。例えば、区切り文字を `###¥;##` とすると、`##;##` として処理されます。これは、セミコロンは区切り文字の一部であるという意味です。
- 区切り文字の一部である内容をスキップするには、区切り文字の間にある内容をスキップするよう、区切り文字の先頭に連続したセミコロン (;;) を指定します。アダプターは `###$$` を区切り文字と見なし、「this is the content that will be skipped by the adapter」をスキップします。例えば、イベント・ファイルに以下の形式のビジネス・オブジェクトが含まれていて、区切り文字が `##;$$` の場合、次のようになります。

```
Name=Smith
Company=IBM
##this is the content that will be skipped by the adapter$$
```

- 区切り文字にはどのような値でも使用でき、制限はありません。次に、有効な区切り文字の例を示します。

- `####;¥n;¥n`
- `####;$$$$;¥n;####`
- `%%%;$$$$;#####`
- `¥n;¥n;$$$$`
- `####¥;####;¥n;$$$$`
- `¥n;¥n;¥n`
- `####;$$$$`
- `¥r`
- `¥r¥n`
- `$$$$;¥r¥n`

- 区切り文字がファイルの末尾にある場合、`SplitCriteria` プロパティーは `END_OF_FILE` を使用して、ファイルの物理的な末尾を見つけます。

#### 例 1:

```
John Doe,123,Washington Ave,222-123-4567
Jane Smith,234,Washington Ave,222-123-4568
```

分離文字は行末文字になります。この例の場合、Windows では `¥r¥n` に、MAC では `¥r` に、UNIX では `¥n` になります。

#### 例 2:

```
John Doe
123 Washington Ave
222-123-4567
```

####  
Jane Smith  
234 Washington Ave  
222-123-4568

分離文字は #### になります。

## イベントのリカバリー

アダプターは、突然の終了時に Inbound 処理のイベント・リカバリーをサポートします。イベントの処理中に、アダプターはデータ・ソース上のイベント・パーシスタンス・テーブルに、そのイベントの状態を残します。イベント・パーシスタンス・テーブルを作成するには、このデータ・ソースをセットアップする必要があります。

WebSphere Process Server のリカバリー機能を使用するには、アクティベーション・スペックの AssuredOnceDelivery プロパティを true に設定する必要があります。false に設定されている場合、失敗したイベントをリカバリーすることはできません。AssuredOnceDelivery が false に設定されている場合、イベントが重複して送信される可能性があります。パフォーマンスを改善するために、イベントのリカバリー、イベントの重複、および AssuredOnceDelivery プロパティを false に設定することができます。

## イベント・パーシスタンス・テーブル

イベント・パーシスタンス・テーブルは、アダプターがイベント・レコードを処理できるまでイベントが保存されるパーシスタント・キャッシュです。アダプターは、Inbound 要求がシステム内を進行するときに、イベント・パーシスタンス・テーブルを使用して Inbound 要求を追跡します。ファイルの作成、更新、または削除が行われるたびに、アダプターはそのアクティビティをイベントとして追跡し、イベント・パーシスタンス・テーブル内のイベントの状況を更新します。各イベントの状況は、イベントが構成済みエクスポートに渡されるまで、リカバリーの目的のために、アダプターによって継続的に更新されます。

イベント・パーシスタンス・テーブルが存在しないことをアダプターが検出すると、モジュールがランタイム環境にデプロイされた際に、アダプターはイベント・パーシスタンス・テーブルを自動的に作成します。アダプターによって作成された各イベント・パーシスタンス・テーブルは、特定の Inbound モジュールに関連付けられます。アダプターは、同じイベント・パーシスタンス・テーブルを指す複数のアダプター・モジュールをサポートしていません。

アダプターは、FTP サーバーをポーリングすると、アクティベーション・スペック・プロパティに指定された検索基準に合致するイベントごとに、イベント・パーシスタンス・テーブルにエントリーを作成します。アダプターは、新しいエントリーの状況を NEW として記録します。アダプターは、FTP サーバーから、ローカル・システム上の進行中フォルダーにイベントをコピーすると、そのエントリーに IN PROGRESS というマークを付けます。アダプターは、データ変換のためにイベントを関数セレクターに送信すると、イベント・テーブルからそのエントリーを削除します。

注: 保証されたイベントの送達が必要ない場合、アダプターはイベント・パーシスタンス・テーブルが存在しなくても、イベントに対してポーリングすることができます。

以下の表に、イベント・パーシスタンス・テーブルのそれぞれの値を示します。

表 3. イベント・パーシスタンス・テーブル構造

列名	タイプ	説明
EVNTID	Varchar(255)	トラッキング用の固有のイベント ID。アダプターは、この ID を使用して、Inbound 処理時にイベントを追跡します。
EVNTSTAT	整数	<p>イベントの状況。アダプターは、状況を使用して、イベントが新しいのか処理中であるのかを判別します。</p> <p>イベント状況の値:</p> <p><b>NEWEVENT (0)</b> イベントの処理の準備ができています。</p> <p><b>PROCESSED (1)</b> アダプターがイベントを正常に処理し送信しました。</p> <p><b>FAILED (-1)</b> 1 つ以上の問題により、アダプターがこのイベントを処理できませんでした。</p>
XID	Varchar(255)	イベント配信とリカバリーが確実に行われるようにするために、アダプターによって使用されます。
EVNTDATA	Varchar(255)	アダプターの始動またはリカバリー時に障害のあるイベントが再び処理されないよう、そのイベントに ARCHIVED というマークを付けるために、アダプターによって使用されます。

## イベント・アーカイブ

アーカイブされたイベントはアーカイブ・ディレクトリーに保管され、FTPRenameExt プロパティーで指定されたファイル拡張子が付けられます。イベント・アーカイブは、オプション機能の 1 つで、処理されたすべてのイベントのレコードを提供します。この情報を使用して、イベントが正常に処理されたかどうかを確認できます。

イベント・アーカイブは、構成ごとに異なる方法で使用されます。

- FTPArchiveDirectory と FTPRenameExt の両方の値を指定し、FTPRenameExt を processed に設定する場合、アーカイブ・ファイルは、指定されたアーカイブ・ディレクトリーに次の構文で置かれます。 *filename\_timestamp.processed*
- FTPArchiveDirectory 値のみを指定する場合、アーカイブ・ファイルは、指定されたアーカイブ・ディレクトリーに以下の構文で置かれます。 *filename\_timestamp*
- FTPArchiveDirectory と FTPRenameExt の値がどちらも指定されていない場合、イベント・ファイルは、ローカル・イベント・ディレクトリーに正常にダウンロードされた後で、FTP サーバーのイベント・ディレクトリーから削除されます。



- FTPRenameExt 値のみを指定して processed に設定した場合、アーカイブ・ファイルは以下の構文で記述され、FTP サーバーのイベント・ディレクトリーに置かれます。 `filename_timestamp.processed`

## MVS™ プラットフォームでのアーカイブ

多重仮想記憶 (MVS) オペレーティング・システムでは、データ・セット名またはレコード・セット名での特殊文字 (下線など) の使用がサポートされていません。Windows および UNIX プラットフォームでは、ファイルのアーカイブ時に元のファイル名のタイム・スタンプを使用してください。これにより、アーカイブ・フォルダー内でファイル名が重複することを避けることができ、既存のファイルが上書きされることを回避できます。MVS システムでは、以下の形式を使用してください。

イベント・ファイル: Test Archived

ファイル: Test.TSyyyyMM.TSDDHHMM.TSSsSss

各項目は、以下のとおりです。

yyyy -- 年

MM -- 月

DD -- 日

HH -- 時

MM -- 分

Ss -- 秒

Sss -- ミリ秒

MVS プラットフォームの場合、データ・セットまたはレコード・セットの分離文字は . (小数点) になります。データ・セットまたはレコード・セットで使用できる . (小数点) の最大数は 6 です。データ・セット名またはレコード・セット名では、. (小数点) の区切りごとに使用できる文字数は 8 文字以内で、合計文字数は 44 文字以内でなければなりません。以下に、この形式のファイル名の例を示します。

FTPRenameExt: ARCHIVE

アーカイブ・ファイル: TEST.TS200304.TS290535.TS42234.ARCHIVE

## ビジネス・オブジェクト

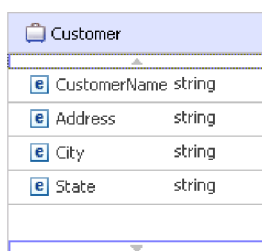
ビジネス・オブジェクトとは、データ、データ上で実行されるアクション、およびデータを処理するための追加の指示 (存在する場合) で構成される構造体のことです。データは、送り状または従業員レコードなどのビジネス・エンティティ、または非構造化テキストのいずれかを表すことができます。

## アダプターによるビジネス・オブジェクトの使用法

アダプターは、ビジネス・オブジェクトを使用して、FTP サーバー にデータを送信、またはデータを取得します。Inbound 操作時でのアダプターの主なジョブは、ネイティブ・フォーマットで作成されたイベント・レコードから情報を取り出し、その情報をビジネス・オブジェクトに変換して、サービスに転送することです。Outbound 操作の場合は、このプロセスが逆方向に行われます。アダプターはサービスからビジネス・オブジェクトを受信し、ビジネス・オブジェクト内で検出した詳細情報からイベント・レコードを作成し、そのイベント・レコードを FTP サーバーに送信します。

## ビジネス・オブジェクト内でのデータの表現方法

ビジネス・オブジェクトは、WebSphere Integration Developer 内のビジネス・オブジェクト・エディターを使用して作成しますが、このエディターでは、ビジネス・オブジェクトのグラフィカル・ビューを表示できます。次の図に示すように、ビジネス・オブジェクトはフィールドとその値の組で構成されます。これはカスタマー・ビジネス・オブジェクトです。このオブジェクトは、名前、住所、電話番号の各情報をカスタマー・レコードに記録することが分かります。この例ではストリング値を使用していますが、ビジネス・オブジェクト・エディターでは、この他にもさまざまな値をサポートしています。



Customer	
CustomerName	string
Address	string
City	string
State	string

図 4. ビジネス・オブジェクト内でのデータの表現方法

## ビジネス・オブジェクトが作成される仕組み

ビジネス・オブジェクトは、外部サービス・ウィザードまたはビジネス・オブジェクト・エディターを使用して作成できます。これらは両方とも、WebSphere Integration Developer から起動できます。

外部サービス・ウィザードを開始する前に、ビジネス・オブジェクト・エディターを使用して XSD ファイルを定義してあれば、これらのスキーマを基にアダプターがビジネス・オブジェクトを作成します。ビジネス・オブジェクト・エディターを使用してビジネス・オブジェクトを作成する方法についての説明は、以下のリンクを参照してください: <http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/in>。ビジネス・オブジェクトを作成した後、ビジネス・オブジェクト・エディターを使用して、ビジネス・オブジェクトの階層を定義します。

## ビジネス・グラフ

アダプターの構成時に、ビジネス・グラフを生成するオプションを選択することもできます。バージョン 6.0.2では、トップレベルのビジネス・オブジェクトがビジネ

ス・グラフにそれぞれ組み込まれています。ここに含まれている動詞を、アプリケーションがバージョン 6.0.2 で使用して、操作を実行するための追加情報を指定することができます。バージョン 6.1.0 では、ビジネス・グラフはオプションになっています。バージョン 6.1.0 より前のバージョンの WebSphere Integration Developer で作成したモジュールにビジネス・オブジェクトを追加する場合にのみ、ビジネス・グラフが必要になります。ビジネス・グラフが存在する場合、ビジネス・オブジェクトが処理されますが、動詞は無視されます。

## 外部サービス・ウィザード

WebSphere Adapter for FTP の外部サービス・ウィザードを使用して、選択したオブジェクトからサービスを作成し、ビジネス・オブジェクトを生成します。また、ウィザードは、アダプターが Service Component Architecture (SCA) コンポーネントとして稼働できるようにするサービス成果物も生成します。

---

## 標準規格の準拠

この製品は、アクセシビリティ標準やインターネット・プロトコル標準といった、いくつかの行政標準および業界標準に準拠しています。

## アクセシビリティ

IBM は、年齢や能力を問わず、すべての人が便利に使用できる製品の提供に努めています。WebSphere Adapters は、完全にアクセス可能で、米国リハビリテーション法第 508 条に準拠しています。アクセシビリティ機能を使用すると、運動障害または視覚障害など身体に障害を持つユーザーがソフトウェア・プロダクトを快適に操作できるようになります。これらの機能は、WebSphere Adapters のインストール機能や管理機能に組み込まれています。

### 管理

ランタイム管理コンソールは、エンタープライズ・アプリケーションのデプロイメントおよび管理用の基本インターフェースです。このコンソールは、標準の Web ブラウザー内に表示されます。Microsoft® Internet Explorer や Netscape Browser などのアクセス可能な Web ブラウザーを使用すると、次のことが可能になります。

- スクリーン・リーダー・ソフトウェアとデジタル・スピーチ・シンセサイザーを使用して、画面上に表示されている内容を聞く
- IBM ViaVoice® などの音声認識ソフトウェアを使用したデータの入力とユーザー・インターフェースへのナビゲート
- マウスの代わりにキーボードを使用して機能を操作する

標準的なテキスト・エディター、スクリプト・インターフェース、コマンド行インターフェースなどを利用して製品機能を構成し、付属のグラフィカル・インターフェースの代わりに使用することができます。

場合によっては、特定の製品機能についての文書に、その機能のアクセシビリティについての追加情報が記載されています。

## 外部サービス・ウィザード

外部サービス・ウィザードは、モジュールを作成するのに使用する主コンポーネントです。このウィザードは、非常に使い勝手が良く、WebSphere Integration Developer から利用できる Eclipse プラグインの 1 つとして実装されています。

## キーボード・ナビゲーション

この製品では、標準の Microsoft Windows ナビゲーション・キーを使用します。

## IBM とアクセシビリティ

IBM のアクセシビリティに対する取り組みについては、*IBM Accessibility Center* の Web サイト (<http://www.ibm.com/able/>) を参照してください。

## インターネット・プロトコル・バージョン 6 (IPv6)

WebSphere Process Server および WebSphere Enterprise Service Bus は、インターネット・プロトコル・バージョン 6 (IPv6) の互換性について、WebSphere Application Server に依存しています。

IBM WebSphere Application Server バージョン 6.1.0 以降では、ピュアなインターネット・プロトコル・バージョン 6.0 (IPv6) がサポートされています。

WebSphere Application Serverでのこの互換性について詳しくは、<http://www.ibm.com/software/webservers/appserv/was/library/>の IPv6 サポートを参照してください。

IPv6 について詳しくは、<http://www.ipv6.org> を参照してください。

---

## 第 2 章 アダプター実装の計画

IBM WebSphere Adapter for FTP を実装するには、Inbound および Outbound 処理を計画し、セキュリティーおよびパフォーマンス要件を考慮する必要があります。

---

### 始める前に

アダプターのセットアップおよび使用を開始する前に、ビジネス・インテグレーションの概念、使用する統合開発ツールおよびランタイム環境の機能と要件について徹底十分に理解しておく必要があります。

WebSphere Adapter for FTP を構成して使用するには、以下の概念、ツール、および作業に関する知識と経験が必要です。

- 構築するソリューションの業務要件。
- Service Component Architecture (SCA) プログラミング・モデルなどのビジネス・インテグレーションの概念およびモデル。
- ソリューションの構築に使用する統合開発ツールによって提供される機能。これらのツールの使用によるモジュールの作成方法、コンポーネントのテスト方法、その他の統合作業の実行方法を理解しておく必要があります。
- 統合ソリューションに使用するランタイム環境の機能および要件。ホスト・サーバーの構成と管理の方法、および管理コンソールの使用によるプロパティ定義の設定と変更の方法、接続の構成方法、イベントの管理方法を理解しておく必要があります。
- ファイル転送プロトコル (FTP)。インターネットを介してファイルを交換するためのプロトコル。
- ご使用のソリューションの特定のファイル・システム上でファイルにアクセスするために使用する FTP サーバー。

---

### セキュリティー

Secure Sockets Layer (SSL) を構成すると、FTP サーバーとアダプターの間で渡される情報の健全性を保護できます。通信の保護のためには、SSL プロトコルをサポートし、秘密鍵と証明書を持つセキュア FTP サーバーをインストールし、構成する必要があります。また、ユーザーが必要とする場合は、連邦情報処理標準 (FIPS) 140 に従って稼働するようにアダプターを構成することもできます。

#### Secure Sockets Layer の構成

ネットワークを介して送信されるデータは、第三者に傍受される可能性があります。このデータがパスワードやクレジット・カード番号などの個人情報を含んでいる場合は、このデータを権限のないユーザーが判読できないようにするための処置を取る必要があります。Secure Sockets Layer (SSL) を構成することによって、FTP サーバーとアダプター間で受け渡される情報の健全性を保護します。

始める前に

SSL を使用可能にするには、以下の前提条件を満たしている必要があります。

- SSL を使用するセキュア通信を FTP サーバーがサポートしている。
- FTP サーバーが固有の秘密鍵と証明書を持っている。
- FTP クライアントがインストールされている。
- セキュア FTP サーバーとのデータ転送に、アダプターがパッシブ FTP モードを使用している。クライアントとサーバーの間にファイアウォールが存在する場合は、ファイアウォールの設定を、このモードを使用可能にする構成にする必要があります。

### このタスクを実行する理由および時期

SSL がアダプターで使用できるように構成されていない場合、FTP サーバーを介したファイルの受け渡しは、第三者の妨害に対して弱い。SSL を使用すると、意図的であるか意図的でないかにかかわらず、トランスポート中のデータの変更が禁止され、データが傍受されないように保護されます。SSL が実効的であるのは、複数の暗号処理を使用するためです (FTP サーバーでの認証には公開鍵暗号方式を使用し、プライバシーとデータ保全性を確保するために秘密鍵暗号方式とデジタル署名を使用します)。SSL を使用すると、アダプターが FTP サーバーの ID を認証することができます。

### このタスクの手順

1. FTP クライアントのトラストストアを設定します。トラストストアによって、FTP クライアントが信頼できるものを判断することができます。SSL を使用すると、WebSphere Process Server は FTP クライアントに検証用の証明書を送信します。FTP クライアントは証明書を検証して、目的とするメール・サーバーと通信していることを確認します。この検証プロセスを使用可能にするためには、FTP サーバーの証明書がクライアントのトラストストア内に存在していなければなりません。
  - a. WebSphere Integration Developer で、サーバー・インスタンスを右クリックし、「管理コンソールの実行 (Run administrative console)」をクリックします。
  - b. 「セキュリティ」を展開します。
  - c. 「SSL 証明書および鍵管理 (SSL certificate and key management)」を選択します。
  - d. 「関連項目」の下で、「鍵ストアおよび証明書 (Key stores and certificates)」を選択します。
  - e. **NodeDefaultTrustStore** を選択します。次の 25 ページの図 5 を参照してください。

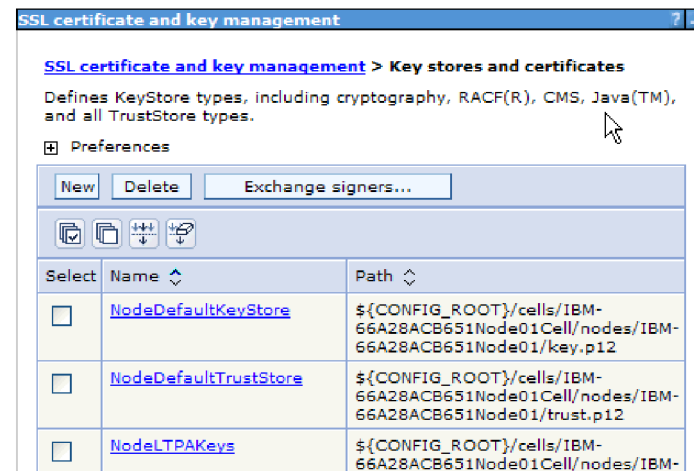


図 5. NodeDefaultTrustStore の選択

- f. 「追加プロパティ (Additional properties)」の下で、「署名者証明書 (Signer certificates)」を選択します。
- g. 「追加」をクリックします。
- h. 「別名」フィールドに、証明書の名前を入力します。次の図 6 を参照してください。

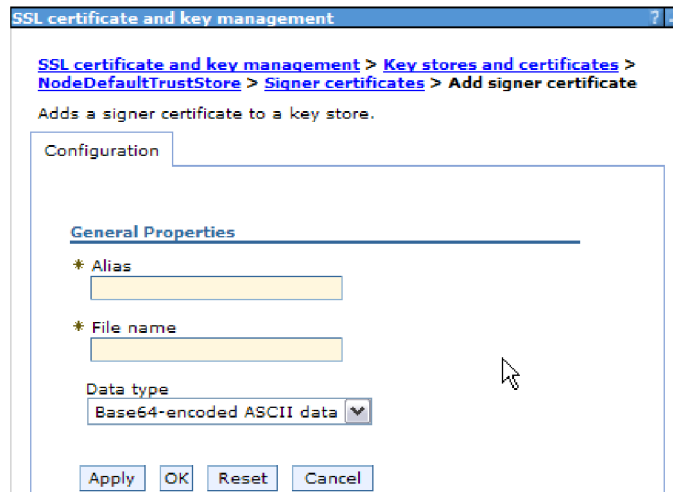


図 6. FTP サーバーの証明書用の署名者証明書プロパティを追加する

- i. 「ファイル名」フィールドに、FTP サーバーの証明書の絶対パスを入力します。
  - j. 「OK」をクリックします。
2. アダプターの SSL プロパティを構成します。
    - a. 外部サービス・ウィザードで、「enableSSL」を True に設定します。デフォルトでは、「enableSSL」は False に設定されています。

## 連邦情報処理標準 140 に対応するアダプターの構成

連邦情報処理標準 140 (FIPS) は、米国連邦政府が定めた標準で、ソフトウェア製品およびモジュール内で使用される暗号機能 (暗号化、暗号解除、ハッシング (メッセージ・ダイジェスト)、Secure Sockets Layer、Transport Layer Security、Internet Protocol Security、セキュア・シェル、シグニチャー、鍵交換、鍵生成、証明書生成など) に関するものです。FIPS 標準に準拠する必要がある米国連邦政府機関で働くユーザーの場合は、アダプターを FIPS 方式で稼働するように構成することができます。

### このタスクを実行する理由および時期

FIPS 方式で稼働するようにアダプターを構成すると、アダプターが FIPS 承認の方式およびプロバイダーに適合する暗号機能を持つモジュールで動作するように制限されます。アダプター側から見ると、FIPS 方式で稼働することによって、アダプターは Transport Layer Security (TLS) セキュア・ソケット・プロトコルを使用するように制限されます。

**注:** アダプターを FIPS 形式で稼働させる場合、FTP サーバーが SSL v3.1 (TLS v1.0 と同様) をサポートしていなければなりません。これは、FTP サーバーのウィザードを通じて使用可能にする必要があります。SSL v3.1 が適切にサポートされていないと、アダプターによる SSL ハンドシェイクが失敗する場合があります。

アダプターを FIPS 方式で稼働させるには、IBM Java Secure Socket Extension (IBMJSSE2) プロバイダー・パッケージを使用するように、アダプターに指示する必要があります。IBMJSSE2 プロバイダーは、IBM SDK バージョン 5.0 の Java セキュリティー・ファイル内に事前登録された、Java Secure Socket Extension プロバイダーです。IBMJSSE2 は、FIPS 承認パッケージを使用します。

アダプターを FIPS 方式で稼働するには、以下の手順を実行します。

### このタスクの手順

1. IBMJSSE2 プロバイダーで、`com.ibm.jsse2JSSEFIPS` プロパティを `True` に設定します。
2. IBMJSSE2 プロバイダーがすべての JSSE 要求を処理するように、以下のセキュリティー・プロパティを設定します。
  - a. `ssl.SocketFactory.provider` プロパティを `com.ibm.jsse2SSLSocketFactoryImpl` に設定します。
  - b. `ssl.ServerSocketFactory.provider` プロパティを `com.ibm.jsse2SSLServerSocketFactoryImpl` に設定します。
3. セキュリティー・プロパティ・ファイルで、プロバイダー・リストの IBMJCE プロバイダーの上に、`IBMJCEFIPS` プロバイダー `com.ibm.crypto.fips.provider.IBMJCEFIPS` を追加します。  
`security.provider.n=providername` (`n` はプロバイダーの順序を示します) の形式に従ってください。値 1 を持つプロバイダーは、値 2 を持つプロバイダーより前とみなされます。IBMJCE プロバイダーを除去しないでください。
4. WebSphere Process Server 管理コンソールの Java 仮想マシン (JVM) プロパティでシステム・プロパティを設定します。 `-Dpropertyname=propertyvalue` の形式に従ってください。



5. Java.security ファイル (*WebSphere Process Server Java virtual machine/lib/security* ディレクトリー内にあります) のセキュリティー・プロパティーを設定します。

セキュリティー詳細の構成の詳細については、WebSphere Process Server または WebSphere Enterprise Service Bus のセキュリティーに関する資料を参照してください。

---

## ユーザー認証

アダプターでは、FTP サーバーへの接続に必要なユーザー名およびパスワードを指定する方法がいくつかサポートされています。それぞれの方法の特徴および制限を理解した上で、ご使用のアプリケーションにとって適切なセキュリティー・レベルであり、かつ都合のよい方法を選択してください。

アダプターをアプリケーションに統合するには、アダプターが WebSphere Process Server または WebSphere Enterprise Service Bus での実行時に FTP サーバーに接続して Outbound 要求および Inbound イベントを処理するために使用するユーザー名およびパスワードを指定する必要があります。

実行時、アダプターは、FTP サーバーに接続するためにユーザー名およびパスワードを提供する必要があります。ユーザー介入なしに接続するためには、アダプターは保管されているユーザー情報のコピーにアクセスしなければなりません。サーバー環境では、ユーザー情報の保管方法はいくつかあります。外部サービス・ウィザードでは、アダプターが次のいずれかの方法でユーザー情報を取得するように構成できます。

- アダプター・プロパティー
- データ・ソース
- J2C 認証別名

アダプター・プロパティーへのユーザー名およびパスワードの保管は、実行時にこの情報を提供するための直接的な方法です。外部サービス・ウィザードを使用してモジュールを構成するときに、このユーザー名およびパスワードを指定します。ユーザー名とパスワードを直接指定する方法はもっとも簡単なように見えますが、この方法には重要な制限があります。アダプター・プロパティーは暗号化されません。パスワードは、サーバー上で他のユーザーがアクセスできるフィールドに平文で保管されます。さらに、パスワードが変更された場合は、その FTP サーバーにアクセスするすべてのアダプター・インスタンスで、パスワードを更新しなければなりません。これは、アプリケーション EAR ファイルに組み込まれているアダプターだけでなく、サーバーに個別にインストールされたアダプターも該当します。

データ・ソースを使用する方法では、他のアプリケーション用にすでに確立された接続を使用します。例えば、複数のアプリケーションが同じユーザー名およびパスワードを使用して同じデータベースにアクセスする場合は、同じデータ・ソースを使用してそれらのアプリケーションをデプロイできます。ユーザー名およびパスワードを知るユーザーを、そのデータ・ソースにアプリケーションをデプロイする最初のユーザー、またはデータ・ソースを個別に定義する最初のユーザーのみに限定できます。

Java Authentication and Authorization Service (JAAS) で作成された J2C 認証別名を使用する方法は、堅固でセキュアなアプリケーション・デプロイ方法です。管理者は、システムにアクセスする必要がある 1 つ以上のアプリケーションで使用される認証別名を作成します。ユーザー名およびパスワードを知るユーザーを、その管理者のみに限定できます。管理者は、変更が必要な場合は単一の場所でパスワードを変更できます。

---

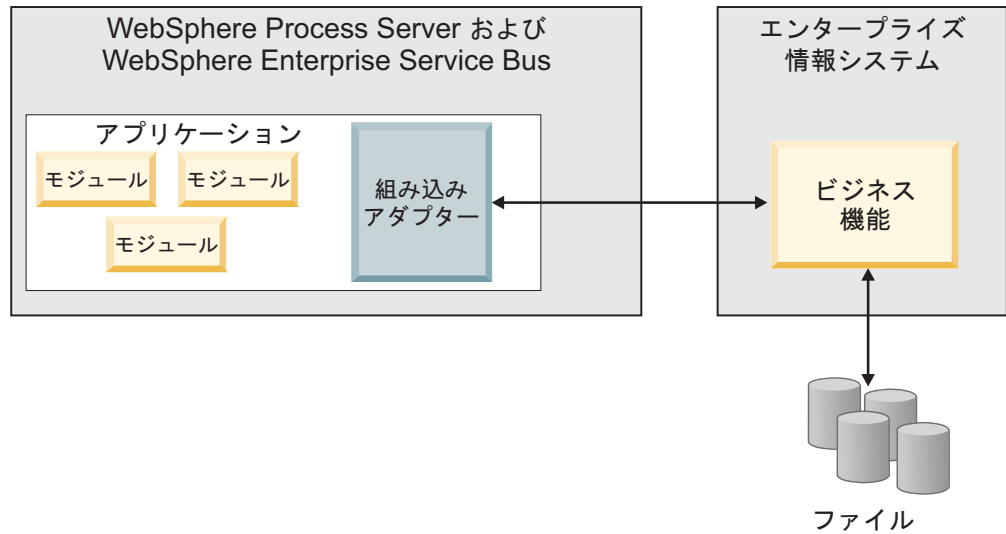
## デプロイメント・オプション

デプロイされるアプリケーションの一部としてアダプターを組み込むか、スタンドアロンとして RAR ファイルをデプロイするか、いずれかを選択することができます。

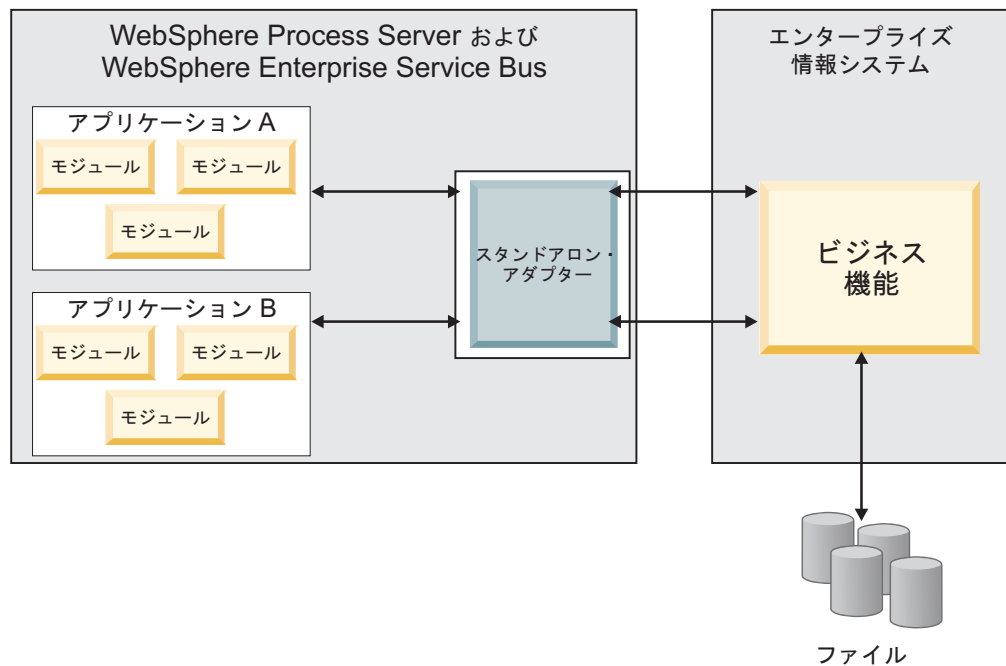
デプロイメント・オプションについての説明は、以下のとおりです。

- 「**単一アプリケーションが使用するモジュールで (With module for use by single application)**」。アダプター・ファイルをモジュール内に組み込むと、モジュールをすべてのアプリケーション・サーバーにデプロイすることができます。単一のモジュールによってアダプターを使用する場合、または複数のモジュールを使ってさまざまなバージョンのアダプターを実行する必要がある場合には、組み込みアダプターを使用します。組み込みアダプターを使用すると、単一のモジュールのアダプターのみをアップグレードすることができるため、その他のモジュールのアダプター・バージョンが変更されて動作が不安定になることはありません。
- 「**複数アプリケーションが使用するサーバー上 (On server for use by multiple applications)**」。モジュール内にアダプター・ファイルを組み込まない場合は、このモジュールを実行させるアプリケーション・サーバーごとに、アダプター・ファイルをスタンドアロン・アダプターとしてインストールする必要があります。複数のモジュールで同じバージョンのアダプターを使用できる場合に、中央の場所でアダプターを管理するには、スタンドアロン・アダプターを使用します。スタンドアロン・アダプターでは、複数のモジュールに対して単一のアダプター・インスタンスを実行することによって、必須リソースを削減することもできます。

エンタープライズ・アーカイブ (EAR) ファイル内には、組み込みアダプターがバンドルされています。この組み込みアダプターは、一緒にパッケージされ、デプロイされたアプリケーションでのみ使用することができます。



スタンドアロン・アダプターを表すのは、スタンドアロンのリソース・アダプター・アーカイブ (RAR) ファイルです。これは、デプロイされた後、サーバー・インスタンス内のすべてのデプロイ済みアプリケーションから使用することができます。



WebSphere Integration Developer を使用してアプリケーション用のプロジェクトを作成する際に、アダプターのパッケージ方法 [(EAR) ファイルにバンドルするか、スタンドアロンの (RAR) ファイルにするか] を選択することができます。この選択に応じて、アダプターをランタイム環境で使用する方法、および管理コンソールでアダプターのプロパティを表示する方法が異なります。

アダプターをアプリケーションに組み込むか、スタンドアロン・モジュールとしてデプロイするかのどちらを選択するかによって、アダプターの管理の仕方が決まります。アダプターのコピーを 1 つだけ保持して、アダプターのアップグレード時に

複数のアプリケーションを中断させなくても済むようにしたい場合は、アダプターをスタンドアロン・モジュールとしてデプロイすることをお勧めします。

複数のバージョンを稼働させる計画があるため、アダプターのアップグレード時に起こる可能性のある中断により配慮する場合は、アダプターをアプリケーションに組み込むことになります。アダプターをアプリケーションに組み込む場合、アダプターのバージョンをアプリケーションのバージョンに関連付けて、単一のモジュールとして管理することができます。

## アダプターのアプリケーションへの組み込みに関する考慮事項

アダプターをアプリケーションに組み込む計画がある場合は、以下の点を考慮してください。

- 組み込みアダプターには、クラス・ローダーの独立性があります。

クラス・ローダーは、アプリケーションのパッケージ化、およびランタイム環境にデプロイされたパッケージ済みアプリケーションの動作に影響を与えます。クラス・ローダーの独立性とは、アダプターが他のアプリケーションまたはモジュールからクラスをロードできないことを指します。クラス・ローダーの独立性によって、似た名前を持つ 2 つのクラスがそれぞれ別のアプリケーションに存在する場合に、互いに干渉することがなくなります。

- アダプターが組み込まれた各アプリケーションを、別々に管理する必要があります。

## スタンドアロン・アダプターを使用する際の考慮事項

スタンドアロン・アダプターを使用する場合は、以下の点を考慮してください。

- スタンドアロン・アダプターには、クラス・ローダーの独立性がありません。

スタンドアロンのアダプターにはクラス・ローダーの独立性がないため、指定した Java 成果物については、あるバージョンのもののみが実行されます。実行される成果物のバージョンとその順序については、事前には分かりません。例えば、スタンドアロンのアダプターを使用する場合は、リソース・アダプターのバージョンを 1 つ だけか、Adapter Foundation Class (AFC) のバージョンを 1 つ だけか、あるいはサード・パーティーの JAR バージョンを 1 つ だけか、いずれかになります。スタンドアロン・アダプターとしてデプロイされたアダプターはすべて、単一の AFC バージョンを共有し、1 つのアダプターのすべてのインスタンスは同じコードのバージョンを共有します。1 つのサード・パーティー・ライブラリーを使用するアダプター・インスタンスはすべて、そのライブラリーを共有しなければなりません。

- これらの共有成果物のいずれかを更新する場合、その成果物を使用するすべてのアプリケーションが影響を受けることになります。

例えば、サーバー・バージョン X で動作しているアダプターを使用しているときに、クライアント・アプリケーションのバージョンをバージョン Y に更新すると、元のアプリケーションの処理が停止することがあります。

- AFC には前のバージョンとの互換性がありますが、スタンドアロンとしてデプロイする RAR ファイルには、すべて最新の AFC バージョンが必要になります。

スタンドアロン・アダプターのクラスパス内に、JAR ファイルのコピーが複数ある場合、使用される JAR ファイルはランダムになります。このため、すべてを最新バージョンにしておく必要があります。

---

## クラスター環境での WebSphere Adapters

モジュールをクラスター化されたサーバー環境にデプロイすることで、アダプターのパフォーマンスおよび可用性を向上させることができます。スタンドアロン・アダプター、または組み込みアダプターのどちらを使用してモジュールをデプロイする場合も、モジュールは、クラスター内のすべてのサーバー内に複製されます。

WebSphere Process Server、WebSphere Application Server Network Deployment、および WebSphere Extended Deployment では、クラスター環境がサポートされています。クラスターとは、ワークロードの平衡を取り、高可用性とスケーラビリティを提供するために、一緒に管理されるサーバー・グループのことです。サーバー・クラスターをセットアップするときには、デプロイメント・マネージャー・プロファイルを作成してください。デプロイメント・マネージャーのサブコンポーネントである HAManager により、アダプター・インスタンスを活動状態にするよう JCA (Java EE Connector Architecture) コンテナに通知されます。JCA コンテナにより、アダプター・インスタンスのランタイム環境が提供されます。クラスター環境の作成について詳しくは、リンク [http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun\\_wlm\\_cluster\\_v61.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm_cluster_v61.html) を参照してください。

必要に応じて WebSphere Extended Deployment を使用することで、クラスター環境内のアダプター・インスタンスのパフォーマンスを向上させることができます。

WebSphere Extended Deployment は、WebSphere Application Server Network Deployment で使用されている静的作業負荷マネージャーの代わりに、動的作業負荷マネージャーを使用することにより、WebSphere Application Server Network Deployment の機能を拡張します。動的作業負荷マネージャーは、要求による負荷の平衡化を動的に行うことによって、クラスター内のアダプター・インスタンスのパフォーマンスを最適化できます。これは、負荷の変動に応じて、アプリケーション・サーバー・インスタンスを自動的に停止したり始動したりできることを意味します。これにより、能力や構成が異なる複数のマシンが負荷の変動に一樣に対処できるようになります。WebSphere Extended Deployment の利点について詳しくは、以下のリンクを参照してください。 <http://publib.boulder.ibm.com/infocenter/wxdinfo/v6r1/index.jsp>

クラスター化された環境では、アダプター・インスタンスにて、Inbound 処理および Outbound 処理の両方を処理することができます。

**制約事項:** Inbound 通信時に、WebSphere Process Server のクラスター・バックアップ・ノードとクラスターの 1 次ノードがそれぞれ異なるオペレーティング・システムにインストールされていると、WebSphere Adapter for FTP はこれらのノードの間でプーリングを切り替えることができません。例えば、Windows の 1 次ノードでアダプターがプーリングを開始した場合は、UNIX のバックアップ・ノードに切り替えることはできません。進行中のイベントが保管されているディレクトリーで使用されている Windows のパスを処理することができないためです。

## Inbound 処理の高可用性

Inbound 処理は、FTP サーバーのデータを更新した結果、起動するイベントに基づいています。WebSphere Adapter for FTP は、イベント・テーブルをポーリングすることで更新を検出するよう構成されます。その後、アダプターはイベントをそのエンドポイントにパブリッシュします。

**重要:** クラスター環境では、イベント・ディレクトリーをファイル共有システムに置いて、どのクラスター・マシンに対してもローカル側にならないようにすることをお勧めします。

モジュールをクラスターにデプロイすると、JCA (Java EE Connector Architecture) コンテナにより、enableHASupport リソース・アダプター・プロパティーが検査されます。enableHASupport プロパティーの値が真である場合 (デフォルトの設定)、すべてのアダプター・インスタンスはポリシー N のうちの 1 つを持つ HAManager に登録されます。このポリシーは、アダプター・インスタンスのうちの 1 つのみがイベントのポーリングを開始することを意味します。クラスター内のその他のアダプター・インスタンスが開始していても、それらのインスタンスは、アクティブなアダプター・インスタンスがイベントの処理を完了するまで、アクティブ・イベントに関して休止のままとなります。ポーリング・スレッドが開始しているサーバーが何らかの理由でシャットダウンした場合は、バックアップ・サーバーのいずれかで稼働しているアダプター・インスタンスが活動状態になります。

**重要:** enableHASupport プロパティーの設定は変更しないでください。

## Outbound 処理の高可用性

クラスター化された環境では、Outbound 処理要求の実行に、複数のアダプター・インスタンスが使用可能です。そのため、Outbound 要求のために WebSphere Adapter for FTP と対話するアプリケーションが、ご使用の環境に複数存在する場合、モジュールをクラスター環境にデプロイすることで、パフォーマンスが向上することがあります。クラスター化された環境では、複数の Outbound 要求が同じレコードを処理しようとしないう限り、複数の Outbound 要求を同時に処理することができます。

複数の Outbound 要求によって、同一のレコード (顧客の住所など) が処理されようとした場合、その要求は WebSphere Application Server Network Deployment のワークロード管理機能によって、使用可能なアダプター・インスタンスの間で受信順に分配されます。このため、クラスター化された環境では、この種の Outbound 要求は、単一サーバー環境内と同じように処理されます。つまり、1 つのアダプター・インスタンスが一度に処理するのは、1 つの Outbound 要求のみです。ワークロード管理について詳しくは、リンク [http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun\\_wlm.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm.html) を参照してください。

---

## バージョン 6.1.0 へのマイグレーション

バージョン 6.1 の WebSphere Adapter for FTP へのマイグレーションを行うことで、以前のバージョンのアダプターが自動的にアップグレードされます。さらに、以前のバージョンのアダプターが組み込まれているアプリケーションをマイグレーションすることによって、バージョン 6.1 に搭載されているさまざまな機能や能力を、アプリケーションで利用できるようになります。

### マイグレーションに関する考慮事項

WebSphere Adapter for FTP バージョン 6.1.0 には、既存のアプリケーションに影響を与える可能性のある更新が含まれています。WebSphere Adapter for FTP を使用するアプリケーションをマイグレーションする前に、以下のセクションの情報を考慮してください。

#### 前のバージョンとの互換性

WebSphere Adapter for FTP バージョン 6.1.0 は、アダプターのバージョン 6.0.2 と完全互換で、カスタム・ビジネス・オブジェクト (XSD ファイル) およびデータ・バイndingを扱うことができます。

WebSphere Adapter for FTP のバージョン 6.1 はバージョン 6.0.2 と完全互換であるため、WebSphere Adapter for FTP のバージョン 6.0.2 を使用していたアプリケーションは、バージョン 6.1 にアップグレードしても、変更することなく実行できます。ただし、バージョン 6.1 のアダプターで搭載されたフィーチャーおよび機能をアプリケーションで使用したい場合は、マイグレーション・ウィザードを実行してください。

マイグレーション・ウィザードは、バージョン 6.0.2 のアダプターをバージョン 6.1 に置換 (アップグレード) して、アプリケーションでバージョン 6.1 のフィーチャーおよび機能を使用できるようにします。

**注:** マイグレーション・ウィザードでは、マッパーやメディエーターなどのマイグレーション・コードを、バージョン 6.1 のアダプターで処理するために新規に作成することも、既存のものを変更することはありません。いずれかのアプリケーションに 6.0.2.x 以前のバージョンのアダプターが組み込まれている場合に、バージョン 6.1.0 にアップグレードして、アプリケーションで 6.1 のフィーチャーおよび機能を利用するには、それらのアプリケーションに変更を加えなければならない場合があります。

単一モジュール内の成果物でバージョンに関する不整合がある場合、そのモジュールには全体に対する不整合のマークが付けられ、マイグレーション時に選択できなくなります。バージョンの不整合は、プロジェクトの破損を招く症状となるおそれがあるため、ワークスペース・ログに記録されます。

#### アップグレード、またはアップグレードとマイグレーションの決定

マイグレーション・ウィザードのデフォルト処理は、アダプターのアップグレードと、アプリケーション成果物のマイグレーションを実行することです。これによって、アプリケーションがバージョン 6.1 のアダプターのフィーチャーと機能を使用

できるようになります。コネクタ・プロジェクトを選択してコネクタをアップグレードするようにした場合、このウィザードではマイグレーションに関連する成果物を自動的に選択します。

アダプターをバージョン 6.0.2 からバージョン 6.1 にアップグレードするが、アダプター成果物はマイグレーションしないようにするには、マイグレーション・ウィザードの該当するページからアダプター成果物を選択解除します。

アダプター成果物を一切選択せずにマイグレーション・ウィザードを実行すると、アダプターはインストールされてアップグレードされますが、成果物はマイグレーションされないため、バージョン 6.1 のアダプターのフィーチャーと機能をアプリケーションで利用できなくなります。

## マイグレーション・ウィザードをテスト環境で最初に実行する

アダプターのマイグレーション時には、バージョン 6.1 の WebSphere Adapter for FTP を使用するアプリケーションへの変更が必要になる場合があるため、必ず最初に関開発環境でマイグレーションを実行して、アプリケーションをテストをしてから、アプリケーションを実稼働環境にデプロイする必要があります。

マイグレーション・ウィザードは、開発環境に完全に統合されています。

## 非推奨の機能

非推奨の機能とは、サポートされてはいるけれども推奨されない機能、および廃止される可能性のある機能を指します。これまでのバージョンの WebSphere Adapter for FTP の機能のうち、バージョン 6.1.0 で非推奨になったものは、次のとおりです。

- アクティベーション・スペック・プロパティ `EventContentType` および `DefaultObjectName`。
- 管理接続ファクトリー・プロパティ `FTPURL`。
- データ・バインディング `FTPFileDataBinding`。
- XSD ファイルに含まれる注釈タグ。

## マイグレーションの実行

バージョン 6.1.0 のアダプター・マイグレーション・ウィザードを使用して、プロジェクトまたは EAR ファイルをマイグレーションすることができます。このツールが終了すると、マイグレーションが完了したことになります、プロジェクトで作業またはモジュールをデプロイできるようになります。

### 始める前に

『マイグレーションに関する考慮事項』に記載されている情報を確認してください。

### このタスクを実行する理由および時期

WebSphere Integration Developer でマイグレーションを実行するには、以下のステップを完了してください。



注: マイグレーションが完了すると、旧バージョンの WebSphere Process Server、WebSphere Enterprise Service Bus、または WebSphere Integration Developer とモジュールとの互換性は失われます。

注: WebSphere Integration Developer の J2EE パースペクティブ内のコネクタ・プロジェクトのコンテキスト・メニューから、アダプター・マイグレーション・ウィザードを実行する方法について、以下のステップで説明します。

注: マイグレーションは、以下の手順のいずれかで行うことができます。

- J2EE パースペクティブ内でプロジェクトを右クリックし、「**マイグレーション**」 → 「**プロジェクトのマイグレーション (Migrate project)**」と選択します。
- 「問題」ビューで、マイグレーション固有のメッセージを右クリックし、「**クイック・フィックス (Quick fix)**」を選択して、問題箇所を訂正します。

### このタスクの手順

1. 既存のプロジェクトの PI (プロジェクト交換) ファイル、またはデプロイ済みアプリケーションの EAR (エンタープライズ・アーカイブ) ファイルをワークスペースにインポートします。
2. J2EE パースペクティブに切り替えます。
3. モジュールを右クリックして、「**マイグレーション**」 → 「**コネクタ・プロジェクトの更新**」と選択します。
4. ウェルカム・ページに表示されるタスクと警告の内容を確認し、「**次へ**」を選択します。
5. 「プロジェクトの選択」ウィンドウで、「**次へ**」を選択します。

デフォルトでは、コネクタ・プロジェクトと従属プロジェクト (存在する場合) が、ウィザードによってマイグレーションされます。ご使用のプロジェクトに従属プロジェクトがあり、今回それらをマイグレーションしない場合は、「**従属アダプター・プロジェクト (Dependent adapter project)**」リストのチェック・ボックスをクリアします。後でウィザードを再実行して、従属プロジェクトをマイグレーションすることができます。以前にマイグレーションしたプロジェクト、現行バージョンのプロジェクト、およびエラーが起きているプロジェクトは、マイグレーションできないため、選択できません。

6. 「アダプターのマイグレーション」ウィンドウで、オプションとしてマイグレーションでの変更内容を確認することができます。ただし、選択内容を変更することはできません。「**終了**」をクリックします。
7. マイグレーション・ウィザードからのメッセージ (文字列 CWPAD で始まるもの) を、「問題」ビューで確認します。
8. EAR ファイルをマイグレーションする場合、オプションでマイグレーション済みのアダプターと成果物を使って EAR ファイルを新規作成し、WebSphere Process Server または WebSphere Enterprise Service Bus にデプロイすることができます。EAR ファイルのエクスポートおよびデプロイについて詳しくは、本書の該当するトピックを参照してください。

### 結果

プロジェクトまたは EAR ファイルがバージョン 6.1.0 にマイグレーションされます。アダプター・マイグレーション・ウィザードの終了後に、外部サービス・ウィザード を実行する必要はありません。

## バージョン 6.0.2 プロジェクトをマイグレーションせずに更新する

アダプター・プロジェクトの成果物をマイグレーションすることなく、バージョン 6.0.2からバージョン 6.1.0 にアダプターをアップグレードすることができます。

### このタスクを実行する理由および時期

バージョン 6.1.0 ではアダプターの内部名が変更されているため、バージョン 6.0.2 プロジェクトの成果物を更新して、新しい名前を使えるようにしておかないと、WebSphere Integration Developer バージョン 6.1.0 のアダプター・ウィザードを使用することができません。バージョン 6.0.2 プロジェクトを更新するには、マイグレーション・ウィザードを使用します。その後で、WebSphere Integration Developer のクイック・フィックス機能を使用して、プロジェクト成果物のアダプター名を変更します。

### このタスクの手順

1. プロジェクト交換 (PI) ファイルをワークスペースにインポートします。
2. J2EE パースペクティブでプロジェクトを右クリックし、「**マイグレーション**」 → 「**コネクター・プロジェクトの更新**」とクリックします。 アダプター・マイグレーション・ウィザードが開きます。
3. ウェルカム・ページで、「**次へ**」をクリックします。
4. 「プロジェクトの選択」ウィンドウで、依存関係のある成果物プロジェクトを選択解除し、「**終了**」をクリックします。
5. 「クイック・フィックス (Quick Fix)」ウィンドウで、フィックス「**参照しているアダプターの名前を変更 (Rename the referenced adapter)**」が選択されていることを確認してから、「**OK**」をクリックします。
6. エラーが表示されたままならば、「**プロジェクト**」 → 「**クリーン**」をクリックして、更新したばかりのプロジェクトを選択してから、「**OK**」をクリックします。

### 結果

これで、プロジェクトが WebSphere Adapter for FTP バージョン 6.1.0 で使用できるようになりました。

---

## 第 3 章 サンプルおよびチュートリアル

WebSphere Integration Developer のサンプル/チュートリアルのオンライン・ギャラリーには、WebSphere Adaptersを使用する際に役立つサンプルとチュートリアルが用意されています。

サンプル/チュートリアルのオンライン・ギャラリーにアクセスするには、次のようにします。

- WebSphere Integration Developerの開始時に開くウェルカム・ページからアクセスします。WebSphere Adapter for FTPのサンプルとチュートリアルを表示するには、「取得 (Retrieve)」をクリックします。次に、表示されるカテゴリーを参照して、必要なものを選択します。
- 以下の Web サイトにアクセスします。<http://publib.boulder.ibm.com/bpcsamp/index.html>



---

## 第 4 章 デプロイメント用のモジュールの構成

WebSphere Process Server または WebSphere Enterprise Service Bus にデプロイできるようにアダプターを構成するには、WebSphere Integration Developer を使用して、アダプターのデプロイ時に EAR ファイルとしてエクスポートされるモジュールを作成します。次に、作成するビジネス・オブジェクトと、その作成を行うシステムを指定します。以上の手順を行うことで、外部サービスの作成が完了したことになります。

---

### モジュールの構成のためのロードマップ

ランタイム環境で WebSphere Adapter for FTP を使用できるようにするには、まずモジュールを構成する必要があります。このタスクの概要を理解すれば、タスクを達成するのに必要な手順を実行できるようになります。

WebSphere Adapter for FTPのモジュールを構成するには、WebSphere Integration Developerを使用します。以下の図は、構成作業の流れを示しています。また、図の後に示す手順で、この作業の概要を説明します。これらの各ステップの実行方法の詳細については、このロードマップの後に記載されたトピックを参照してください。

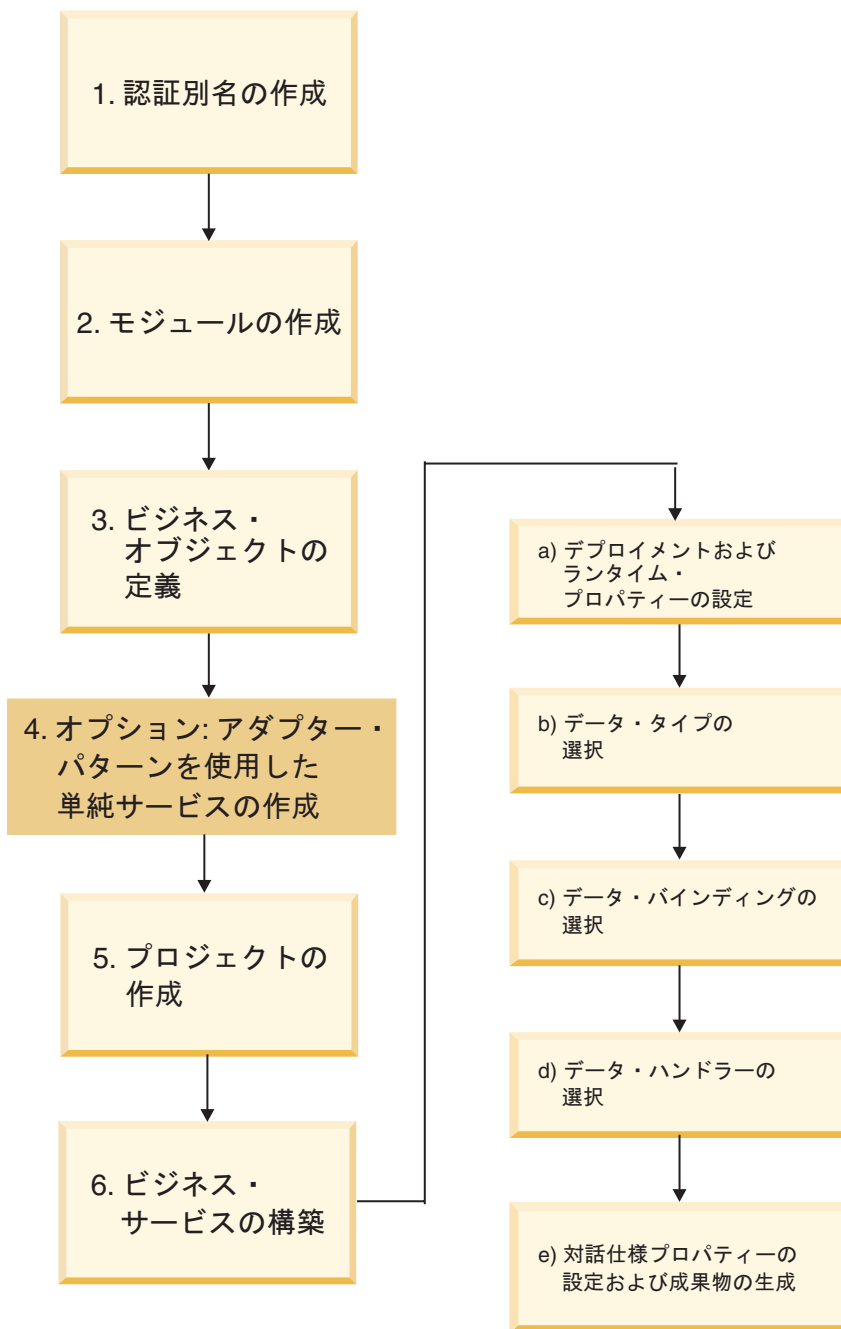


図7. モジュールの構成のためのロードマップ

### モジュールの構成

この作業は、以下の各ステップで構成されています。ここでは概略を説明します。

**注:** これらのステップでは、データ形式変更を必要とするユーザー定義のビジネス・オブジェクトを使用することを想定しています。データ形式変更が不要な汎用ビジネス・オブジェクトを使用する場合には、以下のステップの一部は無視してください。例えば、データ・バインディングおよびデータ・ハンドラーを選択する必要はありません。

1. FTP サーバーにアクセスするための認証別名を作成します。この手順を実行するには、サーバー上の管理コンソールを使用します。
2. WebSphere Integration Developer でモジュールを作成します。ビジネス・オブジェクトを、このモジュール内に作成します。
3. プロジェクトが使用するビジネス・オブジェクトを定義します。
4. アダプター・パターン・ウィザードを使用して、単純サービスを作成します。詳しくは、47 ページの『アダプター・パターン・ウィザードによる単純サービスの作成』を参照してください。
5. WebSphere Integration Developer で外部サービス・ウィザードを使用して、アダプターに関連したファイルの編成に使用するプロジェクトを作成します。
6. WebSphere Integration Developer から外部サービス・ウィザードを実行して、ビジネス・サービスを構築した後、以下の手順を実行します。
  - a. 以下のデプロイメント・プロパティおよびランタイム・プロパティを指定します。
    - 接続プロパティ
    - セキュリティ・プロパティ
    - デプロイメント・オプション
    - 関数セレクター - Inbound のみ
  - b. データ・タイプを選択し、そのデータ・タイプに関連付けられる操作に名前を付けます。操作ごとに、以下の項目を指定します。
    - 操作の種類。例えば、Create、Append、Exists など。
    - 操作をパススルーにするか、ユーザー定義にするかを指定します。
  - c. データ・バインディングを選択します。各データ・タイプには、ビジネス・オブジェクトのフィールドを読み取ったり、ファイルの対応するフィールドを設定したりするために使用するデータ・バインディングが対応しています。
  - d. ビジネス・オブジェクトとネイティブ・フォーマット間の変換を実行するデータ・ハンドラーを選択します。
  - e. 対話仕様プロパティ値を選択し、成果物を生成します。外部サービス・ウィザードを実行した結果生成される出力は、ビジネス・インテグレーション・モジュールに保存されます。ここでは、ビジネス・オブジェクト (1 つまたは複数)、およびインポートまたはエクスポート・ファイルが格納されます。

---

## 認証別名の作成

認証別名は、アダプターがFTP サーバーへのアクセスに使用するパスワードを暗号化する機能です。認証別名が作成されると、アダプターの構成時に (ユーザー ID とパスワードを直接入力する代わりに) この名前を使用できます。アダプター・プロパティは暗号化されないため、パスワードを直接入力すると、他のユーザーが確認できる平文で保管されてしまいます。認証別名の使用は、外部サービス・ウィザードのデフォルト選択項目です。

始める前に

認証別名を作成するには、WebSphere Process Server または WebSphere Enterprise Service Bus の管理コンソールへのアクセス権が必要です。以下の手順は、WebSphere Integration Developer を介して管理コンソールへのアクセス権を取得する方法を示しています。管理コンソールを直接使用する (WebSphere Integration Developerを経由していない) 場合は、管理コンソールにログインし、ステップ 2 をスキップしてください。

### このタスクを実行する理由および時期

認証別名を作成するには、以下の手順を実行します。

### このタスクの手順

1. 管理コンソールを開始します。

管理コンソールを開始するには、以下の手順を実行します。

- a. 「スタート」 → 「プログラム」 → 「IBM Software Development Platform」 → 「WebSphere Integration Developer 6.1」 → 「WebSphere Integration Developer 6.1」 をクリックします。
  - b. ワークスペースを指定するように指示された場合は、デフォルト値を受け入れます。(ワークスペースとは、WebSphere Integration Developer がプロジェクトを保管するディレクトリーのことです。)
  - c. WebSphere Integration Developer ウィンドウが表示されたら、「**Business Integration** パースペクティブヘジャンプ」をクリックします。
  - d. 「**サーバー (Servers)**」タブをクリックします。
  - e. サーバーで「**開始済み**」という状況が表示されない場合は、サーバーの名前(例えば、「**WebSphere Process Server**」)を右クリックして、「**開始**」をクリックします。
  - f. サーバーの名前を右クリックし、「**管理コンソールの実行 (Run administrative console)**」をクリックします。
  - g. 管理コンソールにログオンします。管理コンソールにユーザー ID およびパスワードが必要な場合は、ID およびパスワードを入力して、「**ログイン**」をクリックします。ユーザー ID およびパスワードが不要な場合は、「**ログイン**」をクリックします。
2. 管理コンソールで、「**セキュリティ**」 → 「**管理、アプリケーション、およびインフラストラクチャーの保護 (Secure administration, applications, and infrastructure)**」をクリックします。
  3. 「**認証**」の下の、「**Java 認証および承認サービス (Java Authentication and Authorization Service)**」 → 「**J2C 認証データ (J2C authentication data)**」をクリックします。



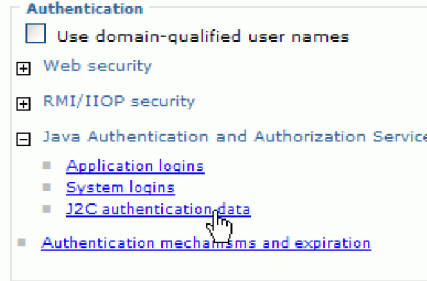


図 8. 「管理、アプリケーション、およびインフラストラクチャーの保護 (Secure administration, applications, and infrastructure)」ウィンドウの「認証」セクション

4. 認証別名を作成します。
  - a. 表示された J2C 認証別名のリストで、「新規作成」をクリックします。
  - b. 「構成」タブをクリックしてから、「別名」フィールドに認証別名の名前を入力します。
  - c. FTP サーバーへの接続の確立に必要なユーザー ID およびパスワードを入力します。
  - d. オプション: 別名の説明を入力します。
  - e. 「OK」をクリックします。

新規に作成された別名が表示されます。

別名のフルネームに注意してください。このフルネームは、後続の構成ウィンドウで使用する名前です。

- f. 「保管」をクリックした後、再度「保管」をクリックします。
5. 「新規 (New)」をクリックします。

## 結果

アダプター・プロパティを構成する際に使用する認証別名が作成されました。

---

## モジュールの作成

WebSphere Integration Developer でモジュールを作成します。モジュールを使用して、プロジェクトが使用するビジネス・オブジェクトを定義できます。

### このタスクの手順

1. WebSphere Integration Developer が現在実行されていない場合は、開始します。
  - a. 「スタート」 → 「プログラム」 → 「IBM WebSphere」 → 「Integration Developer V6.1.0」 → 「WebSphere Integration Developer V6.1.0」をクリックします。
  - b. ワークスペースを指定するようにプロンプトが出された場合は、デフォルト値を受け入れるか、または別のワークスペースを選択します。

ワークスペースとは、WebSphere Integration Developer がプロジェクトを保管するディレクトリのことです。

- c. オプション: WebSphere Integration Developer ウィンドウが表示されたら、「**Business Integration** パースペクティブヘジャンプ」をクリックします。
2. WebSphere Integration Developer ウィンドウの「ビジネス・インテグレーション」セクション内のいずれかの場所を右クリックします。

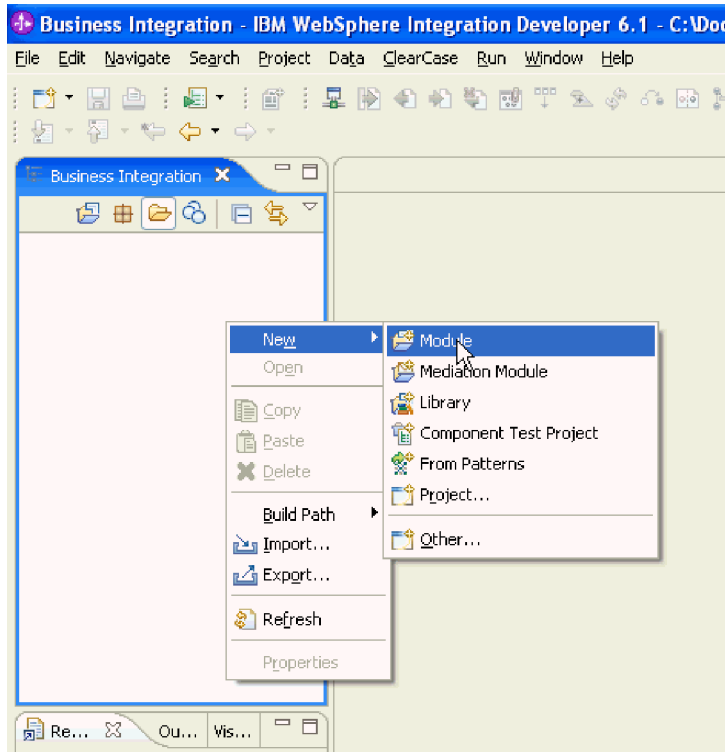


図9. ウィンドウの「ビジネス・インテグレーション」セクション

3. 「新規モジュール」ウィンドウで、「モジュール名」に新しいモジュールの名前を入力します。

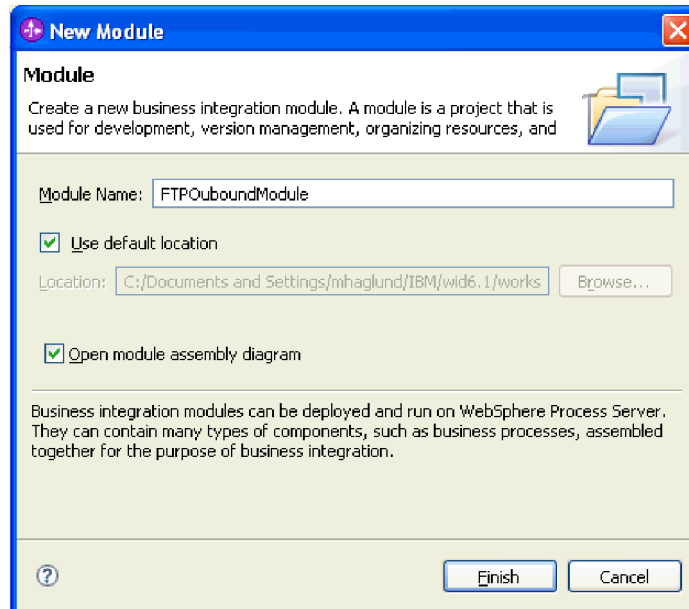


図 10. 「新規モジュール」 ウィンドウ

4. 「終了」 をクリックします。

### 結果

「ビジネス・インテグレーション」 ウィンドウに新しいモジュールがリストされます。

### 次のタスク

プロジェクトを作成します。これは、アダプターに関連付けられたファイルを編成するために使用されます。

---

## ビジネス・オブジェクトの定義

次のトピックで作成するプロジェクトが使用することになる WebSphere Integration Developer 内のビジネス・オブジェクトを事前に定義します。

### このタスクの手順

1. WebSphere Integration Developer ウィンドウの「ビジネス・インテグレーション」 セクション内にある新しいモジュールを展開します。
2. 「データ・タイプ」 フォルダを右クリックして、「新規」 > 「ビジネス・オブジェクト」 を選択します。

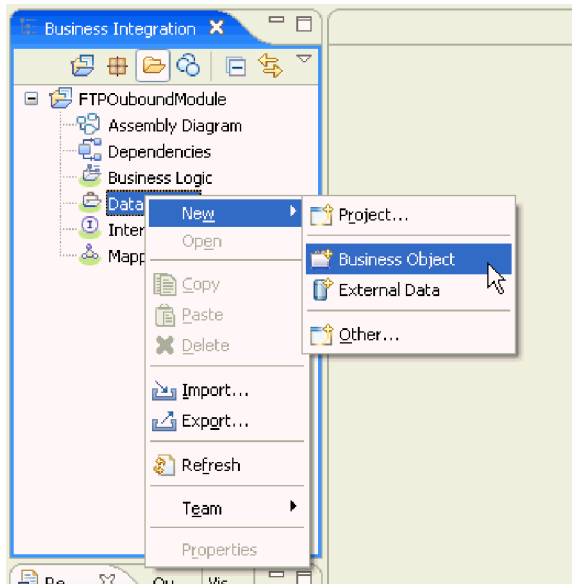


図 11. 新規ビジネス・オブジェクトの選択ビュー

3. 「ビジネス・オブジェクト」ウィンドウで、「名前」に新しい名前を入力します。

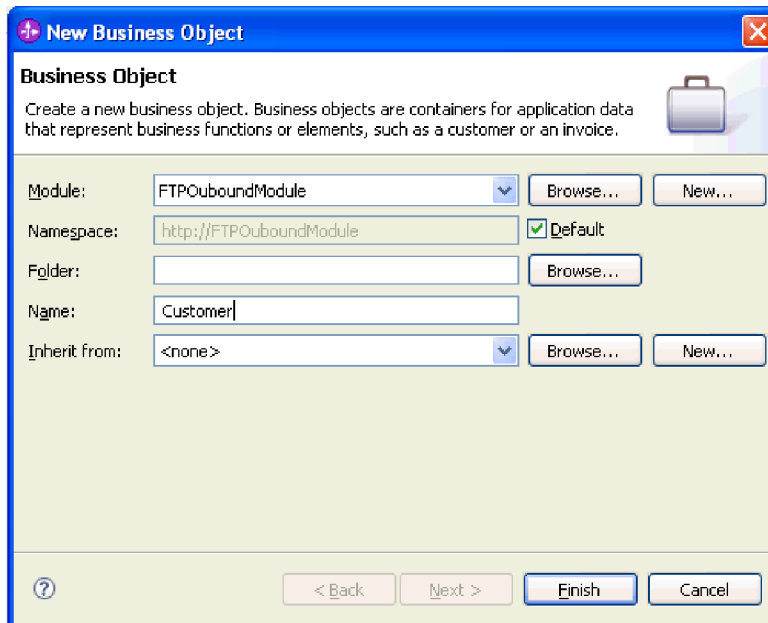


図 12. 「ビジネス・オブジェクト」ウィンドウ

4. 「終了」をクリックします。「データ・タイプ」フォルダーに新しいビジネス・オブジェクトが追加されます。
5. 「フィールドをビジネス・オブジェクトに追加します」アイコンをクリックして、ビジネス・オブジェクトに必要なフィールドを追加します。

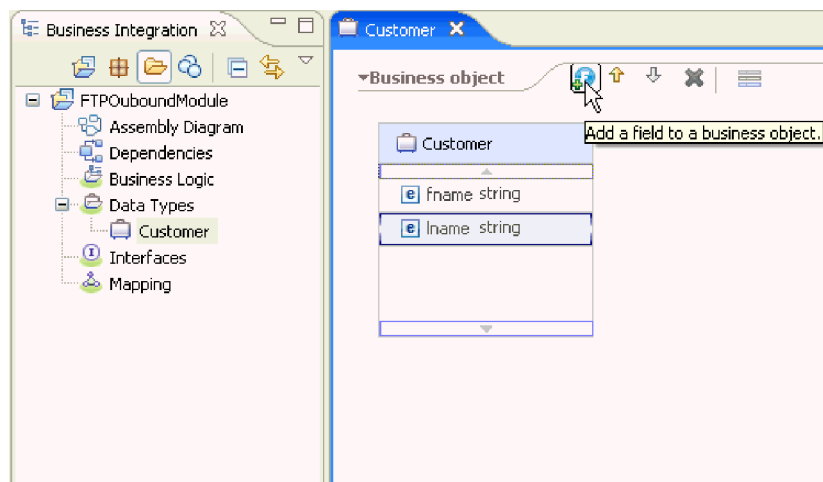


図 13. 「ビジネス・オブジェクト・フィールドの追加 (Add Business object fields)」アイコン

6. 「保管」アイコンをクリックします。
7. 作成する各ビジネス・オブジェクトについて、前述の手順を繰り返します。

### 結果

新しいビジネス・オブジェクトが定義されます。

### 次のタスク

プロジェクトを作成します。これは、アダプターに関連付けられたファイルを編成するために使用されます。

## アダプター・パターン・ウィザードによる単純サービスの作成

アダプター・パターンによって、アダプターを使用して単純サービスを素早く簡単に作成できます。

### 始める前に

RetrieveAFileModule というモジュールと、Customer というビジネス・オブジェクトがすでに作成されています。

### このタスクを実行する理由および時期

Adapter for FTP では、以下のアダプター・パターンを使用できます。

表 4.

アダプター・パターン	説明
Inbound FTP パターン	FTP Inbound パターンは、FTP サーバー上の特定のディレクトリ内にあるファイルを取得するサービスを作成します。ファイルが XML 形式ではない場合は、ファイル内容の形式をビジネス・オブジェクトに変換するためのデータ・ハンドラーを指定できます。ファイル内容にデータ構造のコピーが複数含まれている場合には、その内容を分割して処理できます。

表 4. (続き)

アダプター・パターン	説明
Outbound FTP パターン	FTP Outbound パターンは、FTP サーバー上の特定のディレクトリー内にあるファイルにデータを保管するサービスを作成します。出力形式として XML 形式が要求されていなければ、ビジネス・オブジェクトをファイル内容の形式に変換するためのデータ・ハンドラーを指定できます。

この例では、ファイル・システムからファイルを受け取って処理する FTP Inbound サービスを作成します。この例で作成したサービスは、ファイルを読み込み、区切り文字に応じて個別のファイルに内容を分割します。

アダプター・パターン・ウィザードを使用してサービスを作成するには、以下のステップを実行します。

#### このタスクの手順

1. WebSphere Integration Developer ウィンドウの「ビジネス・インテグレーション」セクション内の「RetrieveAFileModule」を右クリックし、「新規」→「パターンから」を選択します。「パターンから新規作成」ウィンドウが開きます。
2. 「リモート・ファイルから読み取るインバウンド FTP サービスを作成する service to read from a remote file)」を選択し、「次へ」をクリックします。

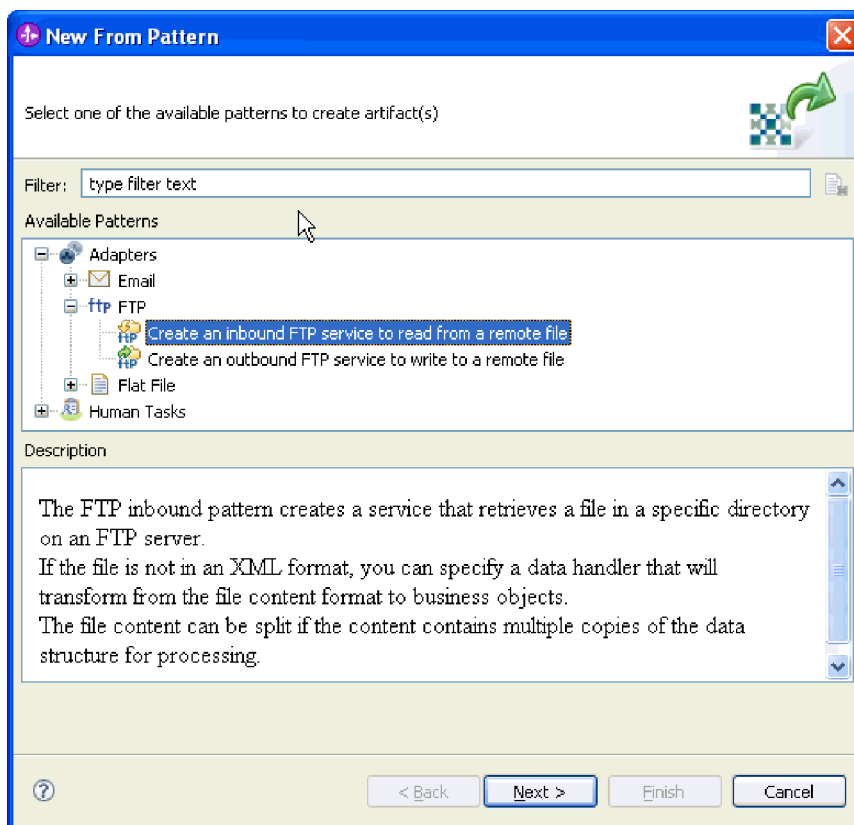


図 14. 「パターンから新規作成」ウィンドウ

3. 「新規インバウンド FTP サービス」ウィンドウで、FTPInboundInterface のような分かりやすい名前に変更し、「次へ」をクリックします。

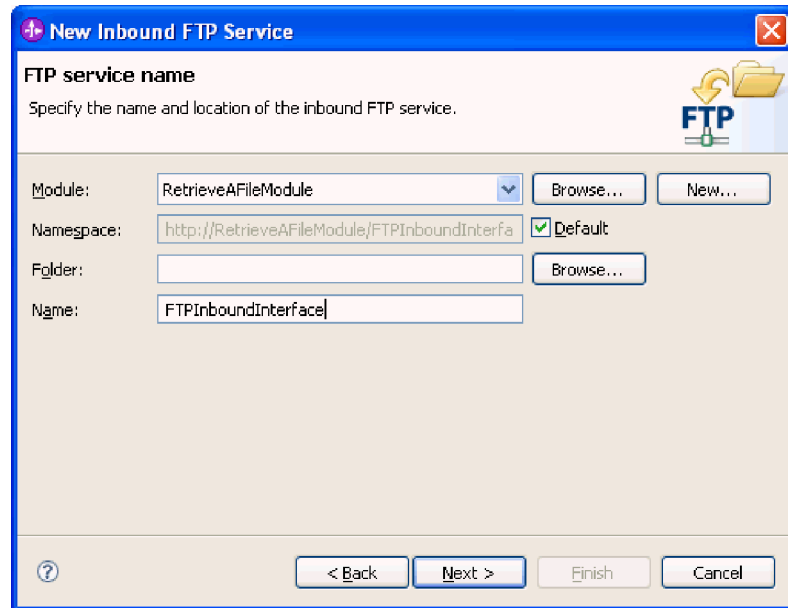


図 15. 「FTP サービス名」ウィンドウ

4. 「ビジネス・オブジェクトおよび場所 and location)」ウィンドウで「参照」をクリックして、「Customer」ビジネス・オブジェクトにナビゲートします。
5. 「リモート・ディレクトリー」と「ローカル・ステーjing・ディレクトリー」の両方を指定し、「次へ」をクリックします。

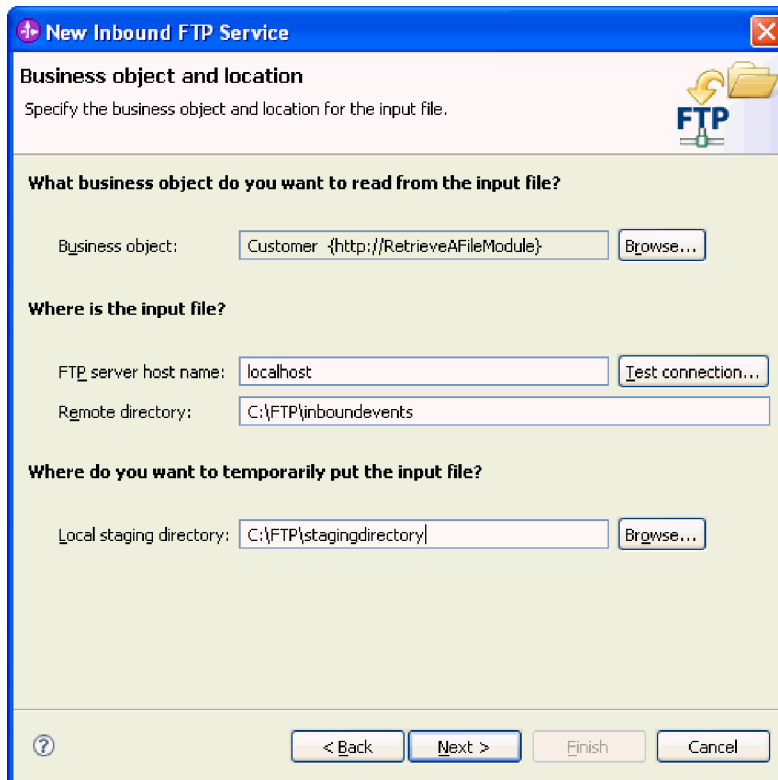


図 16. 「ビジネス・オブジェクトおよび場所 and location)」ウィンドウ

6. 「FTP サーバーのセキュリティー・クレデンシャル」ウィンドウで、「既存の JAAS 別名を使用する」または「ユーザー名とパスワードの使用」のいずれかを選択し、「次へ」をクリックします。



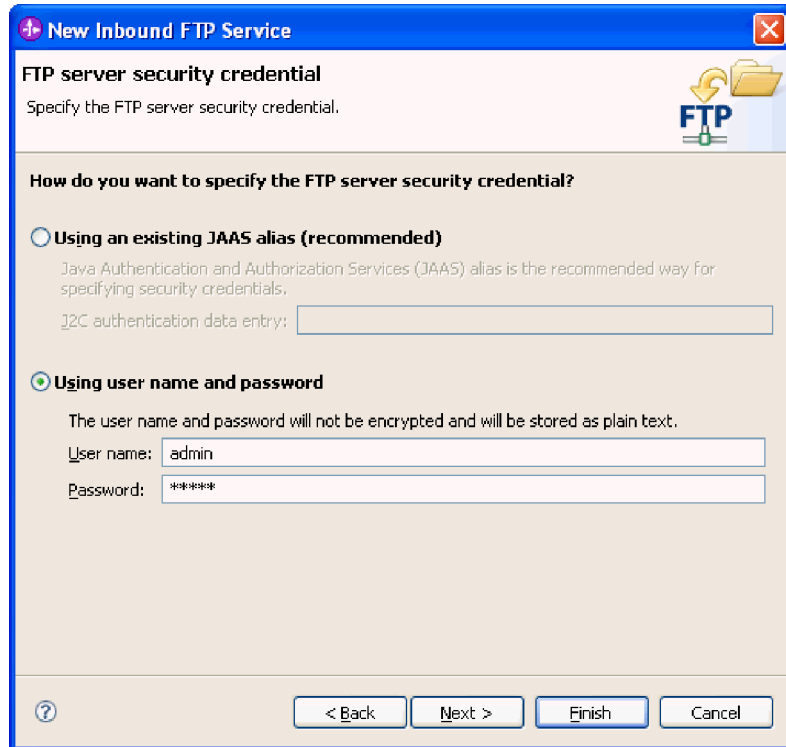


図 17. 「FTP サーバーのセキュリティー・クレデンシャル」ウィンドウ

7. 「入力ファイル・フォーマットとファイル内容の分割オプション and file content split option)」ウィンドウで、デフォルトの XML 入力ファイル形式を受け入れるか、「その他」を選択して、ネイティブ・フォーマットからビジネス・オブジェクト形式にデータを変換するためのデータ・ハンドラーを指定します。
8. 「区切り文字でファイル内容を分割する」を選択し、区切り文字を入力します。この例では ####;¥n を使用します。「次へ」をクリックします。

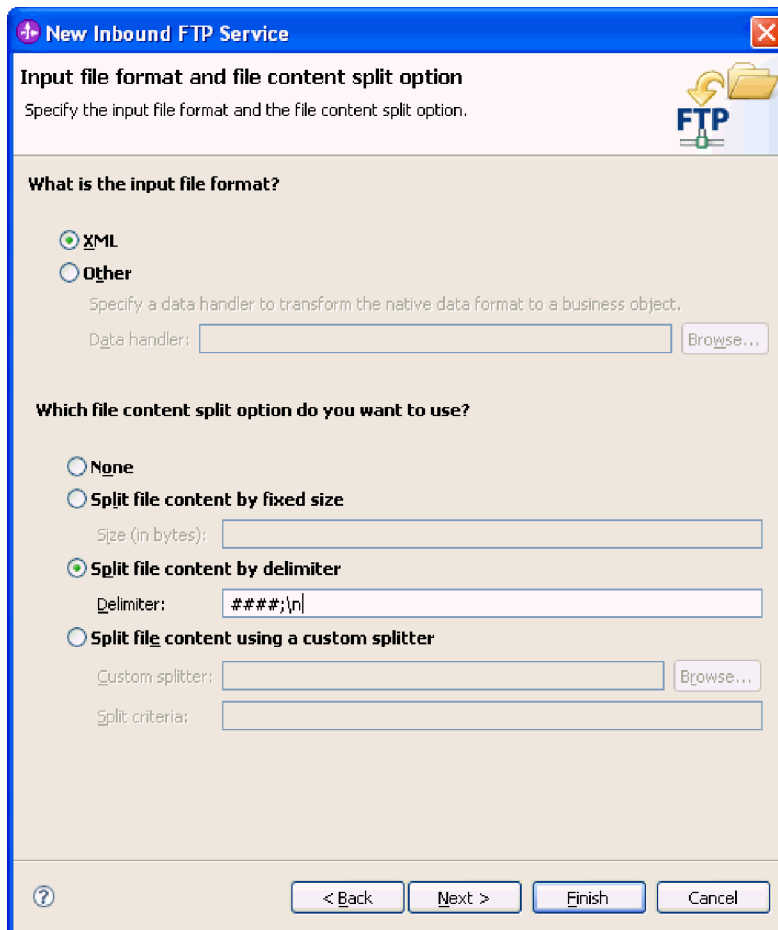


図 18. 「入力ファイル・フォーマットとファイル内容の分割オプション and file content split option)」ウィンドウ

9. 「アーカイブ・ディレクトリーおよびラッパー・ビジネス・オブジェクト」ウィンドウで、「ローカル・アーカイブ・ディレクトリー」を指定します。この例では、FTP¥inboundarchive です。アダプター固有の情報を組み込む場合は、「入力ファイル情報を追加するためにラッパー・ビジネス・オブジェクトを使用する」を選択します。「終了」をクリックします。

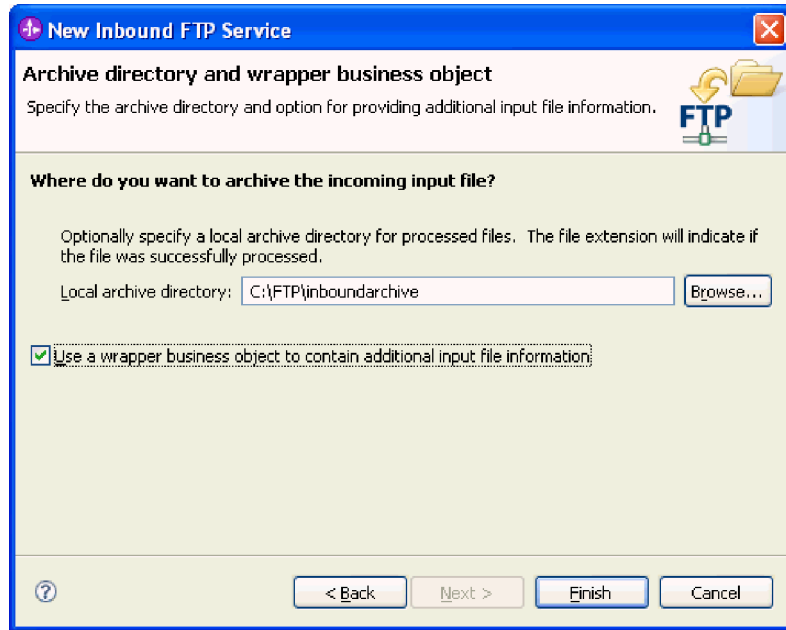


図 19. 「アーカイブ・ディレクトリーおよびラッパー・ビジネス・オブジェクト」ウィンドウ

## 結果

Inbound サービスが作成され、以下の成果物が組み込まれます。

表 5.

成果物	名前	説明
エクスポート	FTPInboundInterface	エクスポートは、モジュールを外部 (ここでは WebSphere Adapter for FTP) に公開します。
ビジネス・オブジェクト	Customer、CustomerWrapper	Customer ビジネス・オブジェクトには、顧客データ用のフィールド (氏名、住所、市、都道府県など) が含まれています。CustomerWrapper ビジネス・オブジェクトには、アダプター固有の情報用の追加フィールドが含まれています。
インターフェース	FTPInboundInterface	このインターフェースには、呼び出すことができる操作が含まれます。
操作	emitCustomerInput	emitCustomerInput は、このインターフェースの唯一の操作です。

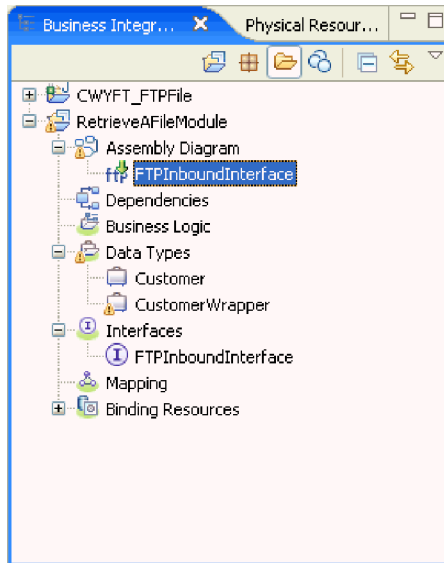


図 20. WebSphere Integration Developer ウィンドウの「ビジネス・インテグレーション」セクションに表示された新規成果物

## プロジェクトの作成

モジュールの作成とデプロイのプロセスを開始するには、WebSphere Integration Developer の外部サービス・ウィザードを開始します。このウィザードはプロジェクトを作成します。これは、モジュールに関連付けられたファイルを編成するために使用されます。

### 始める前に

FTP サーバーへの接続の確立に必要な情報を収集済みであることを確認します。例えば、FTP サーバー の名前または IP アドレス、およびアクセスに必要なユーザー ID とパスワードが必要です。

### このタスクを実行する理由および時期

外部サービス・ウィザードを開始して、WebSphere Integration Developer のアダプター用のプロジェクトを作成します。既存のプロジェクトが存在する場合は、ウィザードで新規作成する代わりに、そのプロジェクトを選択してもかまいません。

外部サービス・ウィザードを開始し、プロジェクトを作成するには、次の手順を実行します。

### このタスクの手順

1. WebSphere Integration Developer が現在実行されていない場合は、開始します。
  - a. 「スタート」 → 「プログラム」 → 「IBM Software Development Platform」 → 「WebSphere Integration Developer 6.1」 → 「WebSphere Integration Developer 6.1」 をクリックします。
  - b. ワークスペースを指定するようにプロンプトが出された場合は、デフォルト値を受け入れるか、または別のワークスペースを選択します。

ワークスペースとは、WebSphere Integration Developer がプロジェクトを保管するディレクトリーのことです。

- c. WebSphere Integration Developer ウィンドウが表示されたら、「**Business Integration** パースペクティブヘジャンプ」をクリックします。
2. 外部サービス・ウィザードを開始するには、「ファイル」→「新規」→「外部サービス」をクリックします。
3. 「新規外部サービス」ウィンドウで、「アダプター」が選択済みであることを確認して、「次へ」をクリックします。

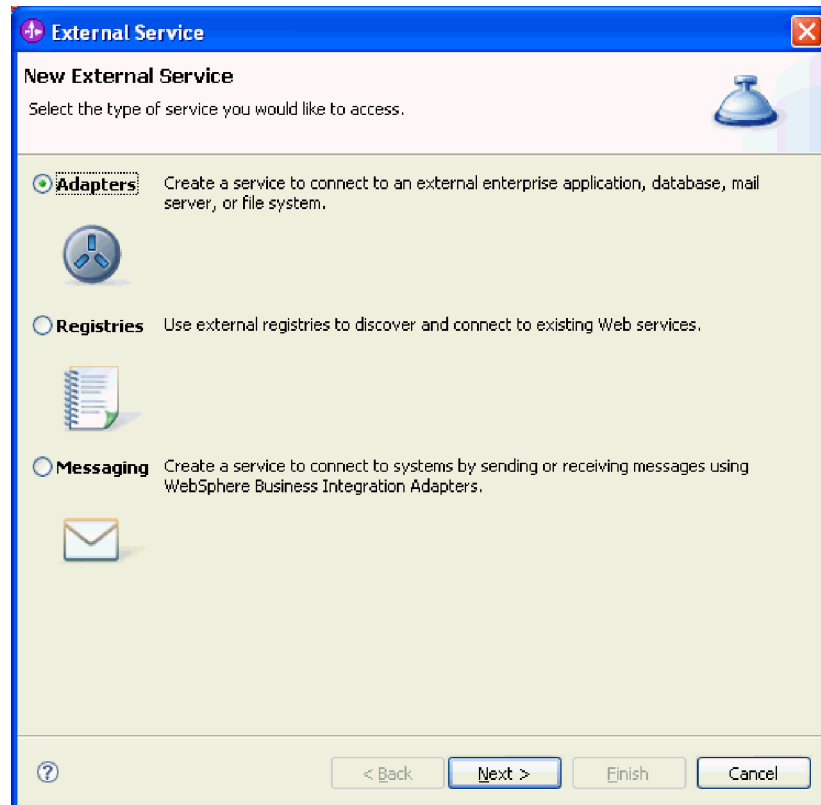


図 21. 「新規外部サービス」ウィンドウ

4. 「アダプターの選択」ウィンドウで、「**IBM WebSphere Adapter for FTP**」を選択し、「次へ」をクリックします。

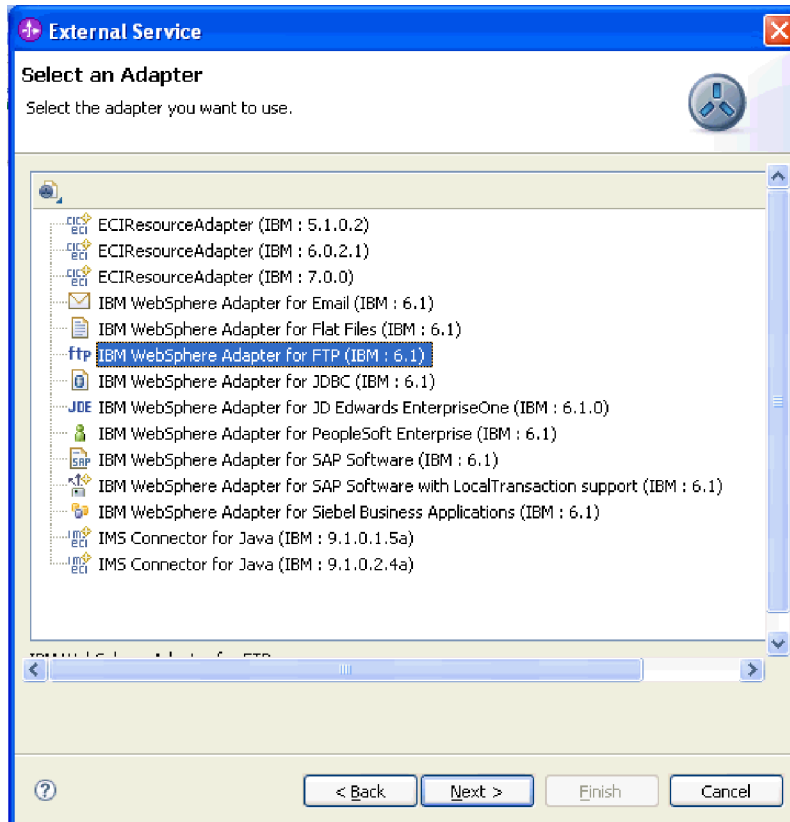


図 22. 「アダプターの選択」 ウィンドウ

5. 「アダプターのインポート」 ウィンドウで、「コネクター・プロジェクト」ワールドのデフォルトの名前をそのまま使用するか、または新しい名前を入力して、「ターゲット・ランタイム」を選択します。

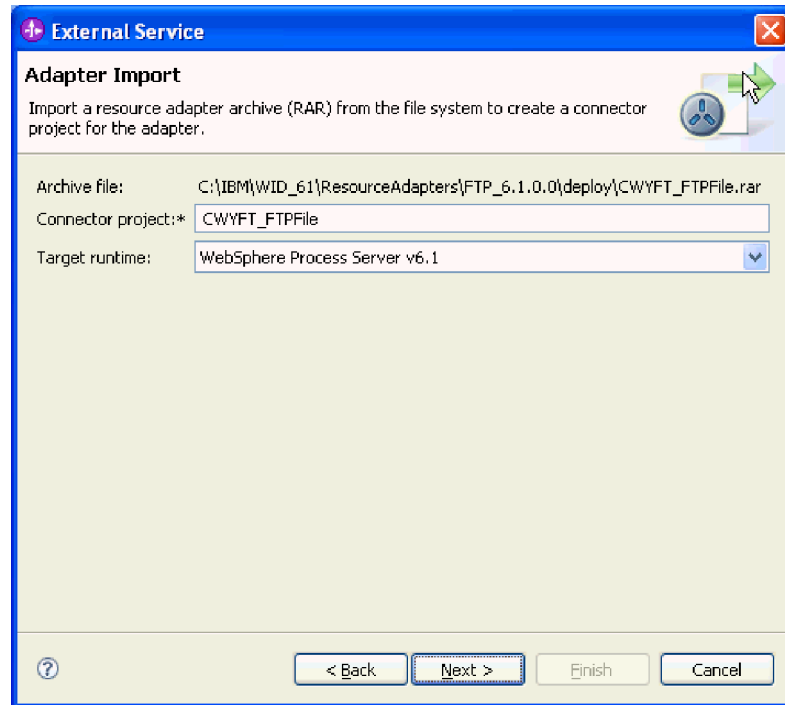


図 23. 「アダプターのインポート」ウィンドウ

6. 「次へ」をクリックします。

### 結果

新規プロジェクトが作成され、「ビジネス・インテグレーション」ウィンドウにリストされます。

---

## Outbound 処理のモジュールの構成

アダプターを Outbound 処理に使用するようにモジュールを構成するには、WebSphere Integration Developer 内で外部サービス・ウィザードを使用して、ビジネス・サービスを作成し、データ変換処理を指定して、ビジネス・オブジェクト定義および関連する成果物を生成します。

## デプロイメント・プロパティおよびランタイム・プロパティの設定

外部サービス・ウィザードが FTP サーバーの接続に使用するデプロイメント・プロパティとランタイム・プロパティを指定します。

### 始める前に

接続プロパティを指定するには、外部サービス・ウィザードを開始しておく必要があります。

### このタスクを実行する理由および時期

外部サービス・ウィザードでは、ディスクバリーおよびサービス記述作成のためのデータベースへの接続を行うために、この情報が必要となります。

### このタスクの手順

1. 「処理方向」ウィンドウで「**Outbound**」を選択し、「次へ」をクリックします。
2. 「サーバー構成プロパティ (Server Configuration Properties)」ウィンドウで、「**J2C 認証データ項目**」を指定します。これは、FTP サーバー上にセットアップする認証別名です。名前は大文字小文字が区別されます。詳しくは、『認証別名の作成』を参照してください。
3. 「**コネクター・プロジェクトのデプロイ**」フィールドで、モジュールにアダプター・ファイルを組み込むかどうかを指定します。次のいずれかのオプションを選択します。

- 「**単一アプリケーションが使用するモジュールで (With module for use by single application)**」

アダプター・ファイルをモジュール内に組み込むと、モジュールをすべてのアプリケーション・サーバーにデプロイすることができます。単一のモジュールによってアダプターを使用する場合、または複数のモジュールを使ってさまざまなバージョンのアダプターを実行する必要がある場合には、組み込みアダプターを使用します。組み込みアダプターを使用すると、単一のモジュールのアダプターのみをアップグレードすることができるため、その他のモジュールのアダプター・バージョンが変更されて動作が不安定になることはありません。

- 「**複数アプリケーションが使用するサーバー上 (On server for use by multiple applications)**」

モジュール内にアダプター・ファイルを組み込まない場合は、このモジュールを実行させるアプリケーション・サーバーごとに、アダプター・ファイルをスタンドアロン・アダプターとしてインストールする必要があります。複数のモジュールで同じバージョンのアダプターを使用できる場合に、中央の場所でアダプターを管理するには、スタンドアロン・アダプターを使用します。スタンドアロン・アダプターでは、複数のモジュールに対して単一のアダプター・インスタンスを実行することによって、必須リソースを削減することもできます。

4. ご使用のモジュールについて、以下の FTP システム接続情報を定義します。このウィンドウのプロパティについて詳しくは、本書の管理接続ファクトリー・プロパティに関するトピックを参照してください。
  - 「**ホスト名**」 - FTP サーバーのホスト名を指定。
  - 「**ディレクトリー**」 - FTP サーバー上の出力ディレクトリーを指定。
  - 「**ポート番号**」 - FTP サーバーのポート番号を指定。
  - 「**プロトコル**」 - 通常 FTP またはセキュア FTP (FTPS) のいずれかを指定。
5. オプション: 「**拡張**」をクリックして、追加プロパティ (2 次 FTP サーバーの処理、BiDi フォーマットの設定、ステーキング・ディレクトリー、およびシークエンス・ファイルの選択を制御するプロパティなど) を指定します。



6. オプション: ウィンドウの「サービス・プロパティ」セクションで、アダプターの実行時に使用する Java Authentication and Authorization Service (JAAS) 別名を指定できます。
7. 「データ・バインディング」フィールドで、以下のいずれかを選択します。
  - **すべての操作にデフォルトのデータ・バインディング 'FTPFileBaseDataBinding' を使用**

サービスで使用されるすべての操作に対して、構成されていないデータ・バインディングを使用します。
  - **すべての操作にデータ・バインディング構成を使用**

サービスで使用されるすべての操作に対して、構成済みのデータ・バインディングを使用します。
  - **操作ごとにデータ・バインディングを指定**

デフォルトのバインディングは指定されません。サービスで使用される操作ごとに、特定のデータ・バインディングを選択します。
8. オプション: このモジュールにロギング・レベルを定義する場合は、「ウィザードのロギング・プロパティを変更します」チェック・ボックスを選択します。

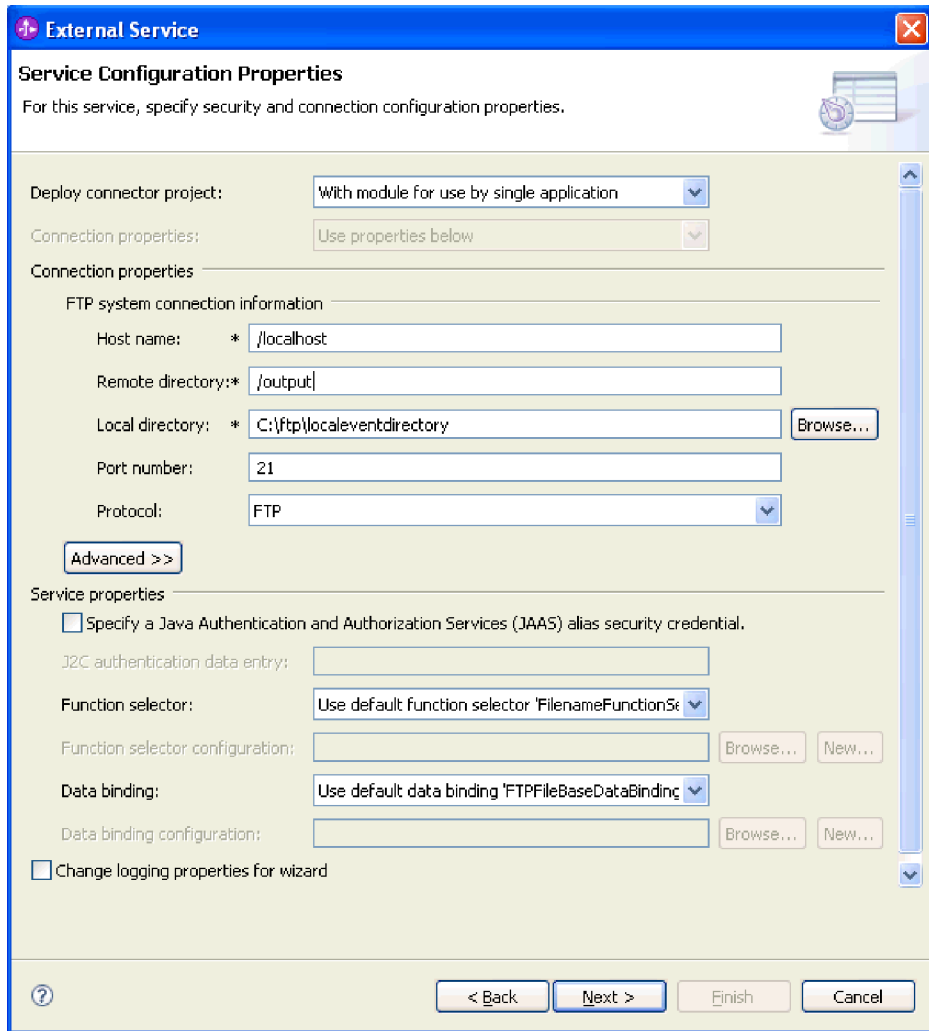


図 24. 「サーバー構成プロパティ (Server Configuration Properties)」 ウィンドウ

## 結果

外部サービス・ウィザードは、FTP サーバーへの接続に必要な情報を保有します。

「すべての操作にデフォルトのデータ・バインディング 'FTPFileBaseDataBinding' を使用」または「操作ごとにデータ・バインディングを指定」のいずれかのデータ・バインディング構成オプションを選択した場合は、「次へ」をクリックして、ウィザードでの作業を続行し、データ・タイプを選択してそのデータ・タイプに関連付けられる操作に名前を付けます。

データ・バインディング構成オプションで「すべての操作にデータ・バインディング構成を使用」を選択した場合は、63 ページの『データ・バインディングの構成』に進んでください。

## データ・タイプおよび操作名の選択

データ・タイプを選択し、データ・タイプに関連付けられる操作に名前を付けるには、外部サービス・ウィザードを使用します。 Outbound 通信の場合、外部サービス・ウィザードでは、ユーザー定義タイプ、汎用 FTP ビジネス・オブジェクト、お

よびビジネス・グラフ付きの汎用 FTP ビジネス・オブジェクトの 3 種類の異なるデータ・タイプから選択することができます。各データ・タイプは、ビジネス・オブジェクト構造に対応しています。

### 始める前に

以下の手順を実行する前に、FTP サーバーにアダプターを接続するための接続プロパティを指定しておく必要があります。

### このタスクを実行する理由および時期

データ・タイプを選択し、それに関連付けられる操作に名前を付けるには、以下の手順を実行します。

### このタスクの手順

1. 「操作」ウィンドウで、「追加」をクリックします。

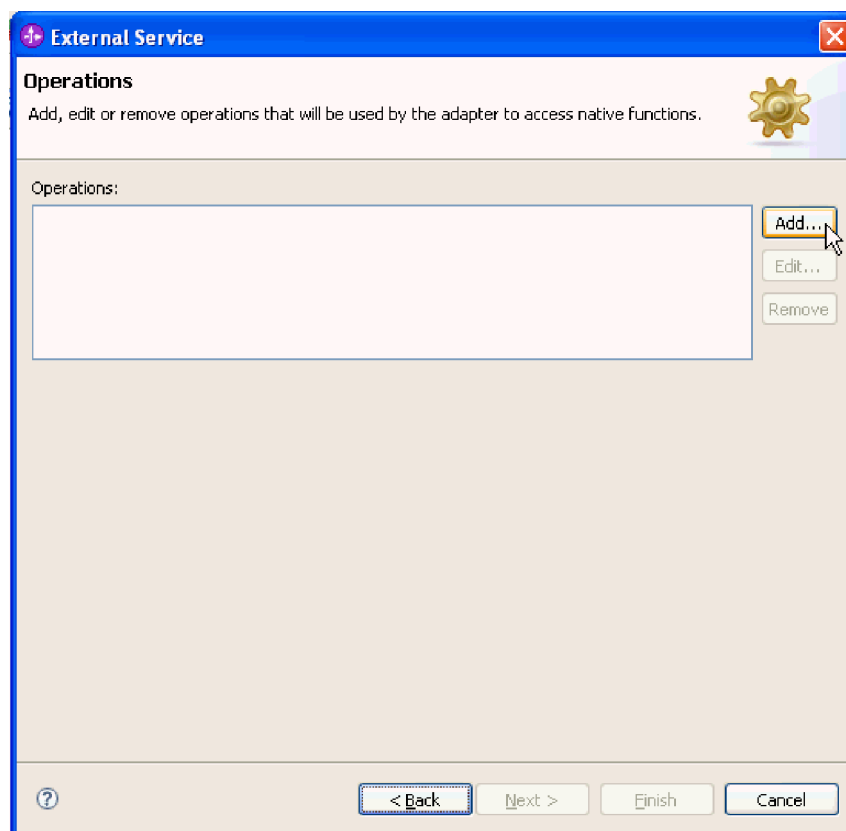


図 25. 操作の追加

2. 「操作の追加」ウィンドウで、「操作の種類」および「操作入力のデータ型」を選択し、「次へ」をクリックします。「ユーザー定義タイプ」を選択する場合は、そのタイプをサポートするユーザー定義データ・バインディングを指定する必要があります。「汎用 FTP ビジネス・オブジェクト」にデータ・バインディングを指定した場合は、汎用入力タイプのみがサポート対象操作となります。

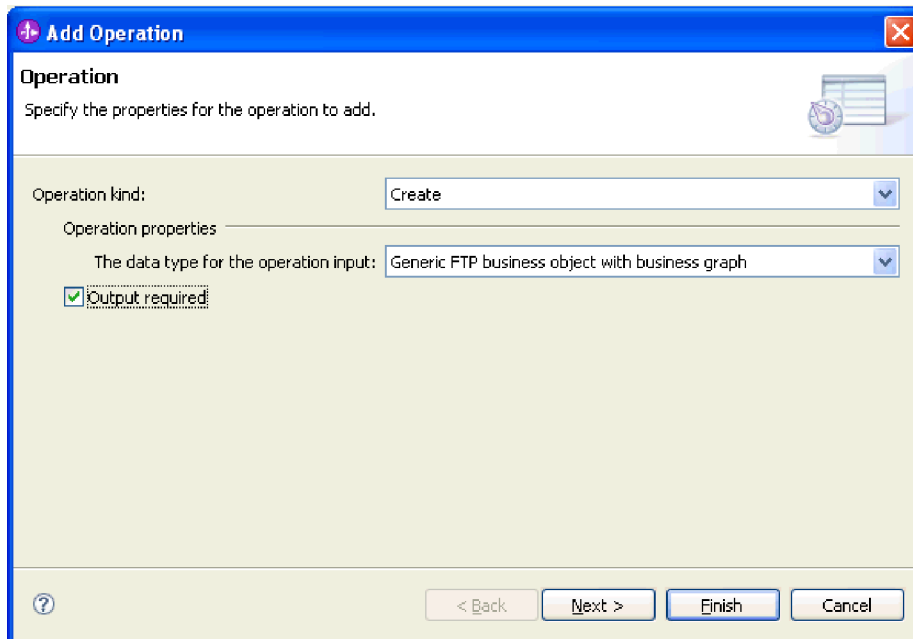


図 26. データ・タイプの選択

3. オプション: 「出力が必要」チェック・ボックスを選択すると、ファイル名が返されるようにすることができます。固有のファイル名を生成する場合、またはファイルの順序付けを可能にする場合に、これを選択します。詳しくは、対話仕様プロパティ `GenerateUniqueFile` および `FileSequenceLog` の説明を参照してください。 `Exists`、`List`、および `Retrieve` の各操作については、出力が必要なため、「出力が必要」チェック・ボックスにチェック・マークを付けて、有効にします。 `Delete` 操作については、出力は戻されないため、「出力が必要」チェック・ボックスのチェック・マークを外して、無効にします。「次へ」をクリックします。
4. 「操作」ウィンドウで、「操作名」を入力します。操作には分かりやすい名前を付けてください。このモジュールを新規のカスタマー・レコードの作成に使用する場合には、`createCustomer` のような名前を指定します。アダプターで実行可能な操作のタイプについて詳しくは、6 ページの表 1 のトピックを参照してください。

注: 名前にスペースを含めることはできません。

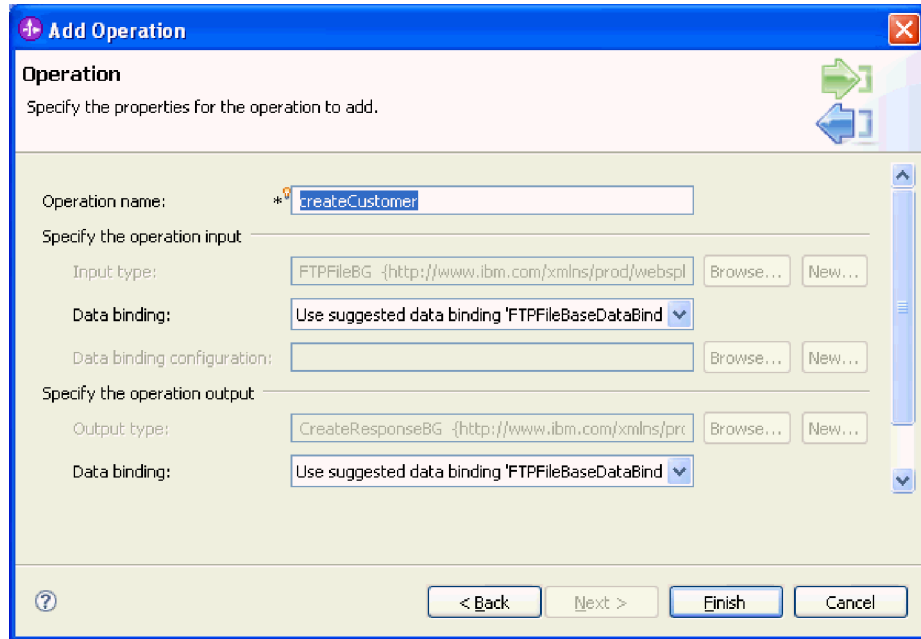


図 27. 操作への名前の指定

## 結果

モジュールのデータ・タイプが定義され、そのデータ・タイプに関連した操作に名前が付けられます。

構成済みのデータ・バインディングを使用する場合は、現在の外部サービス・ウィザードウィンドウで作業を続行し、モジュールで使用するデータ・バインディングを追加して構成します。

デフォルトのデータ・バインディングを使用する場合は、72 ページの『対話仕様プロパティの設定およびサービスの生成』に進んでください。

## データ・バインディングの構成

各データ・タイプには、それに対応するデータ・バインディングが存在し、ビジネス・オブジェクトのフィールドを読み取って、ファイル内の対応するフィールドを設定するために使用されます。外部サービス・ウィザードで、モジュールにデータ・バインディングを追加し、追加したデータ・バインディングを、使用するデータ・タイプに合うように構成します。このようにして、アダプターはファイル内のフィールドに、ビジネス・オブジェクト内で受け取った情報を取り込む方法を識別します。

### 始める前に

データ・タイプを選択し、そのデータ・タイプに関連付ける操作名を選択しておく必要があります。

**注:** データ・バインディングは、外部サービス・ウィザードを実行する前に、WebSphere Integration Developer を使用して構成できます。これを行うには、

WebSphere Integration Developer で「新規」 → 「リソース構成 (Resource configuration)」を選択して、本書に記載されたデータ・バインディング・ウィンドウを完了してください。

### このタスクを実行する理由および時期

モジュール用のデータ・バインディングを追加し、構成するには、以下の手順を実行します。

### このタスクの手順

1. ウィンドウの「操作入力の指定」セクションの「データ・バインディング構成」フィールドの横にある「新規」を選択します。

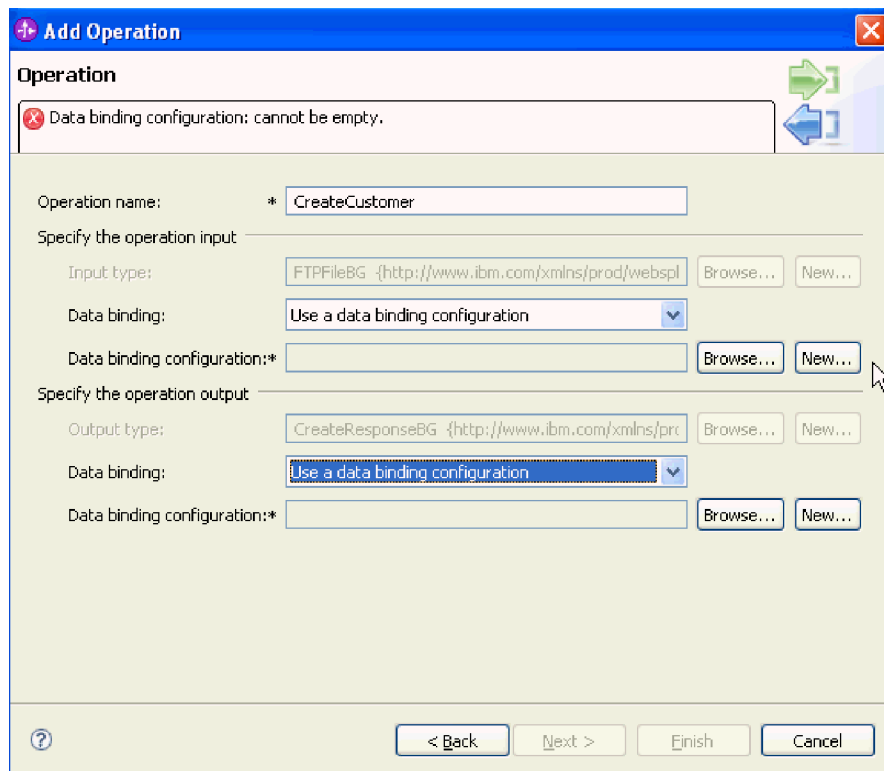


図 28. 「操作」ウィンドウ

2. データ・バインディングの「名前」を入力し、「次へ」をクリックします。データ・バインディングはデータ・ハンドラーへのポインターを持つため、その内容を名前に反映させることをお勧めします。例えば、FTPOutboundDB\_XML または FTPOutboundDB\_De1im1 のようにします。

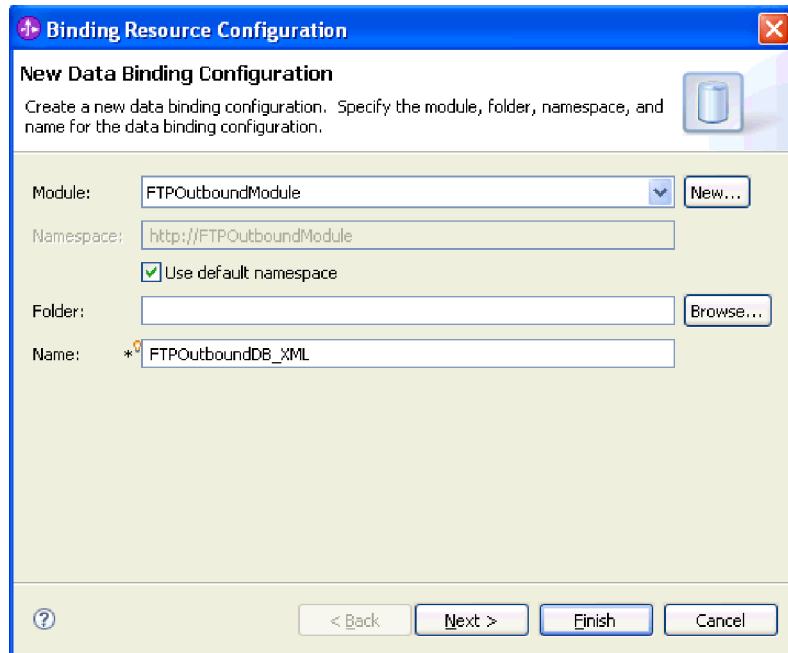


図 29. データ・バインディングの名前の指定

3. 「構成タイプの選択」ウィンドウでは、「データ・バインディング」ラジオ・ボタンを選択したままにします。
4. 「参照」をクリックして、クラス名を選択します。ここでいう「クラス」とは、このモジュールを作成する過程でデータ・バインディングに関連付けるデータ・バインディング・クラスを指します。
5. 「データ・バインディング選択」ウィンドウで、「データ・バインディング・クラスの表示」ラジオ・ボタンをクリックして選択します。
6. データ・タイプに応じた正しいデータ・バインディング・クラスを選択し、「OK」をクリックします。

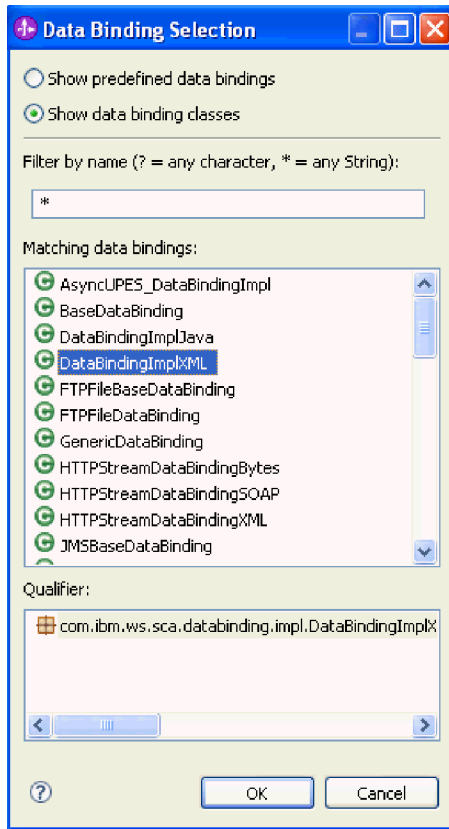


図 30. データ・バインディングの選択

外部サービス・ウィザードが、データ・タイプに応じた正しいデータ・バインディング・クラスをデフォルトとして設定します。データ・バインディングについて詳しくは、本書の **Outbound** データ変換に関するトピックを参照してください。データ・バインディング・クラス名が「構成タイプの選択」ウィンドウに表示されます。



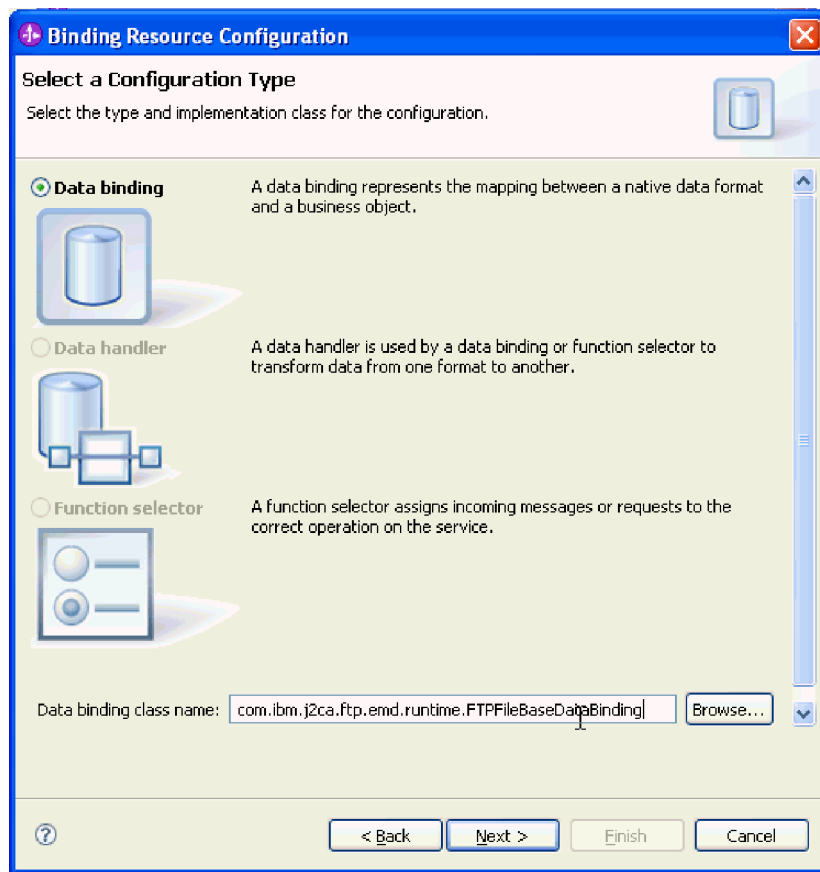


図 31. データ・バインディング・クラスが「構成タイプの選択」ウィンドウに取り込まれます

7. 「次へ」をクリックします。

### 結果

データ・バインディングがモジュールで使用できるように構成されます。

### 次のタスク

現在の外部サービス・ウィザードから続行して、モジュールのデータ・ハンドラーを選択します。

## データ・ハンドラーの構成

ビジネス・オブジェクトを含むデータ・タイプを選択した場合、ビジネス・オブジェクトとネイティブ・フォーマットの間の変換を実行するデータ・ハンドラーを指定する必要があります。

### 始める前に

モジュールにデータ・ハンドラーを指定する前に、データ・バインディングを作成しておく必要があります。

### このタスクを実行する理由および時期

データ・ハンドラーを指定するには、以下の手順を実行します。

注: データ・ハンドラーは、外部サービス・ウィザードを実行する前に、WebSphere Integration Developer を使用して構成できます。これを行うには、WebSphere Integration Developer で「新規」→「リソース構成 (Resource configuration)」を選択して、本書に記載されたデータ・ハンドラー・ウィンドウを完了してください。

#### このタスクの手順

1. 「データ・バインディング・プロパティ」ウィンドウで、「DataHandlerConfigurationName」について「新規」を選択します。
2. 「新規データ・ハンドラー構成」ウィンドウの「名前」フィールドに、データ・ハンドラーの名前を入力します。

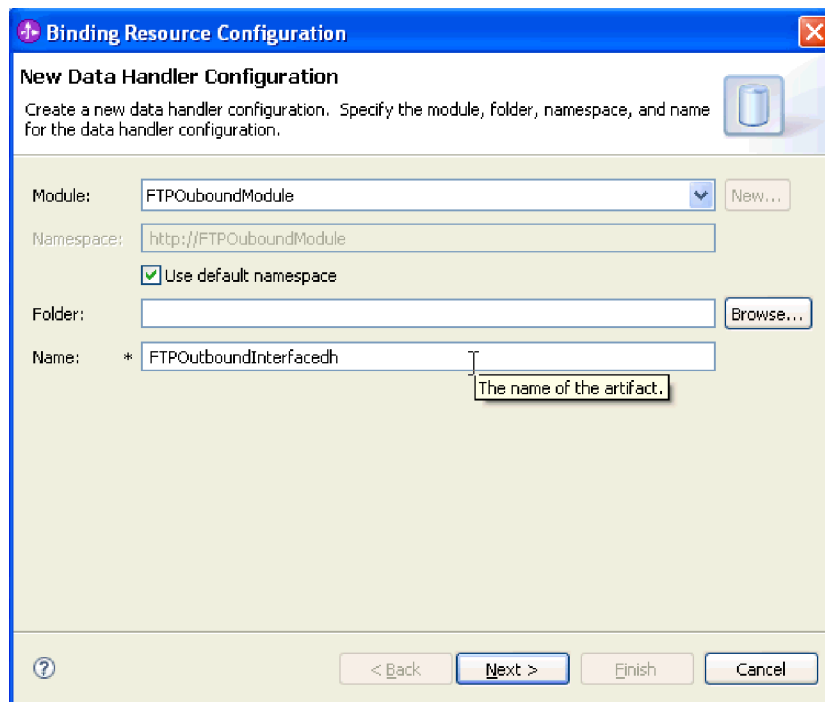


図 32. データ・ハンドラー名の指定

3. 「次へ」をクリックします。
4. 「構成タイプの選択」ウィンドウで、「データ・ハンドラー」ラジオ・ボタンを選択したままにして、「参照」をクリックします。

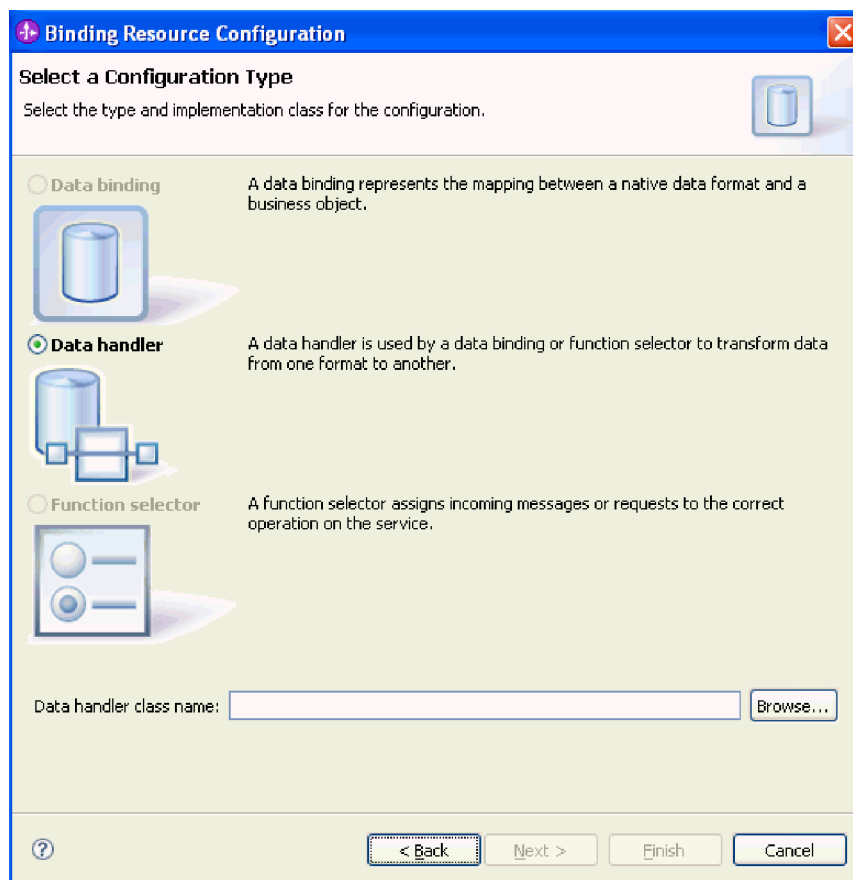


図 33. データ・ハンドラーの構成タイプの選択

5. 「データ・ハンドラー選択」ウィンドウで、ビジネス・オブジェクトが必要とする変換のタイプに対して適切なデータ・ハンドラーを選択し、「OK」をクリックします。

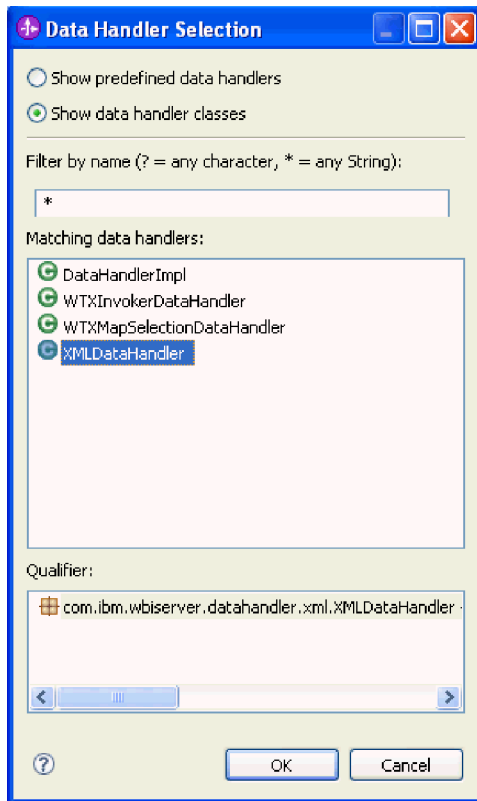


図 34. データ・ハンドラー・クラスの選択

6. 「構成タイプの選択」ウィンドウに、「データ・ハンドラー・クラス名フィールド (Data handler class name field)」が取り込まれます。「次へ」をクリックして先に進みます。
7. 「プロパティの指定」ウィンドウで、「エンコード」フィールドに値を入力して「終了」をクリックします。この値は、アダプターがデータ変換中に使用する文字エンコードのタイプを示します。エンコード・プロパティについては、本書の FTP ビジネス・オブジェクト・プロパティに関するトピックを参照してください。
8. 「データ・バインディング・プロパティ」ウィンドウで「終了」をクリックします。
9. 「操作」ウィンドウで、ウィンドウの「操作出力詳細の指定 (Specify the details of the operation output)」セクションの「DataBinding タイプ」フィールドに対して、「新規」を選択します。

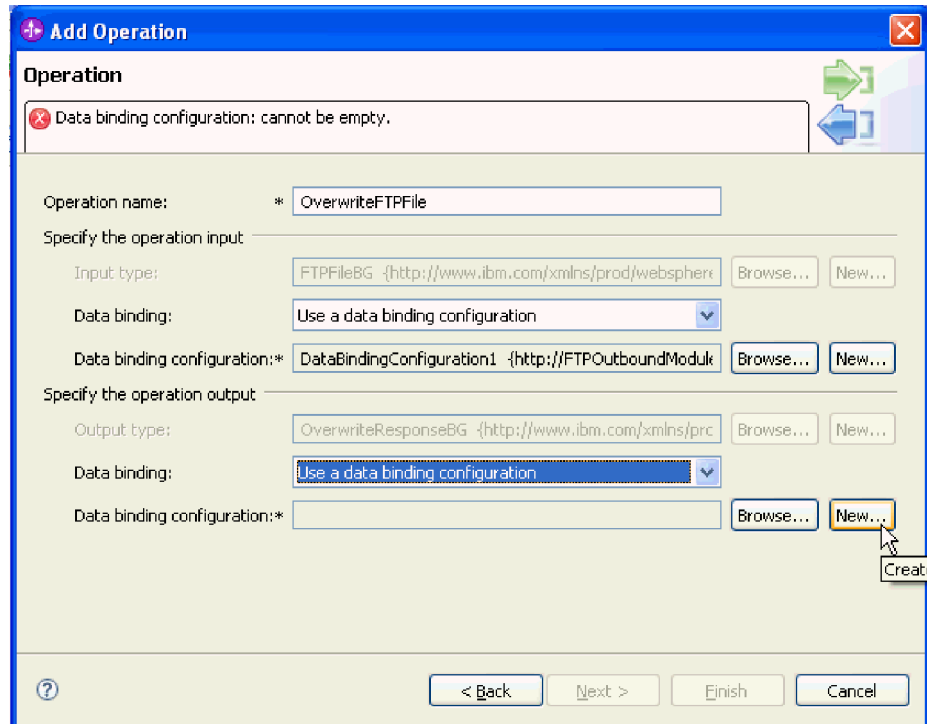


図 35. 「操作の追加」ウィンドウ

- 「名前」フィールドにデータ・バインディングの名前を入力し、「次へ」をクリックします。

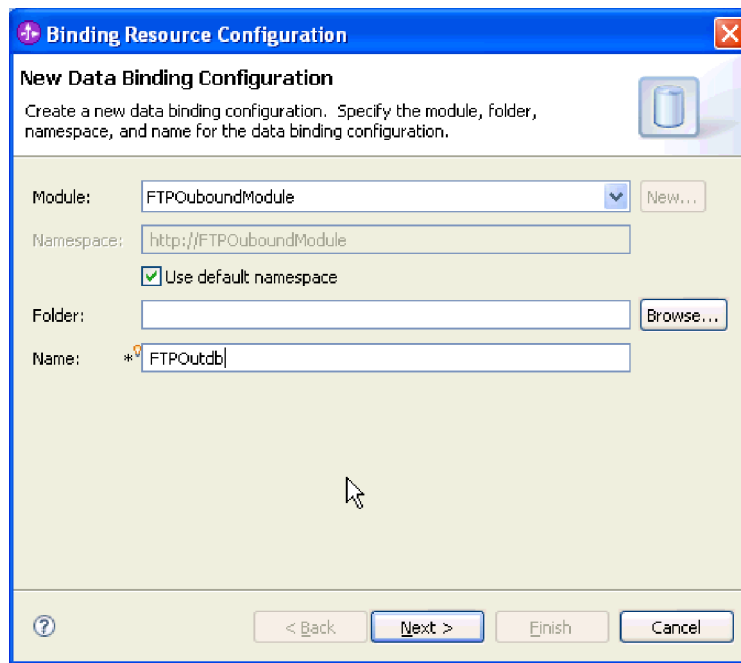


図 36. データ・バインディングの名前の指定

- 「構成タイプの選択」ウィンドウでは、「データ・バインディング」ラジオ・ボタンを選択したままにします。

12. 「終了」をクリックします。
13. 「操作の追加」ウィンドウで「終了」をクリックします。

## 結果

データ・ハンドラーが作成されます。

## 次のタスク

ウィザードを続行して、モジュールの対話仕様プロパティを指定し、成果物を生成します。

## 対話仕様プロパティの設定およびサービスの生成

対話仕様プロパティは、オプションです。設定すると、指定した値が、外部サービス・ウィザードによって生成されるすべての親 FTP ビジネス・オブジェクトのデフォルトとして表示されます。対話仕様プロパティは、操作の対話を制御します。アダプターは、モジュールの成果物を作成するときにインポート・ファイルを生成します。インポート・ファイルには、トップレベル・ビジネス・オブジェクトの操作が含まれます。

### 始める前に

対話仕様プロパティを設定してモジュールの成果物を生成するには、事前にデータ・バインディングを構成し、ビジネス・オブジェクトを選択しておく必要があります。

### このタスクを実行する理由および時期

対話仕様プロパティを設定して成果物を生成するには、以下の手順を実行します。対話仕様プロパティについて詳しくは、本書の該当するトピックを参照してください。

### このタスクの手順

1. オプション: 対話仕様プロパティを設定するには、「操作」ウィンドウにフィールドを取り込みます。「**拡張**」をクリックして、追加のプロパティ詳細を追加することもできます。
  - a. デフォルトとして設定するフィールドすべてに、値を入力します。
  - b. 「**次へ**」をクリックします。

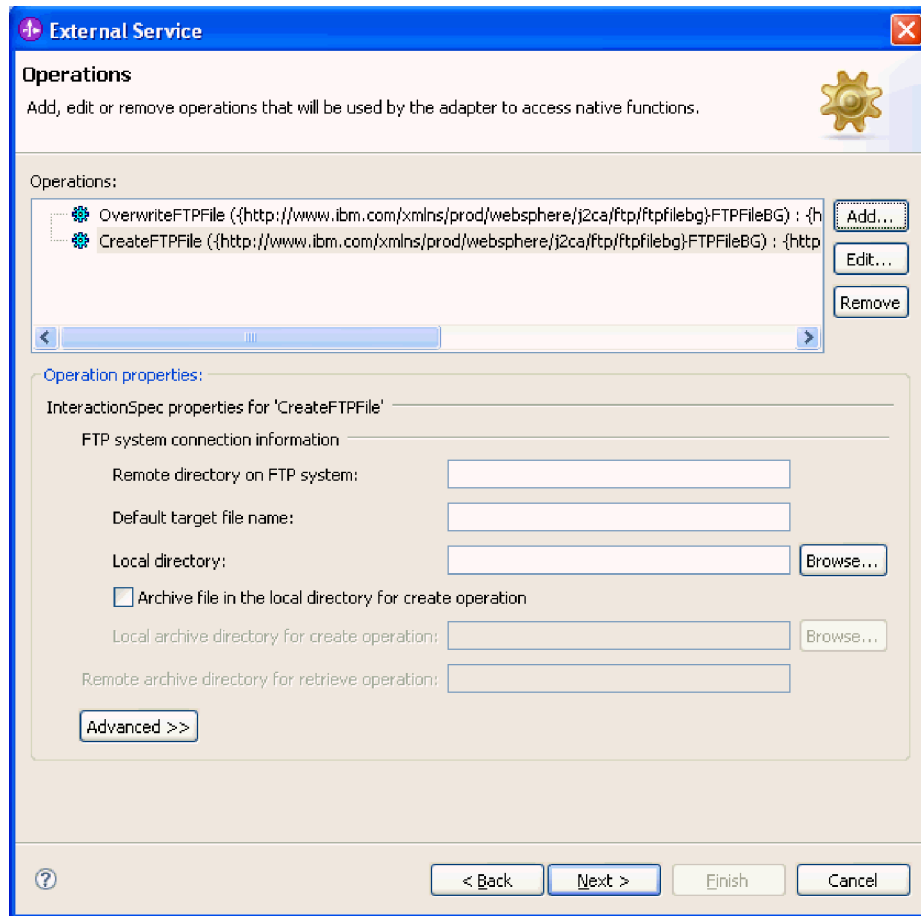


図 37. 対話仕様プロパティ

2. 「サービスの生成」ウィンドウで、インターフェースの名前を指定します。この名前は、WebSphere Integration Developer アセンブリー・ダイアグラムに表示されます。

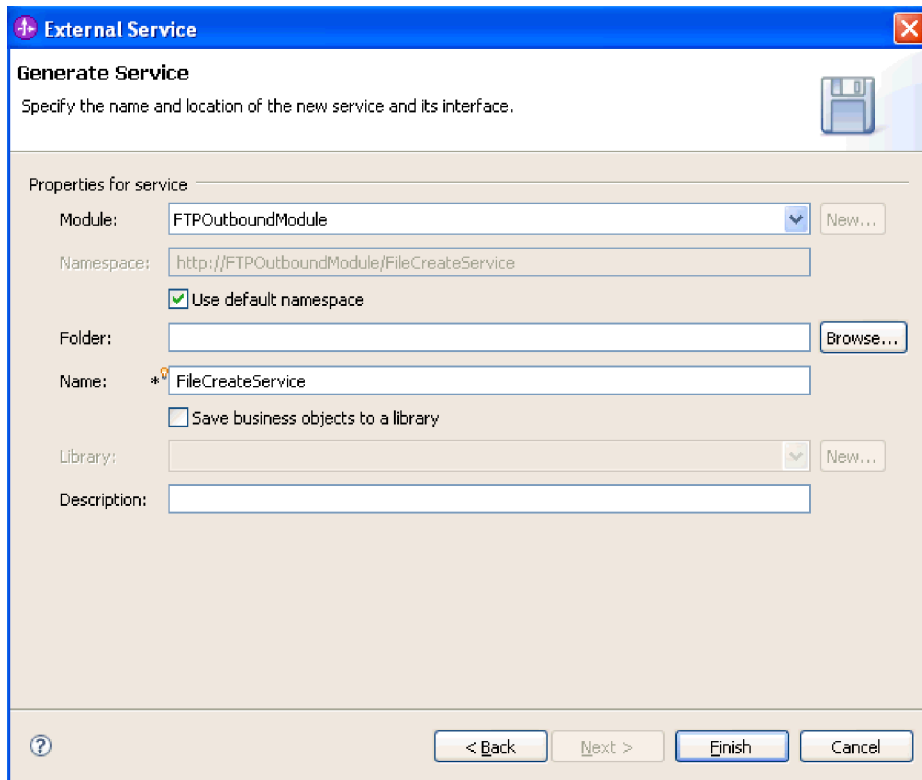


図 38. インターフェースへの命名

3. 「終了」をクリックします。 WebSphere Integration Developer アセンブリー・ダイアグラムが開き、作成したインターフェースが表示されます。



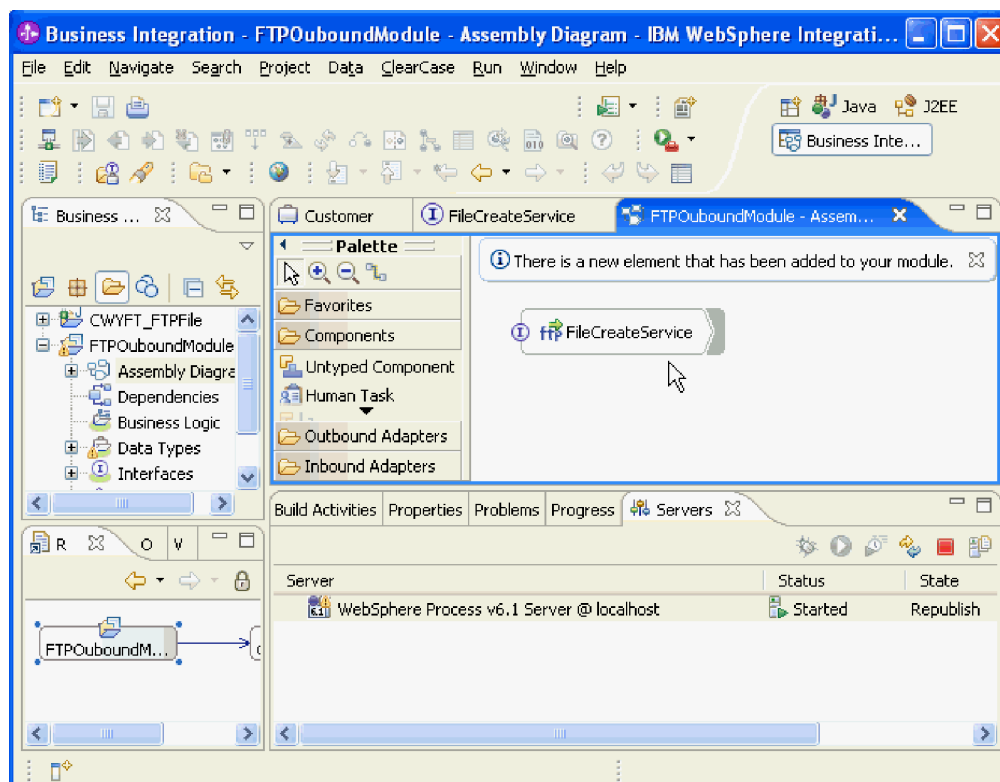


図 39. WebSphere Integration Developer のインターフェース

4. オプション: 上記の手順を繰り返して、バインディング、データ・ハンドラー、対話仕様などの、必要な操作をすべて追加します。

### 結果

WebSphere Integration Developer によって、成果物およびインポートが生成されます。作成された Outbound の成果物は、WebSphere Integration Developer Project Explorer 内のモジュールの下に表示されます。

### 次のタスク

モジュールをサーバーにデプロイします。

## Inbound 処理のモジュールの構成

アダプターを Inbound 処理に使用するようにモジュールを構成するには、WebSphere Integration Developer 内で外部サービス・ウィザードを使用して、ビジネス・サービスを作成し、データ変換処理を指定して、ビジネス・オブジェクト定義および関連する成果物を生成します。

## デプロイメント・プロパティおよびランタイム・プロパティの設定

外部サービス・ウィザードが FTP サーバーの接続に使用するデプロイメント・プロパティとランタイム・プロパティを指定します。

## 始める前に

接続プロパティを指定するには、外部サービス・ウィザードを開始しておく必要があります。

### このタスクを実行する理由および時期

外部サービス・ウィザードでは、ディスカバリーおよびサービス記述作成のためのデータベースへの接続を行うために、この情報が必要となります。

### このタスクの手順

1. 「処理方向」ウィンドウで「**Inbound**」を選択し、「次へ」をクリックします。
2. 「**コネクター・プロジェクトのデプロイ**」フィールドで、モジュールにアダプター・ファイルを組み込むかどうかを指定します。次のいずれかのオプションを選択します。

- 「**単一アプリケーションが使用するモジュールで (With module for use by single application)**」

アダプター・ファイルをモジュール内に組み込むと、モジュールをすべてのアプリケーション・サーバーにデプロイすることができます。単一のモジュールによってアダプターを使用する場合、または複数のモジュールを使ってさまざまなバージョンのアダプターを実行する必要がある場合には、組み込みアダプターを使用します。組み込みアダプターを使用すると、単一のモジュールのアダプターのみをアップグレードすることができるため、その他のモジュールのアダプター・バージョンが変更されて動作が不安定になることはありません。

- 「**複数アプリケーションが使用するサーバー上 (On server for use by multiple applications)**」

モジュール内にアダプター・ファイルを組み込まない場合は、このモジュールを実行させるアプリケーション・サーバーごとに、アダプター・ファイルをスタンドアロン・アダプターとしてインストールする必要があります。複数のモジュールで同じバージョンのアダプターを使用できる場合に、中央の場所でアダプターを管理するには、スタンドアロン・アダプターを使用します。スタンドアロン・アダプターでは、複数のモジュールに対して単一のアダプター・インスタンスを実行することによって、必須リソースを削減することもできます。

3. ご使用のモジュールについて、以下の FTP システム接続情報を定義します。このウィンドウのプロパティについて詳しくは、本書の管理接続ファクトリー・プロパティに関するトピックを参照してください。
  - 「**ホスト名**」 - FTP サーバーのホスト名を指定。
  - 「**リモート・ディレクトリー**」 - アダプターがポーリングしてファイルを取り出す先となる、FTP サーバー上のディレクトリーを指定。
  - 「**ローカル・ディレクトリー**」 - イベント・ファイルを FTP サーバーからダウンロードする先となる、アダプター・ワークステーション上のディレクトリーを指定。
  - 「**ポート番号**」 - FTP サーバーのポート番号を指定。
  - 「**プロトコル**」 - 通常 FTP またはセキュア FTP (FTPS) のいずれかを指定。

「拡張」をクリックし、追加プロパティ（イベント・ポーリングとパーシスタンス、アーカイブ、BiDi フォーマット、およびロギングとトレースを制御するプロパティなど）を指定します。

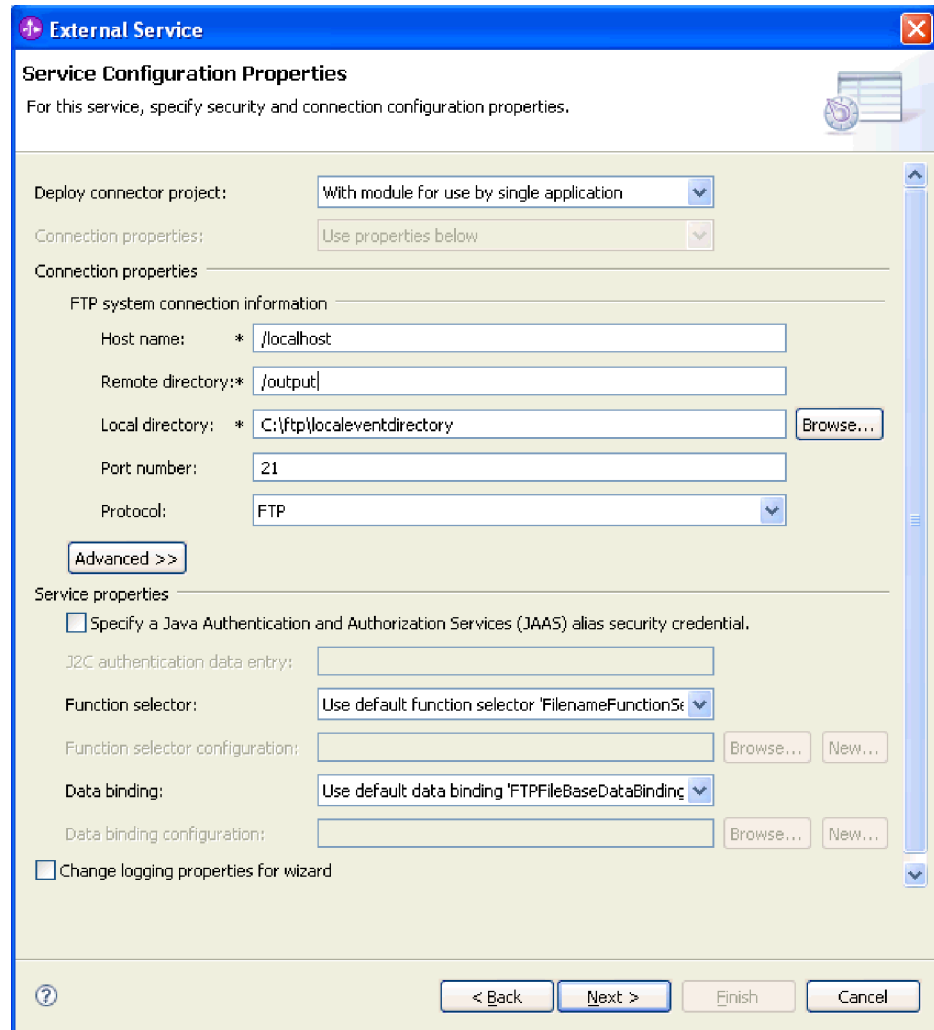


図 40. 「サーバー構成プロパティ (Server Configuration Properties)」 ウィンドウ

4. ウィンドウの「サービス・プロパティ」セクションで、「関数セレクター」フィールドから次のいずれかのオプションを選択します。関数セレクターは、着信したメッセージまたは要求を、サービス内の適切な操作に割り当てます。

- デフォルトの関数セレクター 'FilenameFunctionSelector' を使用

このオプションの使用を選択した場合は、「次へ」をクリックします。

- 関数セレクター構成を使用

このオプションを選択した場合は、以下の手順を実行します。

- a. 「関数セレクター構成」フィールドの横にある「新規」をクリックします。
- b. 「新規関数セレクター構成」ウィンドウで、関数セレクターに「名前」を指定します。「次へ」をクリックします。

注: 外部サービス・ウィザードでは、EIS 関数名は選択できません。デフォルト以外で、アダプターが生成した値を指定したい場合は、アセンブリ・エディターを使用して編集できます。

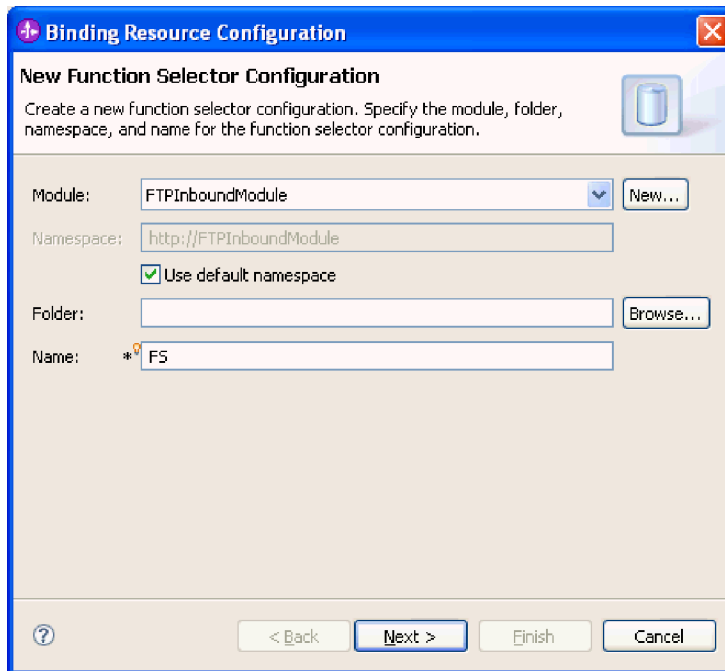


図 41. 「新規関数セレクター構成」ウィンドウ

5. 「構成タイプの選択」ウィンドウから、「関数セレクター・クラス名」フィールドの横にある「参照」をクリックします。
6. 「関数セレクター選択」ウィンドウで、関数セレクターを選択します。「OK」をクリックします。

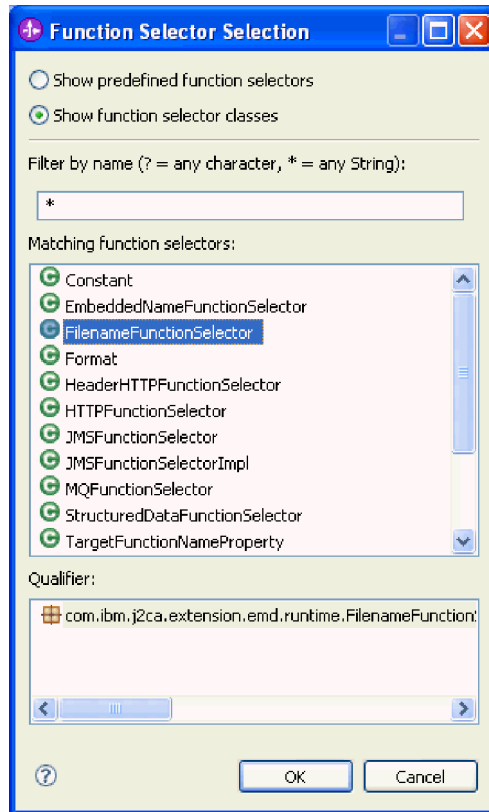


図 42. 「関数セレクター選択」ウィンドウ

7. 「構成タイプの選択」ウィンドウで「次へ」をクリックします。
8. 「関数セレクター・プロパティ」ウィンドウで「終了」をクリックします。
9. 「サービス構成プロパティ」ウィンドウで「次へ」をクリックします。

## 結果

外部サービス・ウィザードは、FTP サーバーへの接続に必要な情報を保有します。

「すべての操作にデフォルトのデータ・バインディング **'FTPFileBaseDataBinding'** を使用」または「操作ごとにデータ・バインディングを指定」のいずれかのデータ・バインディング構成オプションを選択した場合は、「次へ」をクリックして、ウィザードでの作業を続行し、データ・タイプを選択してそのデータ・タイプに関連付けられる操作に名前を付けます。

データ・バインディング構成オプションで「すべての操作にデータ・バインディング構成を使用」を選択した場合は、63 ページの『データ・バインディングの構成』に進んでください。

## データ・タイプおよび操作名の選択

データ・タイプを選択し、データ・タイプに関連付けられる操作に名前を付けるには、外部サービス・ウィザードを使用します。 Inbound 通信の場合、外部サービス・ウィザードでは、ユーザー定義タイプ、汎用 FTP ビジネス・オブジェクト、お

よびビジネス・グラフ付きの汎用 FTP ビジネス・オブジェクトの 3 種類の異なるデータ・タイプから選択することができます。各データ・タイプは、ビジネス・オブジェクト構造に対応しています。

### 始める前に

以下の手順を実行する前に、FTP サーバーにアダプターを接続するための接続プロパティを指定しておく必要があります。

### このタスクを実行する理由および時期

データ・タイプを選択し、それに関連付けられる操作に名前を付けるには、以下の手順を実行します。

### このタスクの手順

1. 「操作」ウィンドウで、「追加」をクリックします。

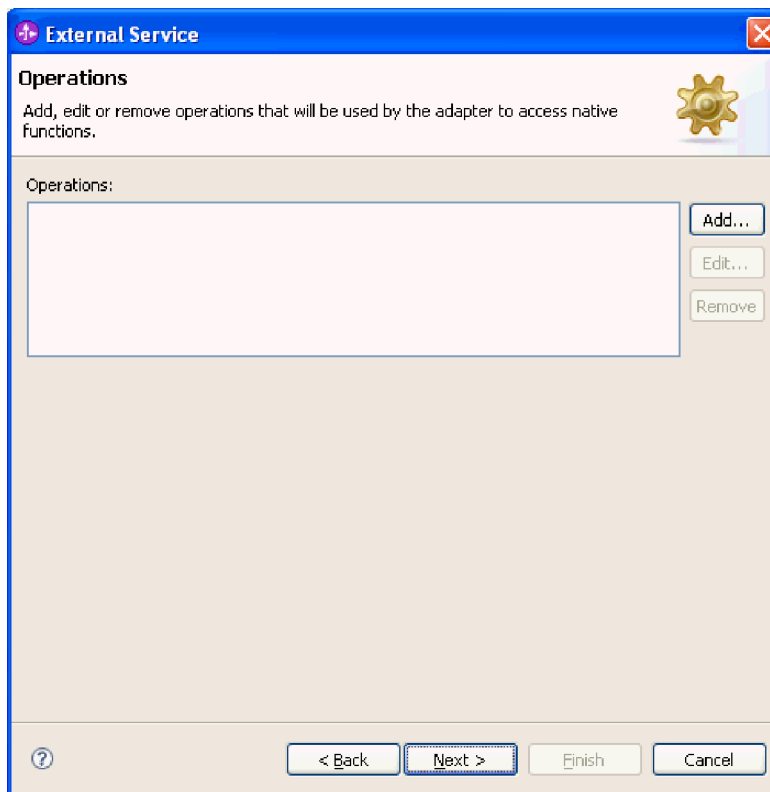


図 43. 操作の追加

2. 「操作の追加」ウィンドウで、「操作入力のデータ型」を選択し、「次へ」をクリックします。「ユーザー定義タイプ」を選択する場合は、そのタイプをサポートするユーザー定義データ・バインディングを指定する必要があります。「汎用 FTP ビジネス・オブジェクト」にデータ・バインディングを指定した場合は、汎用入力タイプのみがサポート対象操作となります。
3. 「操作」ウィンドウで、「操作名」フィールドに名前を入力するか、デフォルトの「emitFTPFile」名のままにしておきます。

注: 名前にスペースを含めることはできません。

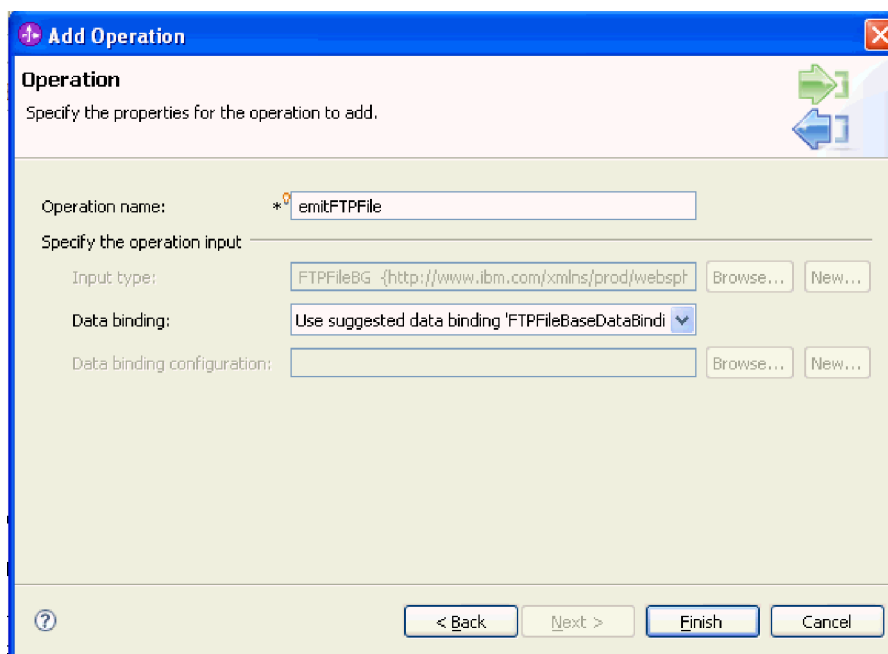


図 44. 操作への名前の指定

## 結果

モジュールのデータ・タイプが定義され、そのデータ・タイプに関連した操作に名前が付けられます。

構成済みのデータ・バインディングを使用する場合は、現在の外部サービス・ウィザードウィンドウで作業を続行し、モジュールで使用するデータ・バインディングを追加して構成します。

デフォルトのデータ・バインディングを使用する場合は、88 ページの『サービスの生成』に進んでください。

## データ・バインディングの構成

各データ・タイプには、それに対応するデータ・バインディングが存在し、ビジネス・オブジェクトのフィールドを読み取って、ファイル内の対応するフィールドを設定するために使用されます。外部サービス・ウィザードで、モジュールにデータ・バインディングを追加し、追加したデータ・バインディングを、使用するデータ・タイプに合うように構成します。このようにして、アダプターはファイル内のフィールドに、ビジネス・オブジェクト内で受け取った情報を取り込む方法を識別します。

### 始める前に

データ・タイプを選択し、そのデータ・タイプに関連付ける操作名を選択しておく必要があります。

### このタスクを実行する理由および時期

モジュール用のデータ・バインディングを追加し、構成するには、以下の手順を実行します。

**注:** データ・バインディングは、外部サービス・ウィザードを実行する前に、WebSphere Integration Developer を使用して構成できます。これを行うには、WebSphere Integration Developer で「新規」 → 「リソース構成 (Resource configuration)」を選択して、本書に記載されたデータ・バインディング・ウィンドウを完了してください。

### このタスクの手順

1. ウィンドウの「サービス・プロパティ」セクションの「データ・バインディング構成」フィールドの横にある「新規」を選択します。

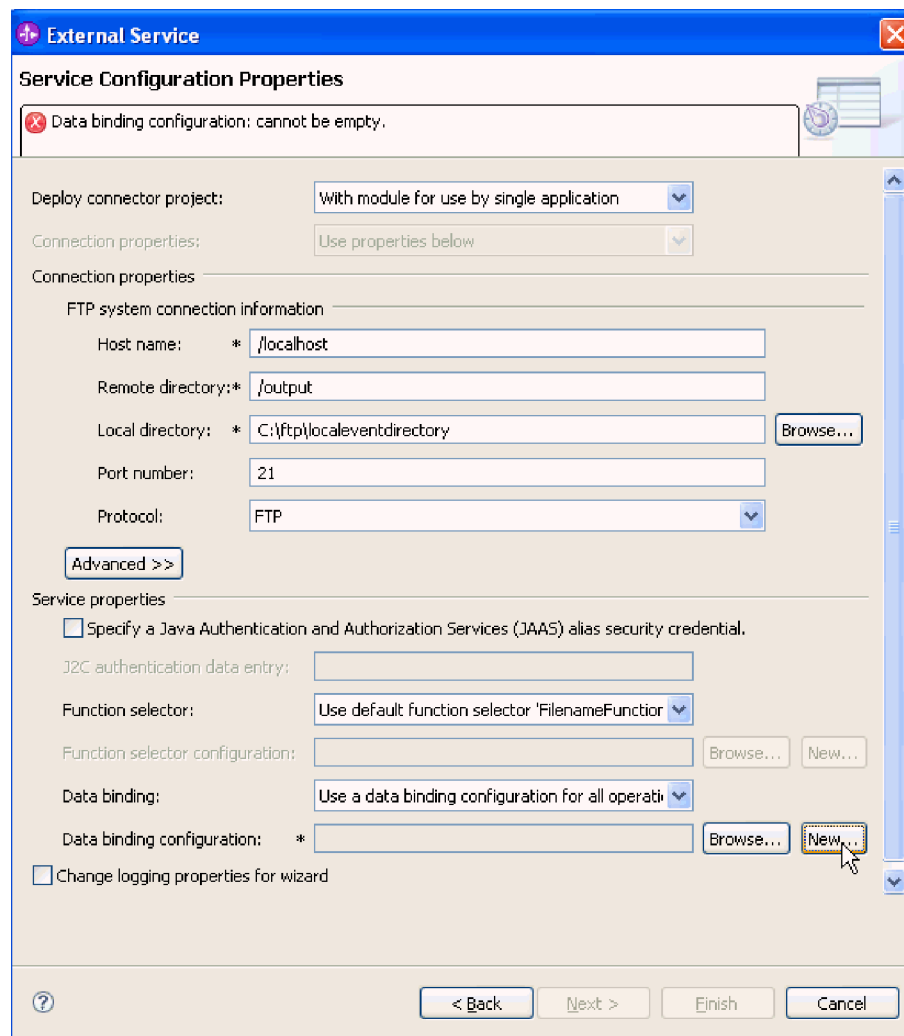


図 45. 「サービス構成プロパティ」ウィンドウ

2. 「名前」フィールドにデータ・バインディングの名前を入力し、「次へ」をクリックします。データ・バインディングはデータ・ハンドラーへのポインターを持つため、その内容を名前に反映させることをお勧めします。例:

FTPI inboundDB\_XML



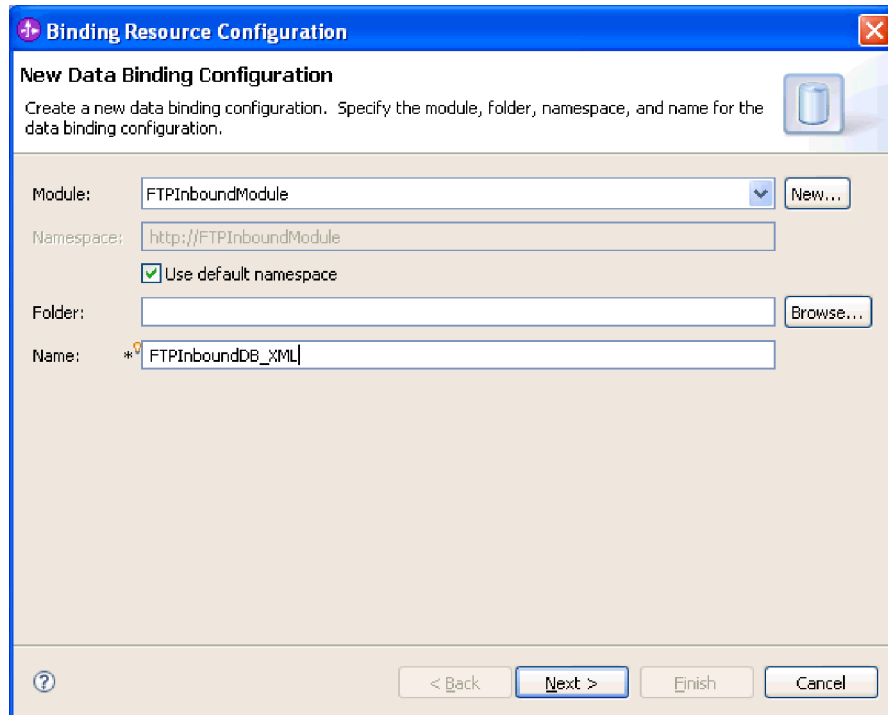


図 46. データ・バインディングの名前の指定

3. 「次へ」をクリックします。
4. 「構成タイプの選択」ウィンドウでは、「データ・バインディング」ラジオ・ボタンを選択した状態にします。
5. 「参照」をクリックして、クラス名を選択します。ここでいう「クラス」とは、このモジュールに作成するデータ・バインディングに関連付けられるデータ・バインディング・クラスのことを指します。
6. 「データ・バインディング選択」ウィンドウで、「データ・バインディング・クラスの表示」ラジオ・ボタンをクリックして選択します。
7. データ・タイプに応じた正しいデータ・バインディング・クラスを選択し、「OK」をクリックします。

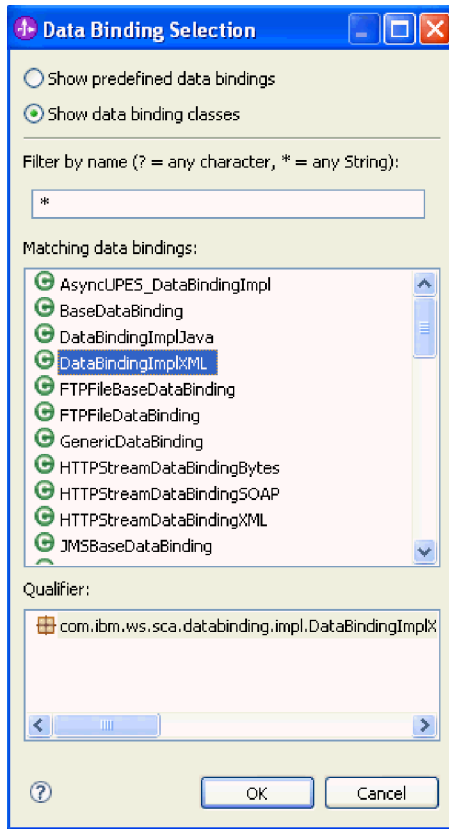


図 47. データ・バインディングの選択

外部サービス・ウィザードが、データ・タイプに応じた正しいデータ・バインディング・クラスをデフォルトとして設定します。データ・バインディングについて詳しくは、本書の **Outbound** データ変換に関するトピックを参照してください。データ・バインディング・クラス名が「構成タイプの選択」ウィンドウに表示されます。

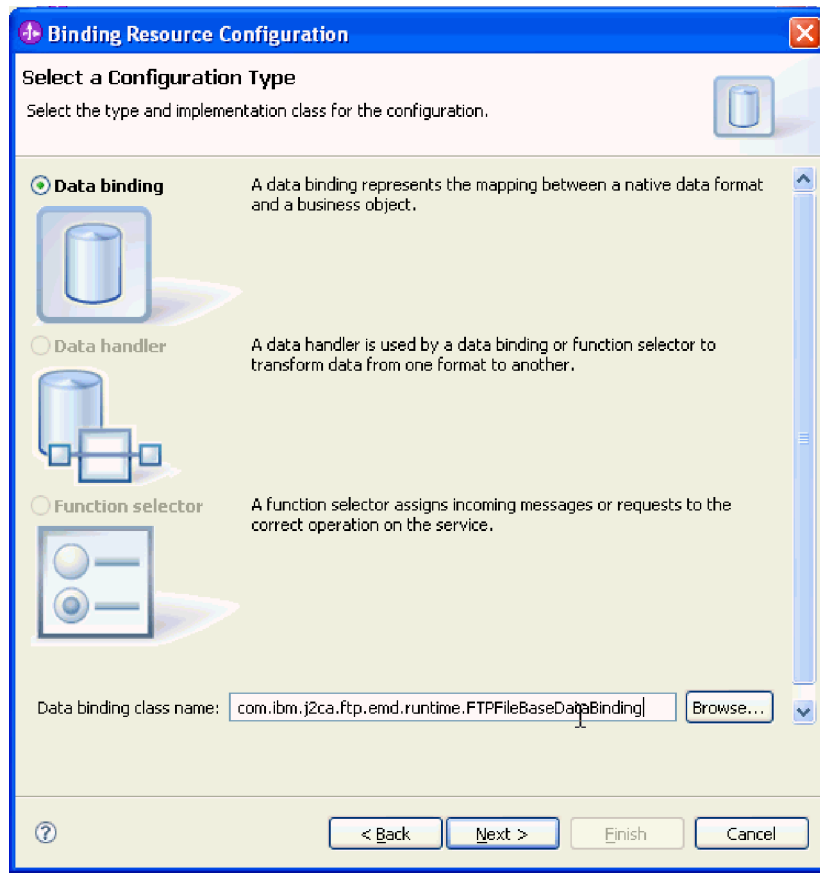


図 48. データ・バインディング・クラスが「構成タイプの選択」ウィンドウに取り込まれます

8. 「次へ」をクリックします。

### 結果

データ・バインディングがモジュールで使用できるように構成されます。

### 次のタスク

現在の外部サービス・ウィザードから続行して、モジュールのデータ・ハンドラーを選択します。

## データ・ハンドラーの構成

ビジネス・オブジェクトを含むデータ・タイプを選択した場合、ビジネス・オブジェクトとネイティブ・フォーマットの間の変換を実行するデータ・ハンドラーを指定する必要があります。

### 始める前に

モジュールにデータ・ハンドラーを指定する前に、データ・バインディングを作成しておく必要があります。

### このタスクを実行する理由および時期

データ・ハンドラーを指定するには、以下の手順を実行します。

注: データ・ハンドラーは、外部サービス・ウィザードを実行する前に、WebSphere Integration Developer を使用して構成できます。これを行うには、WebSphere Integration Developer で「新規」→「リソース構成 (Resource configuration)」を選択して、本書に記載されたデータ・ハンドラー・ウィンドウを完了してください。

#### このタスクの手順

1. 「データ・バインディング・プロパティ」ウィンドウで、「DataHandlerConfigurationName」について「新規」を選択します。
2. 「新規データ・ハンドラー構成」ウィンドウの「名前」フィールドに、データ・ハンドラーの名前を入力します。
3. 「次へ」をクリックします。
4. 「構成タイプの選択」ウィンドウで、「データ・ハンドラー」ラジオ・ボタンを選択した状態で「参照」をクリックします。

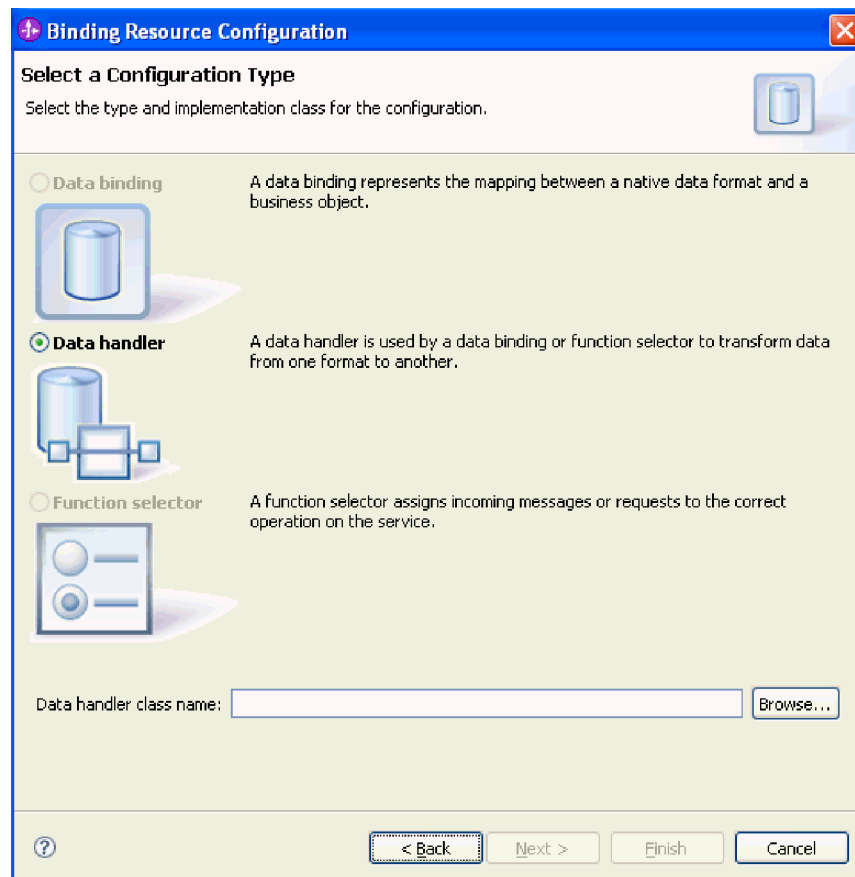


図 49. データ・ハンドラーの構成タイプの選択

5. 「データ・ハンドラー選択」ウィンドウで、ビジネス・オブジェクトが必要とする変換のタイプに対して適切なデータ・ハンドラーを選択し、「OK」をクリックします。

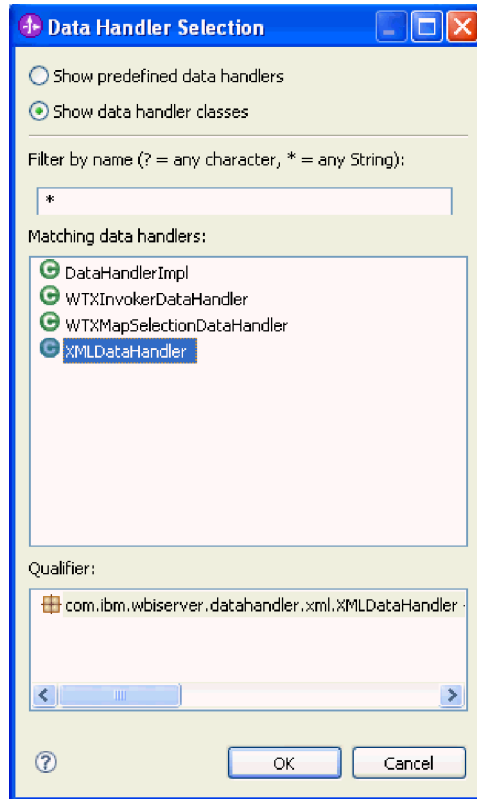


図 50. データ・ハンドラー・クラスを選択

6. 「構成タイプの選択」ウィンドウに、「データ・ハンドラー・クラス名フィールド (Data handler class name field)」が取り込まれます。「次へ」をクリックして先に進みます。
7. 「プロパティの指定」ウィンドウで、「エンコード」フィールドに値を入力して「終了」をクリックします。この値は、アダプターがデータ変換中に使用する文字エンコードのタイプを示します。エンコード・プロパティについて詳しくは、本書の FTP ビジネス・オブジェクト・プロパティに関するトピックを参照してください。
8. 「データ・バインディング・プロパティ」ウィンドウで「終了」をクリックします。
9. 「操作」ウィンドウで、ウィンドウの「操作出力詳細の指定 (Specify the details of the operation output)」セクションの「DataBinding タイプ」フィールドに対して、「新規」を選択します。
10. データ・バインディングの「名前」を入力し、「次へ」をクリックします。

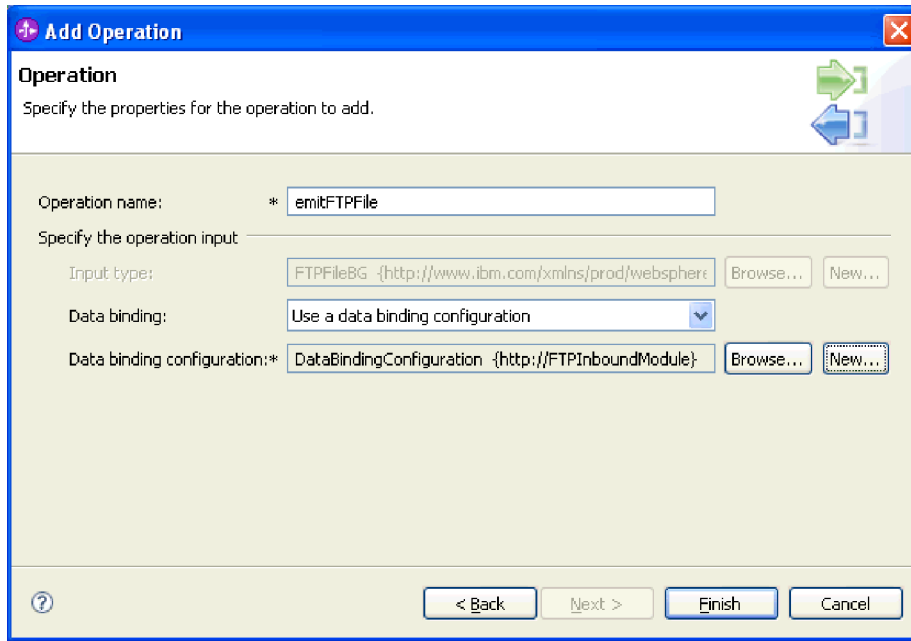


図 51. データ・バインディングの名前の指定

11. 「構成タイプの選択」ウィンドウでは、「データ・バインディング」ラジオ・ボタンを選択した状態にします。
12. 「終了」をクリックします。
13. 「操作」ウィンドウで「終了」をクリックします。

## 結果

データ・ハンドラーが作成されます。

## 次のタスク

ウィザードを続行して、モジュールの対話仕様プロパティを指定し、成果物を生成します。

## サービスの生成

アダプターは、モジュールの成果物を作成するときにエクスポート・ファイルを生成します。エクスポート・ファイルには、トップレベル・ビジネス・オブジェクトの操作が含まれます。

### このタスクを実行する理由および時期

成果物を生成するには、以下の手順を実行します。

### このタスクの手順

1. 「操作」ウィンドウで「次へ」をクリックします。
2. 「サービスの生成」ウィンドウで、インターフェースの名前を指定します。この名前は、WebSphere Integration Developer アセンブリー・ダイアグラムに表示されます。

3. 「終了」をクリックします。WebSphere Integration Developer アセンブリー・ダイアグラムが開き、作成したインターフェースが表示されます。

### 結果

WebSphere Integration Developer によって成果物およびエクスポートが生成されます。作成された Inbound の成果物は、WebSphere Integration Developer Project Explorer 内のモジュールの下に表示されます。

### 次のタスク

モジュールをサーバーにデプロイします。





---

## 第 5 章 アセンブリー・エディターを使用した対話仕様プロパティの変更

サービスの生成後に、ご使用のアダプター・モジュールの対話仕様プロパティを変更するには、WebSphere Integration Developerのアセンブリー・エディターを使用します。

### 始める前に

外部サービス・ウィザードを使用して、アダプター用のサービスをあらかじめ生成しておく必要があります。

### このタスクを実行する理由および時期

アダプター用のサービスを生成した後に、対話仕様プロパティを変更することができます。対話仕様プロパティは、オプションのプロパティで、特定のビジネス・オブジェクトで特定の操作を行うために、メソッド・レベルで設定されます。指定した値は、外部サービス・ウィザードで生成される親ビジネス・オブジェクトすべてに対するデフォルトとして表示されます。EAR ファイルをエクスポートする前であれば、これらのプロパティを変更できます。アプリケーションのデプロイ後に、これらのプロパティを変更することはできません。

対話仕様プロパティを変更するには、以下の手順を実行します。

### このタスクの手順

1. WebSphere Integration Developer の Business Integration パースペクティブから、モジュール名を展開します。
2. 「アセンブリー・ダイアグラム」を展開して、そのインターフェースをダブルクリックします。
3. アセンブリー・エディターのインターフェースをクリックします。(それ以上クリックをしなければ、モジュール・プロパティが表示されます)
4. 「プロパティ」タブをクリックします。(ダイアグラムのインターフェースを右クリックして、「プロパティで表示」をクリックしてもかまいません。)
5. 「バインディング」の下、「メソッド・バインディング」をクリックします。インターフェースのメソッドが、ビジネス・オブジェクトと操作との組み合わせごとに表示されます。
6. 対話仕様プロパティの変更対象となるメソッドを選択します。
7. 「拡張」をクリックして、「汎用 (Generic)」タブでプロパティを変更します。対話仕様プロパティの変更対象となるメソッドごとに、この手順を繰り返します。

### 結果

アダプター・モジュールに関連付けられた対話仕様プロパティが変更されます。

### 次のタスク

モジュールをデプロイします。

---

## 第 6 章 モジュールのデプロイ

モジュールをデプロイして、モジュールとアダプターを構成するファイルを実動またはテスト用の作動環境に配置します。WebSphere Integration Developerの統合テスト環境では、インストール時に選択したテスト環境のプロファイルに応じて、WebSphere Process Server または WebSphere Enterprise Service Bus、あるいは両方のランタイムがサポートされます。

---

### デプロイメント環境

モジュールおよびアダプターをデプロイすることができる、テスト環境および実稼働環境があります。

WebSphere Integration Developer では、テスト環境の 1 つ以上のサーバーにモジュールをデプロイすることができます。ビジネス・インテグレーション・モジュールを実行してテストするには、これがもっとも一般的な方法とされています。ただし、管理コンソールまたはコマンド行ツールを使用して、サーバー・デプロイメント用のモジュールを EAR ファイルとして WebSphere Process Server または WebSphere Enterprise Service Bus にエクスポートすることもできます。

---

### テスト用モジュールのデプロイ

WebSphere Integration Developerでは、組み込むアダプターを持つモジュールをテスト環境にデプロイして、サーバー・ツールで作業することができます。このツールを使うと、サーバー構成の編集、サーバーの始動/停止の切り替え、モジュール・コードのエラー・テストなどといった作業を行うことができます。通常、このテストは、ご使用のコンポーネントのインターフェース操作で実行します。これによって、コンポーネントが適切に実装されているかどうか、適切な参照に配線されているかどうか、などを確認することができます。

### Inbound 処理テスト用のターゲット・コンポーネントの生成および配線

Inbound 処理用のアダプターが組み込まれたモジュールをテスト環境にデプロイする前に、あらかじめターゲット・コンポーネントを生成して配線しておく必要があります。このターゲット・コンポーネントは、アダプターがイベントを送信する先となる宛先として機能します。

#### 始める前に

外部サービス・ウィザード を使用して、エクスポート・モジュールをあらかじめ生成しておく必要があります。

#### このタスクを実行する理由および時期

Inbound 処理用のターゲット・コンポーネントの生成および配線は、テスト環境でのみ必要となります。実稼働環境でアダプターを配置する際には必要ありません。

ターゲット・コンポーネントは、イベントを受信します。WebSphere Integration Developer のアセンブリ・エディターを使用して、エクスポートをターゲット・コンポーネントに「配線」(2つのコンポーネントを接続)します。アダプターは、この配線を使用して、(エクスポートからターゲット・コンポーネントに) イベント・データを受け渡します。

### このタスクの手順

1. ターゲット・コンポーネントを作成します
  - a. WebSphere Integration Developer の Business Integration パースペクティブから、「アセンブリ・ダイアグラム」を展開して、エクスポート・コンポーネントをダブルクリックします。デフォルト値を変更しなかった場合は、エクスポート・コンポーネントの名前は「ご使用のアダプター名 + **InboundInterface**」になります。

インターフェースによって、呼び出し可能な操作と受け渡されるデータ (入力引数、戻り値、例外など) が指定されます。**InboundInterface** は、Inbound 処理をサポートするためにアダプターで必要となる操作を含んでおり、外部サービス・ウィザードの実行時に作成されます。

- b. 新規コンポーネントを作成します。「コンポーネント」を展開し、「型なしコンポーネント」を選択して、コンポーネントをアセンブリ・ダイアグラムにドラッグしてください。

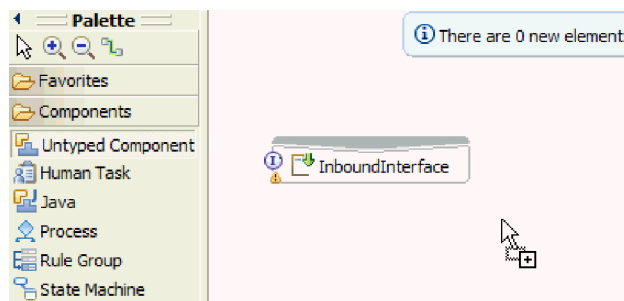


図 52. コンポーネントをアセンブリ・ダイアグラムに追加します

カーソルが配置アイコンに変わります。

- c. コンポーネントをクリックして、アセンブリ・ダイアグラムに表示します。
2. コンポーネントを配線します。
    - a. エクスポート・コンポーネントをクリックして、新規コンポーネントまでドラッグします。これによって、エクスポート・コンポーネントから新規コンポーネントへの配線が作成されます。以下の図を参照してください。

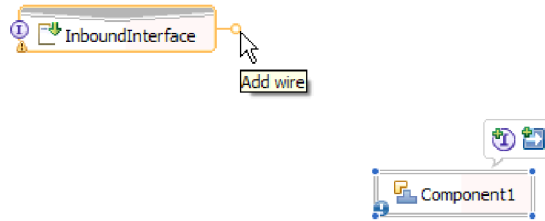


図 53. 配線アイコンを選択します

- b. アセンブリ・ダイアグラムを保存します。「ファイル」→「保管」をクリックします。
3. 新規コンポーネントの実装を生成します。
  - a. 新規コンポーネントを右クリックして、「実装の生成」→「Java」を選択します。

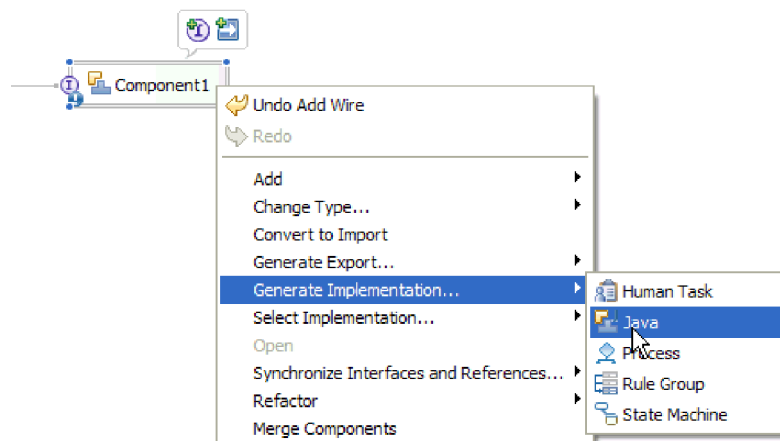


図 54. Java 実装を生成します

- b. 「(デフォルト・パッケージ) ((default package))」を選択し、「OK」をクリックします。これにより、インバウンド・モジュールのエンドポイントが作成されます。

Java 実装は、個別のタブに表示されます。

- c. オプション: エンドポイント・メソッドのそれぞれについて、エンドポイントで受信されたデータ・オブジェクトを出力するために、print ステートメントを追加します。
- d. 「ファイル」→「保管」をクリックして、変更内容を保管します。

### 次のタスク

引き続き、テスト用モジュールのデプロイを行います。

## サーバーへのモジュールの追加

WebSphere Integration Developerでは、テスト環境の 1 つ以上のサーバーにモジュールをデプロイすることができます。

### 始める前に

テスト対象のモジュールがアダプターを使用して Inbound 処理を実行する場合、アダプターがイベントを送信する先となるターゲット・コンポーネントを作成して、配線する必要があります。

### このタスクを実行する理由および時期

モジュールとモジュールでのアダプターの使用内容をテストするには、モジュールをサーバーに追加する必要があります。

### このタスクの手順

1. 条件付き: 「サーバー・ビュー」にサーバーがない場合、以下の手順を実行して、新規サーバーを追加し、定義します。
  - a. 「サーバー・ビュー」にカーソルを置いて右クリックし、「新規」 → 「サーバー」を選択します。
  - b. 「新規サーバーの定義」ウィンドウから、サーバー・タイプを選択します。
  - c. サーバーの設定値を構成します。
  - d. 「終了」をクリックすると、サーバーがパブリッシュされます。
2. モジュールをサーバーに追加します。
  - a. サーバー・ビューに切り替えます。 WebSphere Integration Developer で、「ウィンドウ」 → 「ビューの表示」 → 「サーバー」を選択します
  - a. サーバーを始動します。 WebSphere Integration Developer 画面右下のペインにある「サーバー」タブで、サーバーを右クリックして、「開始」を選択します。
3. サーバー状況が「始動済み」である場合は、サーバーを右クリックして「プロジェクトの追加および除去」を選択します。
4. 「プロジェクトの追加および除去」画面で、ご使用のプロジェクトを選択し、「追加」をクリックします。プロジェクトが、「使用可能プロジェクト」リストから「構成プロジェクト」リストに移動します。
5. 「終了」をクリックします。 これにより、モジュールがサーバーにデプロイされます。

モジュールがサーバーに追加されている間に、右下のペインの「コンソール」タブにログが表示されます。

### 次のタスク

ご使用のモジュールおよびアダプターの機能をテストします。

## テスト・クライアントを使用したモジュールの Outbound 処理テスト

WebSphere Integration Developer の統合テスト・クライアントを使用して、アセンブルされたモジュールとアダプターの Outbound 処理テストを行います。

### 始める前に

最初にモジュールをサーバーに追加する必要があります。

### このタスクを実行する理由および時期

通常、モジュールのテストは、ご使用のコンポーネントのインターフェース操作で実行します。これによって、コンポーネントが適切に実装されているかどうか、適切な参照に配線されているかどうか、などを確認することができます。

### このタスクの手順

1. テストするモジュールを選択し、右クリックして、「テスト」 → 「モジュールのテスト」を選択します。
2. テスト・クライアントを使用したモジュールのテストについて詳しくは、WebSphere Integration Developer インフォメーション・センターのトピック「モジュールおよびコンポーネントのテスト」を参照してください。

### 次のタスク

ご使用のモジュールおよびアダプターのテスト結果に問題がなければ、そのモジュールとアダプターを実稼働環境にデプロイすることができます。

---

## 実動用モジュールのデプロイ

外部サービス・ウィザードを使用して作成されたモジュールを、実稼働環境内の WebSphere Process Server または WebSphere Enterprise Service Bus にデプロイするためのプロセスは、2 つのステップで構成されています。最初に、WebSphere Integration Developer 内にモジュールをエンタープライズ・アーカイブ (EAR) ファイルの形でエクスポートします。次に、WebSphere Process Server 管理コンソールを使用して、EAR ファイルをデプロイします。

## RAR ファイルのインストール (スタンドアロン・アダプターを使用するモジュールの場合のみ)

アダプターをモジュールに組み込まずに、代わりにサーバー・インスタンス内にデプロイしたアプリケーションすべてでアダプターを使用できるようにする場合、アダプターを RAR ファイルの形式でアプリケーション・サーバーにインストールする必要があります。RAR ファイルは、Java archive (JAR) ファイルの一種で、Java 2 Connector (J2C) アーキテクチャーのリソース・アダプターをパッケージ化するために使用されます。

### 始める前に

外部サービス・ウィザードの サービス生成およびデプロイメント構成 (Service Generation and Deployment Configuration) ウィンドウで、あらかじめ「コネクター・

プロジェクトのデプロイを「サーバー上 (複数のアダプターで使用) (On server for use by multiple adapters)」に設定しておく必要があります。

### このタスクを実行する理由および時期

アダプターを RAR ファイルの形式でインストールすると、サーバー・ランタイムで実行される J2EE アプリケーション・コンポーネントすべてで、アダプターが使用可能になります。

### このタスクの手順

1. 管理コンソールを開始します。
2. 「リソース」 → 「リソース・アダプター」 → 「リソース・アダプター」とクリックします。
3. 「リソース・アダプター」 ページで、「RAR のインストール」をクリックします。

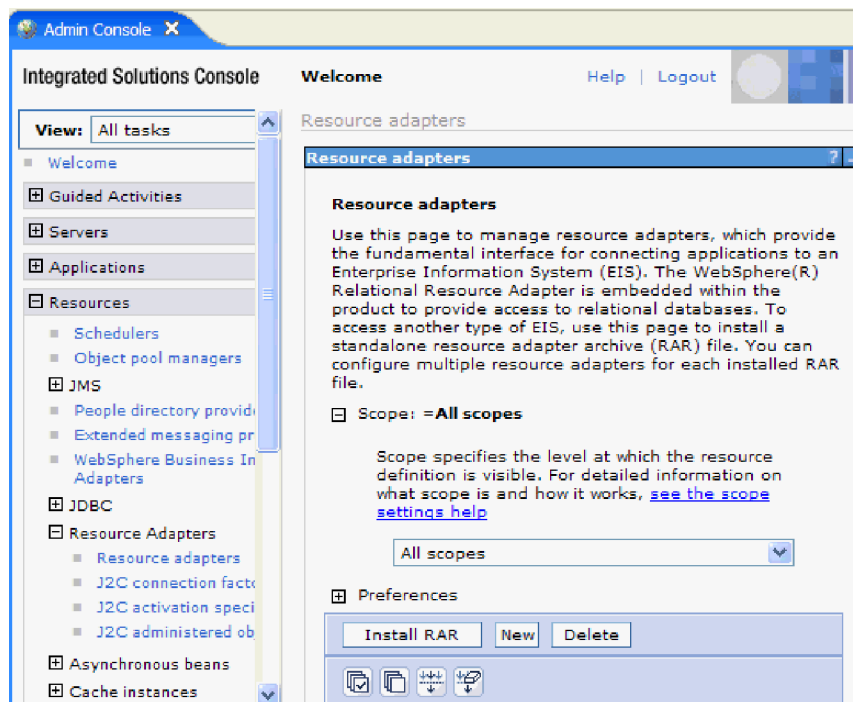


図 55. 「リソース・アダプター」 ページの「RAR のインストール」 ボタン

4. 「RAR ファイルのインストール」 ページで「参照」をクリックして、ご使用のアダプター用の RAR ファイルにナビゲートします。

通常、RAR ファイルは、次のパスにインストールされます。

`WID_installation_directory/ResourceAdapters/adapter_name/deploy/adapter.rar`

5. 「次へ」をクリックします。
6. 「リソース・アダプター」 ページで、必要に応じてアダプターの名前を変更し、説明を追加します。
7. 「OK」をクリックします。
8. ページの上部にある「メッセージ」ボックスで「保管」をクリックします。



## 次のタスク

次のステップは、サーバーにデプロイ可能な EAR ファイルとして、モジュールをエクスポートすることになります。

## EAR ファイルとしてのモジュールのエクスポート

WebSphere Integration Developer を使用して、モジュールを EAR ファイルとしてエクスポートします。EAR ファイルを作成することによって、モジュールのすべての内容を WebSphere Process Server または WebSphere Enterprise Service Bus に容易にデプロイできる形式で取り込みます。

### 始める前に

モジュールを EAR ファイルとしてエクスポートするには、事前にサービスと通信するためのモジュールを作成しておく必要があります。このモジュールを、WebSphere Integration Developer ビジネス・インテグレーション・パースペクティブ内に表示する必要があります。

### このタスクを実行する理由および時期

モジュールを EAR ファイルとしてエクスポートするには、以下の手順を実行します。

### このタスクの手順

1. モジュールを右クリックして、「**エクスポート**」を選択します。
2. 「選択」ウィンドウで、「**J2EE**」を展開します。
3. 「**EAR ファイル**」を選択して、「**次へ**」をクリックします。

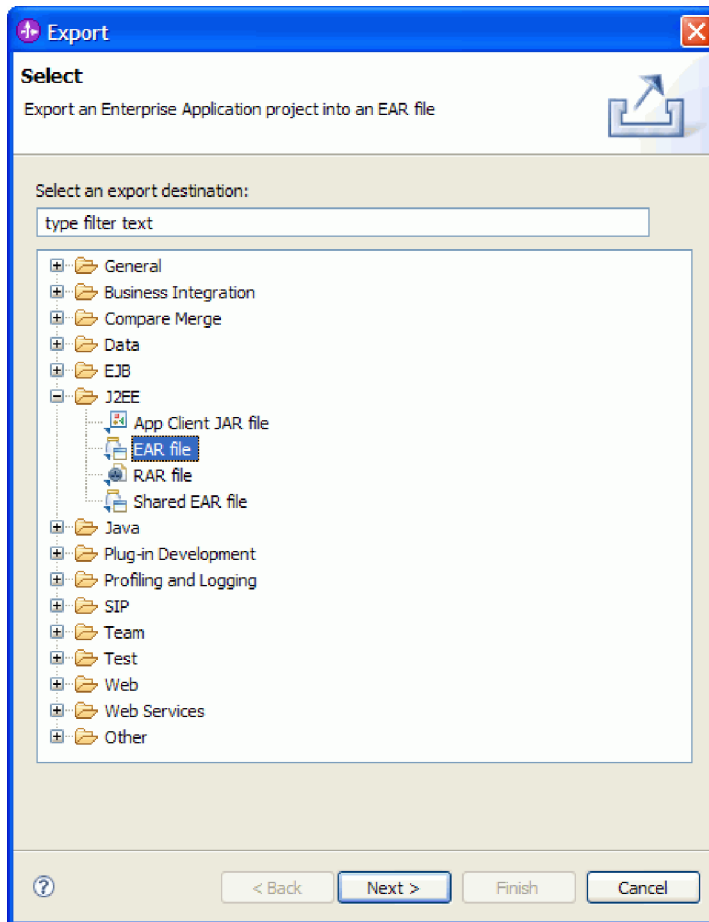


図 56. 「選択」ウィンドウから「EAR ファイル」を選択します

4. オプション: 正しい EAR アプリケーションを選択します。EAR アプリケーションには、ご使用のモジュールにちなんだ名前が付けられますが、名前の最後に「App」が追加されます。
5. EAR ファイルを格納するローカル・ファイル・システム上で、「参照」を選択してフォルダーを参照します。
6. オプションで「ソース・ファイルのエクスポート」を選択できます (ソース・ファイルのエクスポートする場合)。EAR ファイルに加えてソース・ファイルもエクスポートするという場合に備えて、このオプションが提供されています。ソース・ファイルには、Java コンポーネント、データ・マップなどに関連付けられたファイルが含まれています。
7. 既存のファイルを上書きする場合は、「既存のファイルを上書き (Overwrite an existing file)」をクリックします。
8. 「終了」をクリックします。

## 結果

モジュールの内容が EAR ファイルとしてエクスポートされます。

## 次のタスク

このモジュールを管理コンソールにインストールします。これにより、モジュールが WebSphere Process Server にデプロイされます。

## EAR ファイルのインストール

EAR ファイルのインストールは、デプロイメント・プロセスの最後のステップです。EAR ファイルをサーバーにインストールして実行すると、EAR ファイルの一部として組み込まれているアダプターが、インストールされたアプリケーションの一部として実行されます。

### 始める前に

WebSphere Process Server 上にモジュールをインストールする前に、モジュールを EAR ファイルとしてエクスポートしておく必要があります。

### このタスクを実行する理由および時期

EAR ファイルをインストールするには、以下の手順を実行します。アダプター・モジュール・アプリケーションのクラスター化について詳しくは、<http://www.ibm.com/software/webservers/appserv/was/library/>を参照してください。

### このタスクの手順

1. サーバー・インスタンスを右クリックし、「管理コンソールの実行」を選択して、WebSphere Process Server 管理コンソールを開きます。
2. 管理コンソール・ウィンドウで、「アプリケーション (Applications)」 → 「新規アプリケーションのインストール (Install New Applications)」をクリックします。



図 57. 「アプリケーション・インストールの準備」ウィンドウ

3. 「参照」をクリックして、EAR ファイルを位置指定し、「次へ」をクリックします。EAR ファイル名は、モジュール名の後に「App」が付いたものです。

4. オプション: クラスター化された環境にデプロイする場合は、以下の手順を実行します。
  - a. 「**ステップ 2: サーバーにモジュールをマップ**」ウィンドウで、モジュールを選択します。
  - b. サーバー・クラスターの名前を選択します。
  - c. 「**適用**」をクリックします。
5. 「**次へ**」をクリックして、「**要約**」を開きます。すべての設定が正しいことを確認して、「**終了**」をクリックします。
6. オプション: 認証別名を使用している場合は、以下の手順を実行します。
  - a. 「**セキュリティ**」を展開して、「**ビジネス・インテグレーションの認証別名 (Business Integration Authentication Aliases)**」を選択します。
  - b. 構成する認証別名を選択します。 認証別名の構成を変更するための管理者権限またはオペレーター権限を持っている必要があります。
  - c. オプション: 「**ユーザー名**」を入力します (まだ入力されていない場合)。
  - d. 「**パスワード**」を入力します (まだ入力されていない場合)。
  - e. 「**確認パスワード (Confirm Password)**」フィールドに再度パスワードを入力します (まだ入力されていない場合)。
  - f. 「**OK**」をクリックします。

## 結果

この時点で、プロジェクトがデプロイメントされ、「**エンタープライズ・アプリケーション**」ウィンドウが表示されます。

## 次のタスク

いずれかのプロパティを設定または再設定する場合、あるいは、アダプター・プロジェクトのアプリケーションをクラスター化したい場合は、トラブルシューティング・ツールを構成する前に、管理コンソールを使用して対応する変更を行ってください。

---

## 第 7 章 アダプター・モジュールの管理

アダプターをスタンドアロンのデプロイメントで稼働している場合は、アダプター・モジュールの開始、停止、モニター、およびトラブルシューティングには、サーバーの管理コンソールを使用します。組み込みアダプターを使用しているアプリケーションでは、アプリケーションの開始時または停止時にアダプター・モジュールが開始または停止します。

---

### 組み込みアダプターの構成プロパティーの変更

モジュールの一部としてアダプターをデプロイした後に構成プロパティーを変更するには、ランタイム環境の管理コンソールを使用します。リソース・アダプター・プロパティー (一般的なアダプター操作で使用)、管理接続ファクトリー・プロパティー (Outbound 処理で使用)、およびアクティベーション・スペック・プロパティー (Inbound 処理で使用) を更新することができます。

### 組み込みアダプターのリソース・アダプター・プロパティーの設定

アダプターをモジュールの一部としてデプロイした後に、そのアダプターのリソース・アダプター・プロパティーを設定するには、管理コンソールを使用します。構成するプロパティーの名前を選択してから、値の変更または設定を行います。

#### 始める前に

ご使用のアダプター・モジュールは、WebSphere Process Server または WebSphere Enterprise Service Bus にデプロイする必要があります。

#### このタスクを実行する理由および時期

カスタム・プロパティーは、すべての WebSphere アダプターが共用する、デフォルトの構成プロパティーです。

管理コンソールを使用してプロパティーを構成するには、次の手順を使用してください。

#### このタスクの手順

1. 管理コンソールを開始します。
2. 「アプリケーション」の下で「エンタープライズ・アプリケーション」を選択します。
3. 「エンタープライズ・アプリケーション」リストから、プロパティーの変更対象となるアダプター・モジュールの名前をクリックします。
4. 「モジュール」の下で、「モジュールの管理」をクリックします。

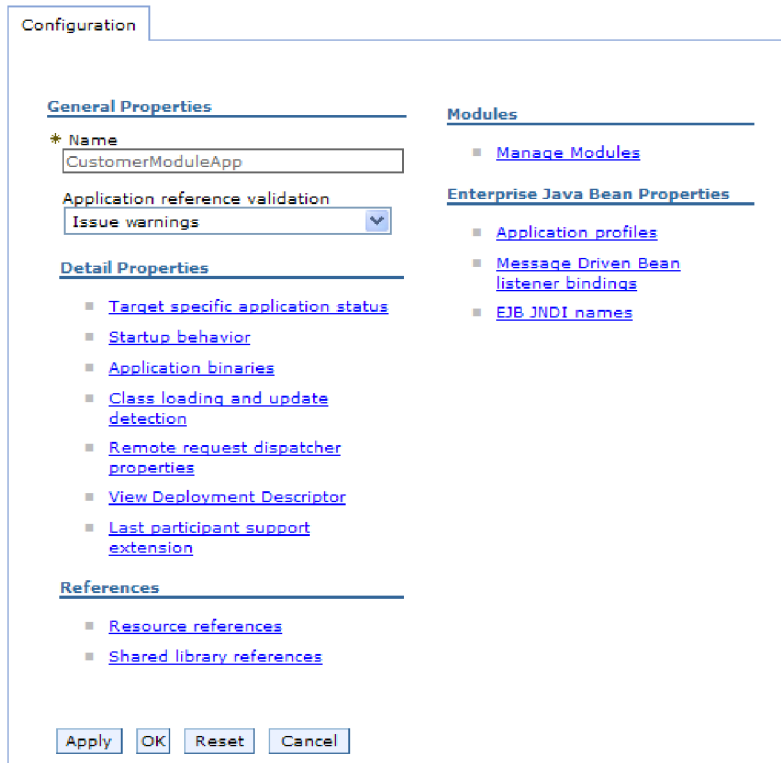


図 58. 「構成」タブ内での「モジュールの管理」の選択

5. 「IBM WebSphere Adapter for FTP」をクリックします。
6. 「追加プロパティ」リストから、「リソース・アダプター」をクリックします。
7. 次のページの「追加プロパティ」リストから、「カスタム・プロパティ」をクリックします。
8. 変更する各プロパティごとに、以下の手順を実行します。

**注:** ここで示すプロパティについて詳しくは、142 ページの『リソース・アダプター・プロパティ』を参照してください。

- a. プロパティの名前をクリックします。
- b. 「値」フィールドの内容を変更します。フィールドが空の場合は、値を入力してください。

例えば、「logNumberOfFiles」をクリックすると、以下のページが表示されます。

Configuration

**General Properties**

\* Scope  
widNode

Required

Name  
logNumberOfFiles

Value  
1

Description

Type  
java.lang.String

Apply OK Reset Cancel

図 59. logNumberOfFiles プロパティの「構成」タブ

「値」フィールドの数値を変更したり、プロパティの説明を追加したりすることができます。

- c. 「OK」をクリックします。
9. ウィンドウの上部にある「メッセージ」ボックスの「保管」リンクをクリックします。

### 結果

アダプター・モジュールに関連付けられたリソース・アダプター・プロパティが変更されます。

## 組み込みアダプターの管理 (J2C) 接続ファクトリー・プロパティの設定

アダプターをモジュールの一部としてデプロイした後に、そのアダプターの管理接続ファクトリー・プロパティを設定するには、管理コンソールを使用します。構成するプロパティの名前を選択してから、値の変更または設定を行います。

### 始める前に

ご使用のアダプター・モジュールは、WebSphere Process Server または WebSphere Enterprise Service Bus にデプロイする必要があります。

### このタスクを実行する理由および時期

管理接続ファクトリー・プロパティは、ターゲット・FTP サーバーのインスタンスを構成する場合に使用します。

注: 管理コンソール内では、このプロパティを「J2C 接続ファクトリー・プロパティ」と呼びます。

管理コンソールを使用してプロパティを構成するには、次の手順を使用してください。

#### このタスクの手順

1. 管理コンソールを開始します。
2. 「アプリケーション」の下で「エンタープライズ・アプリケーション」を選択します。
3. 「エンタープライズ・アプリケーション」リストから、プロパティの変更対象となるアダプター・モジュールの名前をクリックします。
4. 「モジュール」の下で、「モジュールの管理」をクリックします。

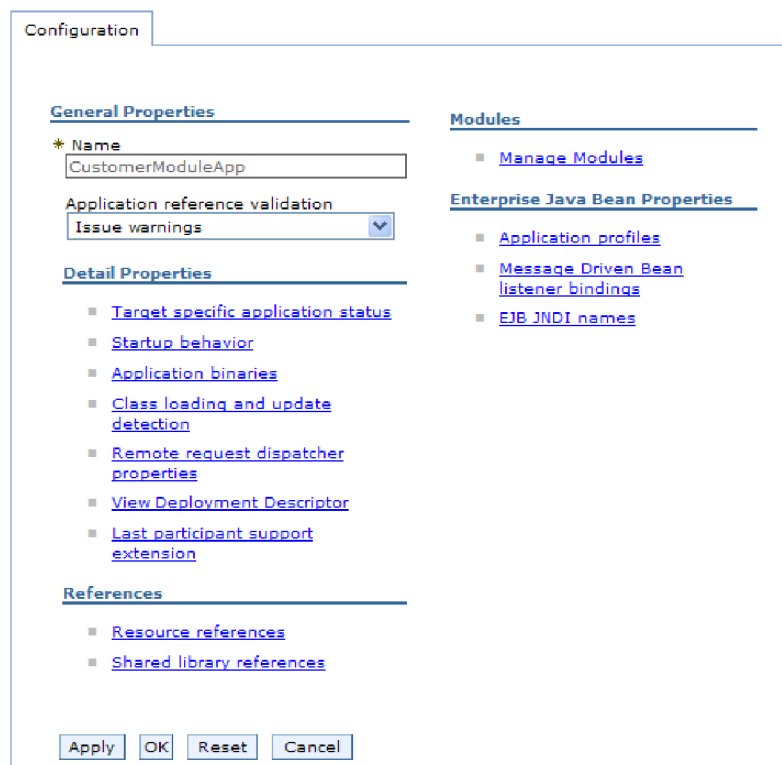


図 60. 「構成」タブ内での「モジュールの管理」の選択

5. 「IBM WebSphere Adapter for FTP」をクリックします。
6. 「追加プロパティ」リストから、「リソース・アダプター」をクリックします。
7. 次のページの「追加プロパティ」リストから、「J2C 接続ファクトリー」をクリックします。
8. アダプター・モジュールに関連付けられた接続ファクトリーの名前をクリックします。
9. 「追加プロパティ」リストから、「カスタム・プロパティ」をクリックします。



カスタム・プロパティは、Adapter for FTP に固有の J2C 接続ファクトリー・プロパティです。接続プールと拡張接続ファクトリーのプロパティは、独自のアダプターを開発する場合に構成するプロパティです。

10. 変更する各プロパティごとに、以下の手順を実行します。

注: ここで示すプロパティについて詳しくは、145 ページの『管理 (J2C) 接続ファクトリー・プロパティ』を参照してください。

- a. プロパティの名前をクリックします。
  - b. 「値」フィールドの内容を変更します。フィールドが空の場合は、値を入力してください。
  - c. 「OK」をクリックします。
11. ウィンドウの上部にある「メッセージ」ボックスの「保管」リンクをクリックします。

### 結果

アダプター・モジュールに関連付けられた管理接続ファクトリー・プロパティが変更されます。

## 組み込みアダプターのアクティベーション・スペック・プロパティの設定

アダプターをモジュールの一部としてデプロイした後に、そのアダプターのアクティベーション・スペック・プロパティを設定するには、管理コンソールを使用します。構成するメッセージ・エンドポイント・プロパティの名前を選択してから、値の変更または設定を行います。

### 始める前に

ご使用のアダプター・モジュールは、WebSphere Process Server または WebSphere Enterprise Service Bus にデプロイする必要があります。

### このタスクを実行する理由および時期

アクティベーション・スペック・プロパティは、エンドポイントを Inbound 処理用に構成する場合に使用します。

管理コンソールを使用してプロパティを構成するには、次の手順を使用してください。

### このタスクの手順

1. 管理コンソールを開始します。
2. 「アプリケーション」の下で「エンタープライズ・アプリケーション」を選択します。
3. 「エンタープライズ・アプリケーション」リストから、プロパティの変更対象となるアダプター・モジュールの名前をクリックします。
4. 「モジュール」の下で、「モジュールの管理」をクリックします。

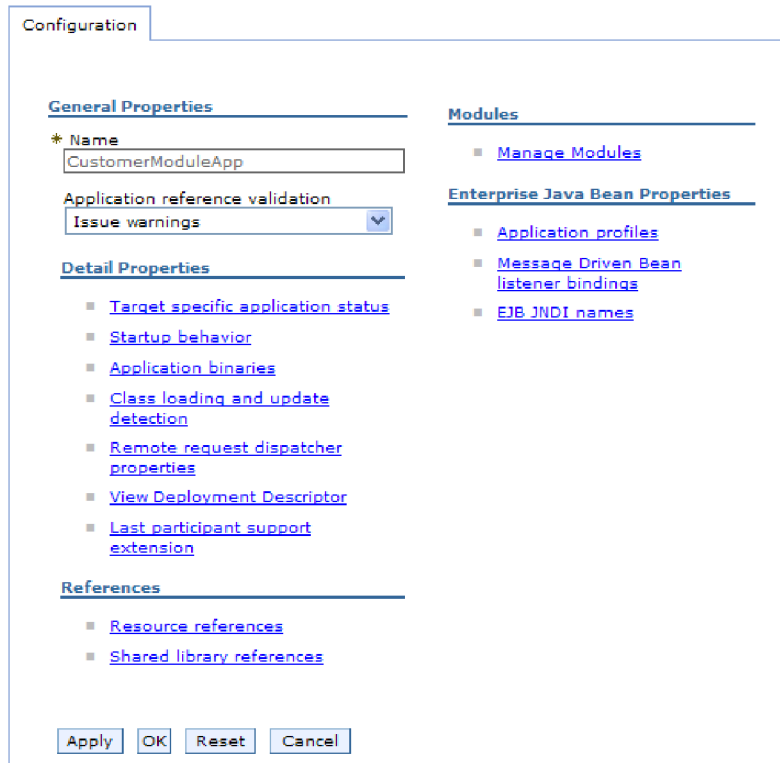


図 61. 「構成」タブ内での「モジュールの管理」の選択

5. 「**IBM WebSphere Adapter for FTP**」をクリックします。
6. 「追加プロパティ」リストから、「リソース・アダプター」をクリックします。
7. 次のページの「追加プロパティ」リストから、「**J2C アクティベーション・スペック**」をクリックします。
8. アダプター・モジュールに関連付けられ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。
9. 「追加プロパティ」リストから、「**J2C アクティベーション・スペック・カスタム・プロパティ**」をクリックします。
10. 変更する各プロパティごとに、以下の手順を実行します。

注: ここで示すプロパティについて詳しくは、171 ページの『アクティベーション・スペック・プロパティ』を参照してください。

- a. プロパティの名前をクリックします。
  - b. 「値」フィールドの内容を変更します。フィールドが空の場合は、値を入力してください。
  - c. 「**OK**」をクリックします。
11. ウィンドウの上部にある「メッセージ」ボックスの「保管」リンクをクリックします。

## 結果

アダプター・モジュールに関連付けられたアクティベーション・スペック・プロパティが変更されます。

---

## スタンドアロン・アダプターの構成プロパティーの変更

スタンドアロン・アダプターをインストールした後に構成プロパティーを設定するには、ランタイム環境の管理コンソールを使用します。アダプターに関する一般情報を提供した後に、リソース・アダプター・プロパティー (一般的なアダプター操作で使用されます) を設定します。アダプターが Outbound 操作で使用される場合は、接続ファクトリーを作成してから、そのプロパティーを設定します。アダプターが Inbound 操作で使用される場合は、アクティベーション・スペックを作成してから、そのプロパティーを設定します。

## スタンドアロン・アダプターのリソース・アダプター・プロパティーの設定

スタンドアロン・アダプターを WebSphere Process Server または WebSphere Enterprise Service Bus にインストールした後に、そのアダプターのリソース・アダプター・プロパティーを設定するには、管理コンソールを使用します。構成するプロパティーの名前を選択してから、値の変更または設定を行います。

### 始める前に

WebSphere Process Server または WebSphere Enterprise Service Bus にアダプターをインストールしておく必要があります。

### このタスクを実行する理由および時期

カスタム・プロパティーは、すべての WebSphere アダプターが共用する、デフォルトの構成プロパティーです。

管理コンソールを使用してプロパティーを構成するには、次の手順を使用してください。

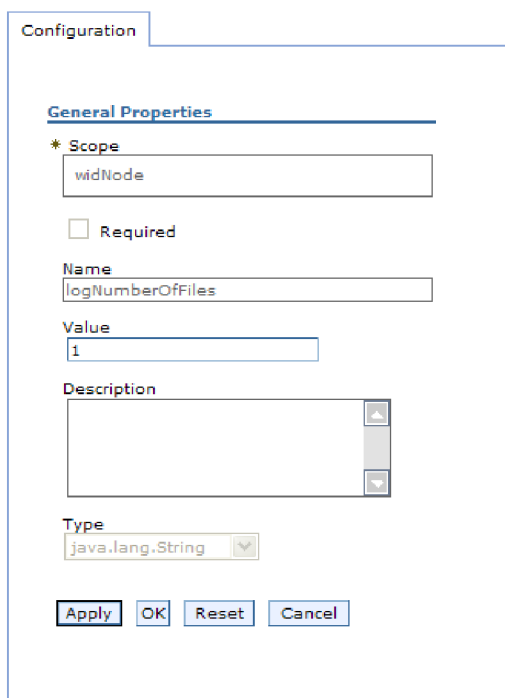
### このタスクの手順

1. 管理コンソールを開始します。
2. 「リソース」 → 「リソース・アダプター」 → 「リソース・アダプター」とクリックします。
3. 「リソース・アダプター」 ページで、「**IBM WebSphere Adapter for FTP**」をクリックします。
4. 「追加プロパティー」リストから、「**カスタム・プロパティー**」をクリックします。
5. 変更する各プロパティーごとに、以下の手順を実行します。

**注:** ここで示すプロパティーについて詳しくは、142 ページの『リソース・アダプター・プロパティー』を参照してください。

- a. プロパティーの名前をクリックします。
- b. 「値」フィールドの内容を変更します。フィールドが空の場合は、値を入力してください。

例えば、「logNumberOfFiles」をクリックすると、以下のページが表示されます。



The screenshot shows a configuration window titled "Configuration" with a sub-tab "General Properties". The "Scope" field is a text box containing "widNode". Below it is an unchecked checkbox labeled "Required". The "Name" field is a text box containing "logNumberOfFiles". The "Value" field is a text box containing "1". The "Description" field is a large empty text area with scrollbars. The "Type" field is a dropdown menu showing "java.lang.String". At the bottom, there are four buttons: "Apply", "OK", "Reset", and "Cancel".

図 62. logNumberOfFiles プロパティの「構成」タブ

「値」フィールドの数値を変更したり、プロパティの説明を追加したりすることができます。

- c. 「OK」をクリックします。
6. ページの上部にある「メッセージ」ボックスで「保管」をクリックします。

### 結果

アダプターに関連付けられたリソース・アダプター・プロパティが変更されます。

## スタンドアロン・アダプターの管理 (J2C) 接続ファクトリー・プロパティの設定

スタンドアロン・アダプターを WebSphere Process Server または WebSphere Enterprise Service Bus にインストールした後に、そのアダプターの管理接続ファクトリー・プロパティを設定するには、管理コンソールを使用します。構成するプロパティの名前を選択してから、値の変更または設定を行います。

### 始める前に

WebSphere Process Server または WebSphere Enterprise Service Bus にアダプターをインストールしておく必要があります。

### このタスクを実行する理由および時期

管理接続ファクトリー・プロパティは、ターゲット・FTP サーバーのインスタンスを構成する場合に使用します。

**注:** 管理コンソール内では、このプロパティを「J2C 接続ファクトリー・プロパティ」と呼びます。

管理コンソールを使用してプロパティを構成するには、次の手順を使用してください。

#### このタスクの手順

1. 管理コンソールを開始します。
2. 「リソース」 → 「リソース・アダプター」 → 「リソース・アダプター」とクリックします。
3. 「リソース・アダプター」ページで、「**IBM WebSphere Adapter for FTP**」をクリックします。
4. 「追加プロパティ」リストから、「**J2C 接続ファクトリー**」をクリックします。
5. 既存の接続ファクトリーを使用する場合は、このステップは飛ばして、ステップ 6 に進んでください。

**注:** 外部サービス・ウィザードを使用してアダプター・モジュールを構成した際に、「事前定義された接続プロパティを使用する」を選択している場合は、接続ファクトリーを作成する必要はありません。

接続ファクトリーを作成する場合は、以下の手順を実行します。

- a. 「**新規 (New)**」をクリックします。
- b. 「構成」タブの「**一般プロパティ**」セクションで、接続ファクトリーの名前を入力します。例えば、AdapterCF と入力します。
- c. 「**JNDI 名**」の値を入力します。例えば、com/eis/AdapterCF と入力します。
- d. 「**コンポーネント管理認証別名**」リストから、認証別名を選択します。
- e. 「**OK**」をクリックします。
- f. ページの上部にある「**メッセージ**」ボックスで「**保管**」をクリックします。

新規に作成された接続ファクトリーが表示されます。

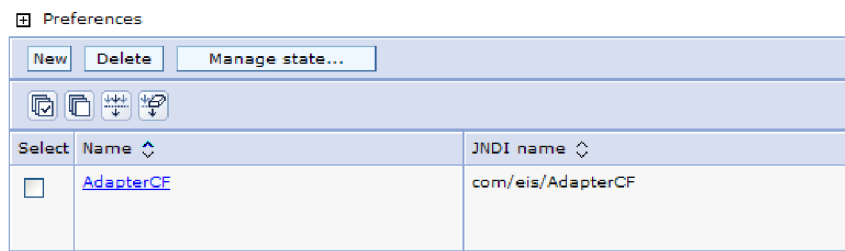


図 63. 接続ファクトリーのリスト

6. 接続ファクトリーのリストから、使用するものをクリックします。

7. 「追加プロパティ」リストから、「カスタム・プロパティ」をクリックします。

カスタム・プロパティは、Adapter for FTP に固有の J2C 接続ファクトリー・プロパティです。接続プールと拡張接続ファクトリーのプロパティは、独自のアダプターを開発する場合に構成するプロパティです。

8. 変更する各プロパティごとに、以下の手順を実行します。

注: ここで示すプロパティについて詳しくは、145 ページの『管理 (J2C) 接続ファクトリー・プロパティ』を参照してください。

- a. プロパティの名前をクリックします。
  - b. 「値」フィールドの内容を変更します。フィールドが空の場合は、値を入力してください。
  - c. 「OK」をクリックします。
9. プロパティの設定が完了したら、「適用」をクリックします。
  10. ウィンドウの上部にある「メッセージ (Messages)」ボックスで「保管」をクリックします。

## 結果

アダプターに関連付けられた管理接続ファクトリー・プロパティが設定されます。

## スタンドアロン・アダプターのアクティベーション・スペック・プロパティの設定

スタンドアロン・アダプターを WebSphere Process Server または WebSphere Enterprise Service Bus にインストールした後に、そのアダプターのアクティベーション・スペック・プロパティを設定するには、管理コンソールを使用します。構成するメッセージ・エンドポイント・プロパティの名前を選択してから、値の変更または設定を行います。

### 始める前に

WebSphere Process Server または WebSphere Enterprise Service Bus にアダプターをインストールしておく必要があります。

### このタスクを実行する理由および時期

アクティベーション・スペック・プロパティは、エンドポイントを Inbound 処理用に構成する場合に使用します。

管理コンソールを使用してプロパティを構成するには、次の手順を使用してください。

### このタスクの手順

1. 管理コンソールを開始します。
2. 「リソース」 → 「リソース・アダプター」 → 「リソース・アダプター」とクリックします。

3. 「リソース・アダプター」ページで、「**IBM WebSphere Adapter for FTP**」をクリックします。
4. 「**追加プロパティ**」リストから、「**J2C アクティベーション・スペック**」をクリックします。
5. 既存のアクティベーション・スペックを使用する場合は、このステップは飛ばして、ステップ 6 に進んでください。

**注:** 外部サービス・ウィザードを使用してアダプター・モジュールを構成した際に、「**事前定義された接続プロパティを使用する**」を選択している場合は、アクティベーション・スペックを作成する必要はありません。

アクティベーション・スペックを作成する場合は、以下の手順を実行します。

- a. 「**新規 (New)**」をクリックします。
- b. 「**構成**」タブの「**一般プロパティ**」セクションで、アクティベーション・スペックの名前を入力します。例えば、AdapterAS と入力します。
- c. 「**JNDI 名**」の値を入力します。例えば、com/eis/AdapterAS と入力します。
- d. 「**認証別名**」リストから、認証別名を選択します。
- e. **メッセージ・リスナー・タイプ**を選択します。
- f. 「**OK**」をクリックします。
- g. ページの上部にある「**メッセージ**」ボックスで「**保管**」をクリックします。

新規に作成されたアクティベーション・スペックが表示されます。

6. アクティベーション・スペックのリストから、使用するものをクリックします。
7. 「**追加プロパティ**」リストから、「**J2C アクティベーション・スペック・カスタム・プロパティ**」をクリックします。
8. 設定するプロパティごとに、以下の手順を実行します。

**注:** ここで示すプロパティについて詳しくは、171 ページの『アクティベーション・スペック・プロパティ』を参照してください。

- a. プロパティの名前をクリックします。
  - b. 「**値**」フィールドの内容を変更します。フィールドが空の場合は、値を入力してください。
  - c. 「**OK**」をクリックします。
9. プロパティの設定が完了したら、「**適用**」をクリックします。
  10. ページの上部にある「**メッセージ**」ボックスで「**保管**」をクリックします。

## 結果

アダプターに関連付けられたアクティベーション・スペック・プロパティが設定されます。

---

## アダプターを使用するアプリケーションの開始

アダプターを使用するアプリケーションを開始するには、サーバーの管理コンソールを使用します。デフォルトでは、サーバーの始動時に、アプリケーションが自動的に開始されます。

### このタスクを実行する理由および時期

組み込みアダプターを使用している場合でも、スタンドアロン・アダプターを使用している場合でも、アプリケーションを開始するにはこの手順を使用します。組み込みアダプターを使用するアプリケーションでは、アプリケーションの開始時にアダプターが開始されます。スタンドアロン・アダプターを使用するアプリケーションでは、アプリケーション・サーバーの始動時にアダプターが開始されます。

### このタスクの手順

1. 管理コンソールで、「アプリケーション」 → 「エンタープライズ・アプリケーション」とクリックします。

注: 管理コンソールには「Integrated Solutions Console」というラベルが付いています。

2. 開始するアプリケーションのチェック・ボックスを選択します。このアプリケーション名は、インストールされている EAR ファイルの名前から、ファイル拡張子 .EAR を除いたものになっています。
3. 「開始 (Start)」をクリックします。

### 結果

アプリケーションの状況が「開始済み」に変化し、アプリケーションが開始したことを示すメッセージが、管理コンソールの上部に表示されます。

---

## アダプターを使用するアプリケーションの停止

アダプターを使用するアプリケーションを停止するには、サーバーの管理コンソールを使用します。デフォルトでは、サーバーの停止時に、アプリケーションも自動的に停止します。

### このタスクを実行する理由および時期

組み込みアダプターを使用している場合でも、スタンドアロン・アダプターを使用している場合でも、アプリケーションを停止するにはこの手順を使用します。組み込みアダプターを使用するアプリケーションでは、アプリケーションの停止時にアダプターが停止します。スタンドアロン・アダプターを使用するアプリケーションでは、アプリケーション・サーバーの停止時にアダプターが停止します。

### このタスクの手順

1. 管理コンソールで、「アプリケーション」 → 「エンタープライズ・アプリケーション」とクリックします。

注: 管理コンソールには「Integrated Solutions Console」というラベルが付いています。



2. 停止するアプリケーションのチェック・ボックスを選択します。このアプリケーション名は、インストールされている EAR ファイルの名前から、ファイル拡張子 .EAR を除いたものになっています。
3. 「停止 (Stop)」をクリックします。

### 結果

アプリケーションの状況が「停止中」に変化し、アプリケーションが停止したことを示すメッセージが、管理コンソールの上部に表示されます。

---

## Performance Monitoring Infrastructure を使用したパフォーマンスのモニター

Performance Monitoring Infrastructure (PMI) は、管理コンソールの機能の 1 つで、これを使用すると、実稼働環境内で Adapter for FTP を含む、コンポーネントのパフォーマンスを動的にモニターすることができます。PMI は、サーバー内のさまざまなコンポーネントから、平均応答時間や要求の総数などのアダプターのパフォーマンス・データを収集して、そのデータをツリー構造に編成します。このデータは、Tivoli® Performance Viewer (WebSphere Process Server の管理コンソールに統合されているグラフィカル・モニター・ツール) で表示することができます。

### このタスクを実行する理由および時期

PMI により、以下の時点のデータを収集することによって、アダプターのパフォーマンスをモニターすることができます。

- Outbound 処理時。Outbound 要求をモニターします。
- Inbound イベントの取り出し時。イベント・テーブルからのイベントの取り出しをモニターします。
- Inbound イベントの送達時。エンドポイント (1 つまたは複数の) へのイベントの送達をモニターします。

使用するアダプター用に PMI を使用可能に設定し、構成するためには、まず、トレース機能の詳細レベルを設定し、パフォーマンス・データの収集元となるいくつかのイベントを実行する必要があります。

ご使用のアダプター環境の全体的なパフォーマンスをモニターして改善するために、PMI を活用する方法について詳しく知るには、WebSphere Application Server の Web サイト (<http://www.ibm.com/software/webservers/appserv/was/library/>) で、PMI を検索してください。

## Performance Monitoring Infrastructure の構成

Performance Monitoring Infrastructure (PMI) を、アダプターのパフォーマンス・データ (平均応答時間や要求の総数など) を収集するように構成することができます。ご使用のアダプター用に PMI を構成した後に、Tivoli Performance Viewer を使用してアダプターのパフォーマンスをモニターすることができます。

### 始める前に

使用するアダプター用に PMI を構成するためには、まず、トレース機能の詳細レベルを設定し、パフォーマンス・データの収集元となるいくつかのイベントを実行する必要があります。

1. トレース機能を使用可能にしてイベント・データを受け取るためには、トレース・レベルを `fine`、`finer`、`finest`、または `all` のいずれかに設定する必要があります。`*=info` の後に、コロンとストリングを追加します。例えば、次のように入力します。

```
*=info: WBILocationMonitor.CEI.ResourceAdapter.  
*=finest: WBILocationMonitor.LOG.ResourceAdapter.*=finest:
```

トレース・レベルの設定方法については、118 ページの『Common Event Infrastructure (CEI) によるトレースの使用可能化』を参照してください。

2. 1 つ以上の Outbound 要求 または Inbound イベントを生成して、構成可能なパフォーマンス・データを生成します。

### このタスクの手順

1. アダプターに対して PMI を使用可能にします。
  - a. 管理コンソールで「**モニターおよびチューニング**」を展開してから、「**Performance Monitoring Infrastructure (PMI)**」を選択します。
  - b. サーバーのリストから、ご使用のサーバーの名前をクリックします。
  - c. 「**構成**」タブを選択してから、「**Performance Monitoring (PMI) を使用可能にする (Enable Performance Monitoring (PMI))**」チェック・ボックスを選択します。
  - d. 「**カスタム**」を選択して、選択的に統計を使用可能または使用不可に設定します。

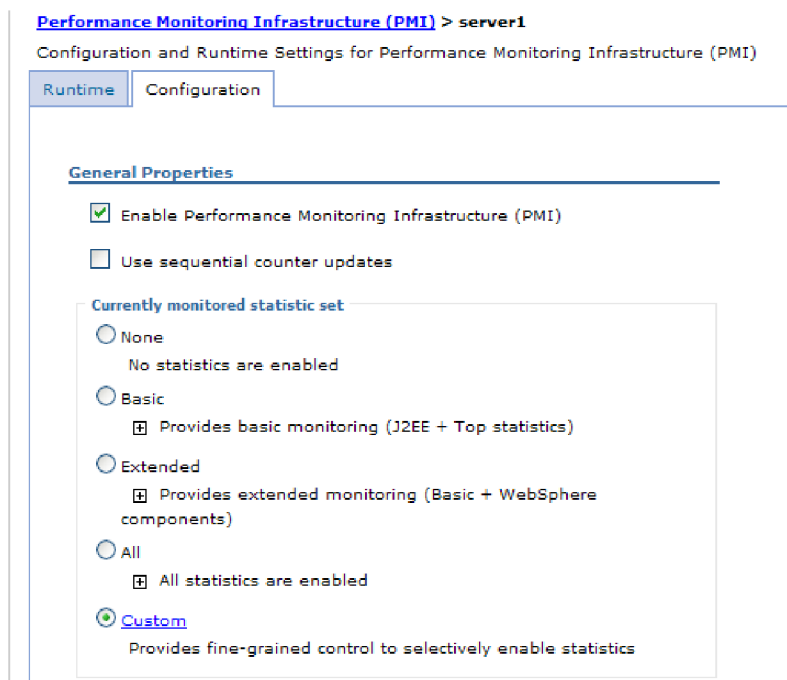


図 64. Performance Monitoring Infrastructure の使用可能化

- e. 「適用」または「OK」をクリックします。
  - f. 「保管」をクリックします。これで、PMI が使用可能になりました。
2. アダプター用に PMI を構成します。
    - a. 管理コンソールで「モニターおよびチューニング」を展開してから、「Performance Monitoring Infrastructure (PMI)」を選択します。
    - b. サーバーのリストから、ご使用のサーバーの名前をクリックします。
    - c. 「カスタム」を選択します。
    - d. 「ランタイム」タブを選択します。以下の図は、「ランタイム」タブを示しています。

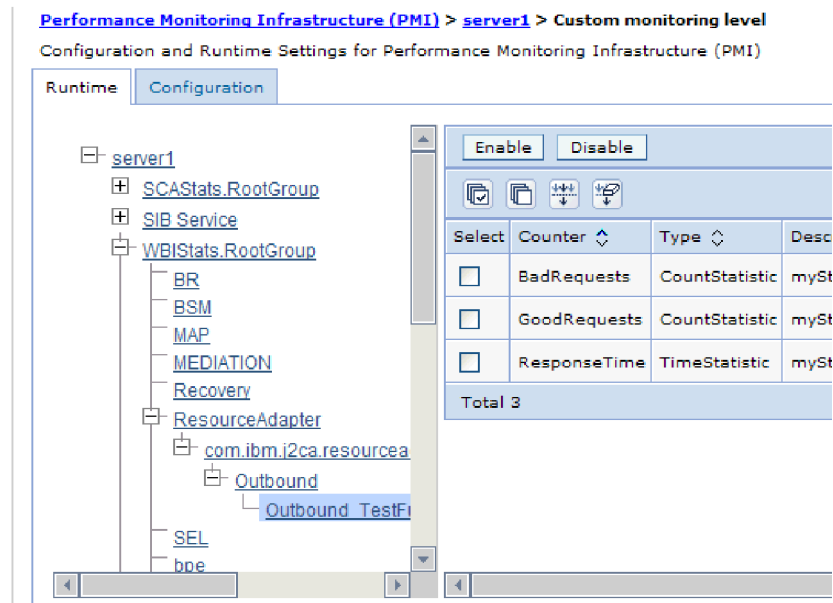


図 65. PMI の構成に使用される「ランタイム」タブ

- e. 「WBISStats.RootGroup」をクリックします。これは、ルート・グループで収集されるデータ用の PMI サブモジュールです。この例では、ルート・グループに WBISStats という名前を使用しています。
- f. 「ResourceAdapter」をクリックします。これは、JCA アダプターについて収集されるデータ用のサブモジュールです。
- g. アダプターの名前をクリックして、モニターするプロセスを選択します。
- h. 右側のペインで、収集する統計のチェック・ボックスを選択してから、「使用可能」をクリックします。

## 結果

PMI がアダプター用に構成されます。

## 次のタスク

これで、アダプターのパフォーマンス統計を表示できるようになりました。

## Common Event Infrastructure (CEI) によるトレースの使用可能化

このアダプターでは、サーバーに組み込まれているコンポーネントである Common Event Infrastructure を使用して、重要なビジネス・イベント (ポーリング周期の開始/停止など) に関するデータを報告することができます。イベント・データは構成設定に応じて、データベースまたはトレース・ログ・ファイルに書き込むことができます。

### このタスクの手順

1. 管理コンソールで「トラブルシューティング」をクリックします。
2. 「ログおよびトレース (Logs and Trace)」をクリックします。
3. サーバーのリストで、ご使用のサーバー名をクリックします。
4. 「ログ詳細レベルの変更」ボックスで、アダプターがイベント・データを書き込む先となる CEI データベースの名前 (WBIEventMonitor.CEI.ResourceAdapter.\* など) またはトレース・ログ・ファイルの名前 (WBIEventMonitor.LOG.ResourceAdapter.\* など) をクリックします。
5. データベースまたはトレース・ログ・ファイルにアダプターが書き込むビジネス・イベントについて、詳細レベルを選択します。オプションとして、メッセージおよびトレースに関連する詳細の細分度を調整することもできます。
  - 「ロギングなし」。イベントのロギングをオフにします。
  - 「メッセージのみ」。アダプターがイベントを報告します。
  - 「すべてのメッセージとトレース」。アダプターがイベントの詳細を報告します。
  - 「メッセージおよびトレース・レベル」。イベントに関連付けられたビジネス・オブジェクト・ペイロードについて、アダプターによる報告の詳細度を制御する設定です。詳細レベルを調整するには、以下のいずれかを選択します。
    - 「詳細 - 中」。アダプターは、イベントを報告しますが、ビジネス・オブジェクト・ペイロードについては一切報告しません。
    - 「詳細 - 高」。アダプターは、イベントを報告して、ビジネス・オブジェクト・ペイロードについての詳細も報告します。
    - 「詳細 - 最高」。アダプターは、イベントを報告して、ビジネス・オブジェクト・ペイロードについてもすべて報告します。
6. 「OK」をクリックします。

### 結果

イベントのロギングが使用可能になります。CEI 項目を確認するには、トレース・ログ・ファイルを表示するか、または管理コンソール内の Common Base Event ブラウザーを使用します。

## パフォーマンスに関する統計の表示

アダプターのパフォーマンス・データを、グラフィカル・モニター・ツール Tivoli Performance Viewer を使用して表示することができます。Tivoli Performance Viewer は、WebSphere Process Server の管理コンソールに組み込まれています。

## 始める前に

アダプター用の Performance Monitoring Infrastructure の構成。

### このタスクの手順

1. 管理コンソールで、「モニターおよびチューニング」を展開し、「Performance Viewer」を展開した後、「現行アクティビティ」を選択します。
2. サーバーのリストで、ご使用のサーバー名をクリックします。
3. サーバー名の下で、「パフォーマンス・モジュール」を展開します。
4. 「WBISStatsRootGroup」をクリックします。
5. 「ResourceAdapter」およびアダプター・モジュールの名前をクリックします。
6. 複数のプロセスがある場合は、統計を表示させるプロセスのチェック・ボックスを選択します。

### 結果

右側のパネルに統計が表示されます。「グラフの表示」をクリックして、データのグラフを表示するか、または「表の表示」をクリックして、統計を表形式で表示することができます。以下の図では、アダプターのパフォーマンス統計をグラフの形で表示しています。

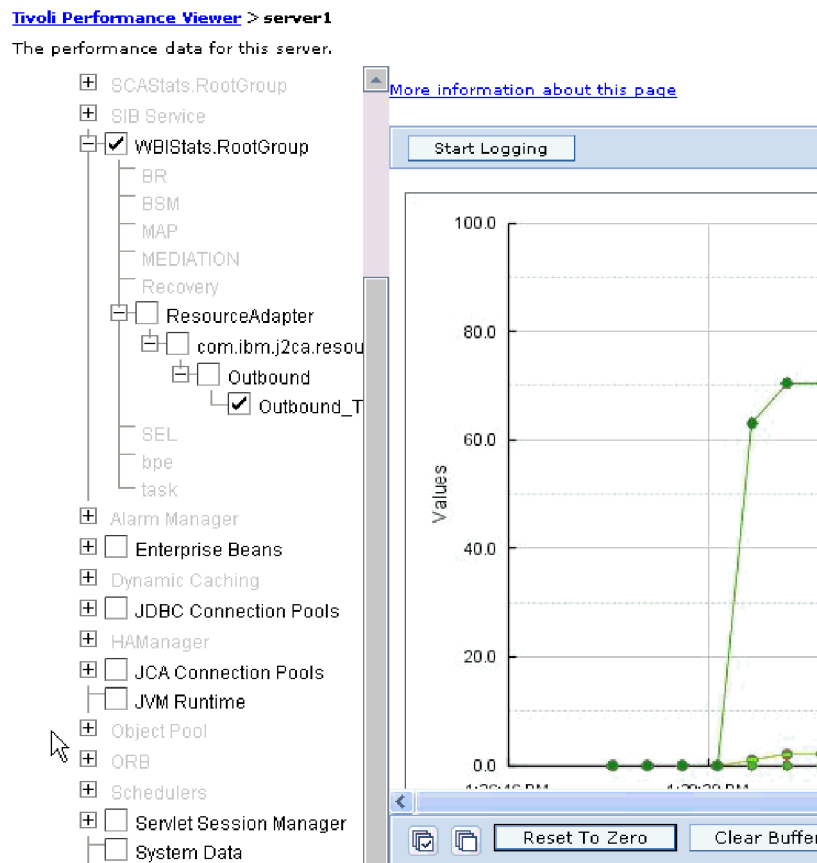


図 66. グラフ表示によるアダプターのパフォーマンス統計

---

## トラブルシューティングおよびサポート

共通のトラブルシューティング手法とセルフ・ヘルプ情報は、問題を迅速に識別して解決するのに役立ちます。

### ロギングおよびトレースの構成

要件に合うようロギングおよびトレースを構成します。アダプターのロギングを使用可能にし、イベント処理の状況を制御します。アダプターのログ・ファイル名およびトレース・ファイル名を変更して、その他のログ・ファイルおよびトレース・ファイルと区別します。

#### ロギング・プロパティの構成

ログを使用可能にし、ログの出力プロパティ（ログのロケーション、詳細レベル、出力形式など）を設定するには、管理コンソールを使用します。

#### このタスクを実行する理由および時期

モニター・イベントをアダプターでログに記録するには、モニターするサービス・コンポーネント・イベント・ポイント、各イベントに必要な詳細レベル、およびイベントをログにパブリッシュする際に使用する出力形式を指定する必要があります。管理コンソールを使用して、以下のタスクを実行します。

- 特定のイベント・ログを使用可能または使用不可にする。
- ログの詳細レベルを指定する。
- ログ・ファイルの保管場所と保持数を指定する。
- ログの出力形式を指定する。

ログ・アナライザーの出力形式を設定した場合は、ログ・アナライザー・ツール（プロセス・サーバーに同梱されるアプリケーション）を使用して、トレース出力を開くことができます。これは、2つの異なるサーバー・プロセスからのトレースを関連しようとする場合に便利です。なぜなら、これにより、ログ・アナライザーのマージ機能が使用できるからです。

プロセス・サーバー（サービス・コンポーネントとイベント・ポイントを含む）のモニターの詳細については、ご使用のプロセス・サーバーの資料を参照してください。

ログ構成は、静的または動的に変更できます。静的構成は、アプリケーション・サーバーを始動または再始動したときに有効になります。動的構成（実行時構成）の変更は、直ちに適用されます。

ログを作成すると、そのログの詳細レベルは構成データから設定されます。特定のログ名に対して構成データが提供されていない場合、そのログのレベルは、親ログから取得されます。親ログに構成データが存在しない場合は、更なる親ログを検査するという動作を繰り返し、非ヌル・レベルの値を持つログが見つかるまで、ツリーをさかのぼっていきます。ログのレベルを変更すると、その変更は子ログに伝搬されます。この変更は、必要に応じて、更なる子ログに再帰的に伝搬されます。

ロギングを使用可能にし、ログの出力プロパティを設定するには、以下の手順を実行します。

#### このタスクの手順

1. 管理コンソールのナビゲーション・ペインで、「サーバー」 → 「アプリケーション・サーバー」をクリックします。
2. 操作するサーバーの名前をクリックします。
3. 「トラブルシューティング」の下で「ログおよびトレース (Logs and trace)」をクリックします。
4. 「ログの詳細レベルの変更 (Change Log Detail Levels)」をクリックします。
5. いつ変更を有効にするのかを指定します。
  - 構成を静的に変更する場合は、「構成」タブをクリックします。
  - 構成を動的に変更する場合は、「実行時 (Runtime)」タブをクリックします。
6. 変更したいロギング・レベルのパッケージの名前をクリックします。  
WebSphere Adapters のパッケージ名は、**com.ibm.j2ca** で始まります。
  - アダプターの基本コンポーネントの場合は、「**com.ibm.j2ca.base**」を選択します。
  - アダプターの基本コンポーネント、およびデプロイされたすべてのアダプターの場合は、「**com.ibm.j2ca.base.\***」を選択します。
  - Adapter for FTP の場合のみ、**com.ibm.j2ca.ftp** パッケージを選択します。
7. ロギング・レベルを選択します。

ロギング・レベル (Logging Level)	説明
致命的	タスクを続行できないか、コンポーネントが機能しません。
重大	タスクは続行できませんが、コンポーネントはまだ機能します。このロギング・レベルには、差し迫った致命的エラーを示す状況、つまりリソースが枯渇寸前であることを強く示す状況も含まれます。
警告	潜在的なエラーが発生したか、重大なエラーが差し迫っています。このロギング・レベルには、進行性の障害 (リソース・リークの可能性など) を示す状況も含まれます。
監査	サーバーの状態またはリソースに影響を及ぼす重大なイベントが発生しました。
情報	タスクが実行中です。このロギング・レベルには、タスクの全体的な進行状況を示す一般情報が含まれます。
構成	構成の状況が報告されるか、構成の変更が発生しました。
詳細	サブタスクが実行中です。このロギング・レベルには、サブタスクの進行状況を詳しく示す一般情報が含まれます。

8. 「適用」をクリックします。
9. 「OK」をクリックします。
10. 静的な構成変更を有効にするには、プロセス・サーバーを停止し、再始動します。

## 結果

これ以降、ログ項目には、選択したアダプター・コンポーネントについての指定したレベルの情報が格納されます。

## ログ・ファイル名およびトレース・ファイル名の変更

アダプター・ログおよびトレース情報を他のプロセスとは分離して保持するには、管理コンソールを使用してファイル名を変更します。デフォルトでは、プロセス・サーバー上にあるすべてのプロセスおよびアプリケーションのログ情報およびトレース情報は、それぞれ `SystemOut.log` ファイルおよび `trace.log` ファイルに書き込まれます。

## 始める前に

アダプター・モジュールをアプリケーション・サーバーにデプロイした後は、ログ・ファイル名およびトレース・ファイル名はいつでも変更できます。

## このタスクを実行する理由および時期

ログ・ファイルおよびトレース・ファイルは、静的または動的に変更できます。アプリケーション・サーバーの始動時または再始動時に、静的変更が反映されます。動的変更またはランタイム構成変更は、即座に適用されます。

ログ・ファイルおよびトレース・ファイルは、`install_root/profiles/profile_name/logs/server_name` フォルダにあります。

ログ・ファイル名およびトレース・ファイル名を設定または変更するには、次の手順を実行します。

## このタスクの手順

1. 管理コンソールのナビゲーション・ペインで、「アプリケーション」>「エンタープライズ・アプリケーション」を選択します。
2. 「エンタープライズ・アプリケーション」リストから、アダプター・アプリケーションの名前をクリックします。これは、アダプターの EAR ファイルの名前から `.ear` ファイル拡張子を除いたものです。例えば、EAR ファイルの名前が `Accounting_OutboundApp.ear` である場合は、**Accounting\_OutboundApp** をクリックします。
3. 「構成」タブの「モジュール」リストから、「モジュールの管理」をクリックします。
4. モジュールのリストで、**IBM WebSphere Adapter for FTP** をクリックします。
5. 「構成」タブの「追加プロパティ」の下で、「リソース・アダプター」をクリックします。
6. 「構成」タブの「追加プロパティ」の下で、「カスタム・プロパティ」をクリックします。
7. 「カスタム・プロパティ」テーブル内で、ファイル名を変更します。
  - a. 「**logFilename**」をクリックして、ログ・ファイルの名前を変更します。あるいは、「**traceFilename**」をクリックして、トレース・ファイルの名前を変更します。



- b. 「構成」タブで、「値」フィールドに新しい名前を入力します。デフォルトでは、ログ・ファイルの名前は SystemOut.log、トレース・ファイルの名前は trace.log になります。
- c. 「適用」または「OK」をクリックします。変更内容がローカル・マシン上に保存されます。
- d. 変更内容をサーバー上のマスター構成に保存するには、次のいずれかの手順を実行します。
  - **静的変更:** サーバーを停止してから再始動します。この方法では、変更を行うことは可能ですが、サーバーを停止してから始動するまで、行った変更は有効になりません。
  - **動的変更:** 「カスタム・プロパティ」テーブルの上にあるメッセージ・ボックス内にある「保管」リンクをクリックします。プロンプトが出されたら、再度「保管」をクリックします。この方法では、行った変更をすぐに有効にすることができます。

## First Failure Data Capture (FFDC) のサポート

アダプターでは、First Failure Data Capture (FFDC) をサポートしています。これは、WebSphere Process Server または WebSphere Enterprise Service Busの実行時に発生した障害およびソフトウェアの深刻な誤動作を、永続的に記録するものです。

FFDC 機能はバックグラウンドで稼働し、実行時に発生したイベントおよびエラーを収集します。この機能によって、さまざまな障害の中から関連性を見つけて、その障害による影響を原因とリンクさせることがソフトウェアで可能になります。したがって、障害の根本原因がどこにあるかを迅速に探し出すことができるわけです。収集したデータは、アダプターの実行時に発生した例外処理を識別するために使用できます。

問題が発生すると、アダプターは例外メッセージおよびコンテキスト・データをログ・ファイルに書き込みます。このログ・ファイルは、`install_root/profiles/profile1/logs/ffdc` ディレクトリーにあります。

First Failure Data Capture (FFDC) について詳しくは、WebSphere Process Server または WebSphere Enterprise Service Bus の資料を参照してください。

## ビジネス・フォールト

このアダプターは、ビジネス・フォールトをサポートしています。これは例外の一種で、予期されるものを Outbound サービス記述、または Outbound インポートで宣言したものです。ビジネス・フォールトは、ビジネス・プロセスにおけるビジネス・ルール違反または制約違反が原因となって起こるため、発生箇所は予測可能です。

WebSphere Process Server および WebSphere Enterprise Service Bus ではさまざまなタイプのフォールトをサポートしていますが、このアダプターが生成するのはビジネス・フォールトのみです (本書では、単純にフォールト と呼びます)。すべての例外がフォールトになるわけではありません。フォールトは、アクション可能なエラー、つまり、アプリケーションの終了を必要としないリカバリー・アクションが可能なエラーに対して生成されます。例えば、必須データが含まれていない Outbound

処理のビジネス・オブジェクトをアダプターが受け取った場合、または Outbound 処理中にアダプターでエラーが発生した場合に、アダプターによってフォールトが生成されます。

## フォールト・ビジネス・オブジェクト

外部サービス・ウィザードは、アダプターで生成可能なフォールト用ビジネス・オブジェクトを作成します。さらに、このウィザードでは WBIFault スーパーセット・ビジネス・オブジェクトも作成します。これには、すべてのフォールトに共通する情報 (message、errorCode、primaryKeySet の各属性など) が含まれています。図 67 を参照してください。

WBIFault	
message	string
errorCode	string
primaryKeySet	PrimaryKeyPairType []

図 67. WBIFault ビジネス・オブジェクトの構造

このウィザードでは、以下のフォールト・ビジネス・オブジェクトを作成します。

- DuplicateRecordFault

アダプターは、Outbound Create 操作の処理時に、指定されたファイルが指定されたディレクトリー・パスに既に存在するためエラーが発生すると、このフォールトをスローします。

- RecordNotFoundFault

Append、Delete、Overwrite、Retrieve、ExecuteFTPScript、および ServerToServerFileTransfer の各操作の処理時に、指定されたディレクトリー・パスにファイル・ディレクトリー・パスまたはスクリプト・ファイルが存在しない場合、アダプターはこのフォールトをスローします。

- MissingDataFault

アダプターは、ファイル内容が NULL またはファイル名やディレクトリー・パスが空の場合など、必要な値が提供されないときに、このフォールトをスローします。Retrieve 操作中に、区切り文字が NULL または無効なためエラーが発生した場合に、アダプターはこのフォールトをスローします。

## フォールト処理用のモジュールの構成

ビジネス・フォールトをサポートするモジュールを構成する前に、外部サービス・ウィザードを使用してモジュールを構成しておく必要があります。

フォールト処理を可能にするには、モジュールの .import ファイルおよび WSDL ファイルを変更しなければなりません。フォールトは、バインディング・レベルまたはメソッド・レベルのいずれかで構成できます。バインディング・レベルで変更を

行うと、インポートのすべてのメソッドに適用されます。メソッド・バインディング・レベルで変更を行う場合は、メソッドごとに異なるフォールトを構成できません。

各フォールトのフォールト名およびフォールト・バインディングを、表 6 にリストします。モジュールを構成するときには、このフォールト名とフォールト・バインディング・クラスを使用してください。

表 6. 各フォールトのフォールト名およびフォールト・バインディング・クラス

フォールト名	関連するフォールト・バインディング・クラス
DUPLICATE_RECORD	com.ibm.j2ca.extension.emd.runtime.WBIFaultDataBindingImpl
MISSING_DATA	com.ibm.j2ca.extension.emd.runtime.WBIFaultDataBindingImpl
RECORD_NOT_FOUND	com.ibm.j2ca.extension.emd.runtime.WBIFaultDataBindingImpl

1. `.import` ファイルを編集して、バインディング・レベルまたはメソッド・レベルのいずれかでフォールトを構成します。
  - バインディング・レベルでフォールトを構成するには、以下のようになります。
    - a. バインディング・セクションに、`faultSelector` 属性およびフォールト・セレクターの名前を追加します。フォールト・セレクターの名前は、`com.ibm.j2ca.extension.emd.runtime.WBIFaultSelectorImpl` です。
    - b. 使用可能にするフォールトごとに、`<faultBinding>` エレメントを追加します。表 6 のフォールト名およびフォールト・データ・バインディング・クラス名を、このエレメントに指定します。

次の `.import` ファイルでは、すべてのメソッドに `DUPLICATE_RECORD`、`MISSING_DATA`、および `RECORD_NOT_FOUND` の各フォールトを構成しています。太字部分は、フォールト処理を可能にするために変更した箇所を示します。

```
<esbBinding xsi:type="eis:EISImportBinding"
  dataBindingType="com.ibm.j2ca.ftpfile.emd.runtime.FTPFileBaseDataBinding"
  faultSelector="com.ibm.j2ca.extension.emd.runtime.WBIFaultSelectorImpl">
  <resourceAdapter name="FTPOutApp.IBM WebSphere Adapter for FTP Files"
    type="com.ibm.j2ca.ftpfile.FTPFileResourceAdapter">
    <properties/>
  </resourceAdapter>
  <faultBinding
    fault="DUPLICATE_RECORD"
    faultBindingType="com.ibm.j2ca.extension.emd.runtime.WBIFaultDataBindingImpl"/>
  <faultBinding
    fault="RECORD_NOT_FOUND"
    faultBindingType="com.ibm.j2ca.extension.emd.runtime.WBIFaultDataBindingImpl"/>
  <faultBinding
    fault="MISSING_DATA"
    faultBindingType="com.ibm.j2ca.extension.emd.runtime.WBIFaultDataBindingImpl"/>
```

- メソッド・レベルでフォールトを構成するには、以下のようになります。
  - a. フォールトに関連付けるメソッドのメソッド・バインディング・セクションに、フォールト・セレクターの名前を追加します。フォールト・セレクターの値は、`com.ibm.j2ca.extension.emd.runtime.WBIFaultSelectorImpl` です。
  - b. フォールト・バインディング・エレメントを、メソッド・バインディング・セクションに追加します。表 6 にあるフォールト名と対応するフォールト・データ・バインディング・クラス名を使用します。

次の .import ファイルでは、createCUSTOMER メソッドに DUPLICATE\_RECORD および RECORD\_NOT\_FOUND の各フォールトを構成しています。太字部分は、フォールト処理を可能にするために変更した箇所を示します。

```
<methodBinding
  inDataBindingType="com.ibm.xmlns.prod.wbi.j2ca.ftp.customerbg.CustomerBGDataBinding"
  method="createCUSTOMER"
  outDataBindingType="com.ibm.xmlns.prod.wbi.j2ca.ftp.customerbg.CustomerBGDataBinding"
  faultSelector="com.ibm.j2ca.extension.emd.runtime.WBIFaultSelectorImp1">
  <interaction>
    <properties>
      <functionName>Create</functionName>
    </properties>
  </interaction>
  <faultBinding fault="DUPLICATE_RECORD"
    faultBindingType="com.ibm.j2ca.extension.emd.runtime.WBIFaultDataBindingImp1"/>
  <faultBinding fault="RECORD_NOT_FOUND"
    faultBindingType="com.ibm.j2ca.extension.emd.runtime.WBIFaultDataBindingImp1"/>
</methodBinding>
```

2. フォールトのターゲット・ネーム・スペースを決定します。使用可能にするフォールトごとに、以下の方法でネーム・スペースを決定します。

- a. フォールト・スキーマ (XSD ファイル) をテキスト・エディターで開きます。
- b. ターゲット・ネーム・スペースを見つけます。以下のフォールト・スキーマの太字で示された部分が、ターゲット・ネーム・スペースになります。

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://com.ibm/j2ca/fault/afc/fault"
  xmlns:basefault="http://com.ibm/j2ca/fault">
<import namespace="http://com.ibm/j2ca/fault" schemaLocation="WBIFault.xsd"/>
```

...

フォールトのターゲット・ネーム・スペースは、すべて同じにすることもできますし、異なるものにすることもできます。

3. WSDL ファイルを編集して、サービスにフォールトを宣言します。このリストの最後に、各変更箇所を強調したサンプル WSDL ファイルが記載されています。
  - a. <definitions> エレメントに、フォールト・スキーマ・ファイルから取得した情報を使用して、フォールト・ネーム・スペースごとにネーム・スペースを追加します。すべてのフォールト・スキーマが同一の targetNamespace を持つ場合は、別名を 1 つだけ追加します。異なる targetNamespace を持つ場合は、個々のネーム・スペースに別名を追加します。
  - b. 使用可能にするフォールトごとに、スキーマをインポートするための <xsd:import> エレメントを作成します。
  - c. 各フォールト・タイプで import ステートメントを宣言します。ステップ 3a で定義した正しい別名を使用して、type=alias:faultBOName.xsd の複合タイプを解決してください。
  - d. 各フォールト・タイプにメッセージ・タグを宣言します。
  - e. フォールトを処理する必要があるメソッドごとに、フォールト宣言を追加します。

次の WSDL ファイルでは、DUPLICATE\_RECORD フォールトおよび RECORD\_NOT\_FOUND フォールトを定義しています。太字部分は、フォールト処理を可能にするために変更した箇所を示します。

```

<definitions
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:CustomerBG="http://www.ibm.com/xmlns/prod/wbi/j2ca/ftpfile/customerbg"
  xmlns:intf="http://FTPOut/FTPFileOutboundInterface"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:fault="http://com/ibm/j2ca/fault/afcfault"
  targetNamespace="http://FTPOut/FTPFileOutboundInterface">
  <types>
  <xsd:schema
    xmlns:tns="http://FTPOut/FTPFileOutboundInterface"
    xmlns:xsd1="http://www.ibm.com/xmlns/prod/wbi/j2ca/ftpfile/customerbg"
    elementFormDefault="qualified"
    targetNamespace="http://FTPOut/FTPFileOutboundInterface"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:import
    namespace="http://www.ibm.com/xmlns/prod/wbi/j2ca/ftpfile/customerbg"
    schemaLocation="CustomerBG.xsd"/>
  <xsd:import namespace="http://com/ibm/j2ca/fault/afcfault"
    schemaLocation="DuplicateRecordFault.xsd"/>
  <xsd:import namespace="http://com/ibm/j2ca/fault/afcfault"
    schemaLocation="RecordNotFoundFault.xsd"/>
  . . .
  <xsd:element name="duplicateRecordFaultX">
  <xsd:complexType>
  <xsd:sequence>
  <xsd:element name="duplicateRecordFaultElement"
    type="fault:DuplicateRecordFault"/>
  </xsd:sequence>
  </xsd:complexType>
  </xsd:element>

  <xsd:element name="recordNotFoundFaultX">
  <xsd:complexType>
  <xsd:sequence>
  <xsd:element name="recordNotFoundFaultElement"
    type="fault:RecordNotFoundFault"/>
  </xsd:sequence>
  </xsd:complexType>
  </xsd:element>
  </xsd:schema>
  </types>
  . . .
  <message name="duplicateRecordFault">
  <part element="intf:duplicateRecordFaultX"
    name="duplicateRecordFaultPart"/>
  </message>
  <message name="recordNotFoundFault">
  <part element="intf:recordNotFoundFaultX"
    name="recordNotFoundFaultPart"/>
  </message>
  <operation name="createCUSTOMER">
  <input message="intf:createCUSTOMERRequest" name="createCUSTOMERRequest"/>
  <output message="intf:createCUSTOMERResponse" name="createCUSTOMERResponse"/>

```

ステップ 3a (126 ページ)

ステップ 3b (126 ページ)

ステップ 3c (126 ページ)

ステップ 3d (126 ページ)

ステップ  
3e (126 ページ)

```
<fault message="intf:duplicateRecordFault"
      name="duplicateRecordFaultFault" />
<fault message="intf:recordNotFoundFault"
      name="recordNotFoundFaultFault" />
</operation>
</portType>
</definitions>
```

## XAResourceNotAvailableException

com.ibm.ws.Transaction.XAResourceNotAvailableException 例外の報告がプロセス・サーバーのログに繰り返し含まれているときは、トランザクション・ログを除去し、問題を訂正してください。

### 症状:

アダプターが始動すると、プロセス・サーバーのログ・ファイルに以下の例外が繰り返し記録されます。

```
com.ibm.ws.Transaction.XAResourceNotAvailableException
```

### 問題:

プロセス・サーバーがリソースのトランザクションをコミットまたはロールバックしている間に、そのリソースが除去されました。アダプターは、始動するとトランザクションのリカバリーを試みますが、リソースが除去されているため、それができません。

### 解決策:

この問題を訂正するには、以下の手順を実行します。

1. プロセス・サーバーを停止します。
2. そのトランザクションを含むトランザクション・ログ・ファイルを削除します。例外トレース内の情報を使用して、トランザクションを識別します。これにより、サーバーは、それらのトランザクションのリカバリーを試みないようになります。

**注:** テスト環境または開発環境では、通常はトランザクション・ログをすべて削除できます。WebSphere Integration Developer では、トランザクション・ログ・ディレクトリー `server_install_directory\profiles\profile_name\tranlog` に含まれるファイルとサブディレクトリーを削除します。

実稼働環境では、処理する必要のないイベントを表すトランザクションのみを削除します。これを行う方法の一つは、アダプターを再インストールし、使用した元のイベント・データベースをそのアダプターに参照させ、不要なトランザクションのみを削除することです。もう一つの方法は、以下のディレクトリー内の `log1` ファイルまたは `log2` ファイルからトランザクションを削除することです。

```
server_install_directory\profiles\profile_name\tranlog\node_name\wps\
server_name\transaction\tranlog
```

3. プロセス・サーバーを始動します。

## org.xml.sax.SAXParseException

アダプターが XML データ・ハンドラーで構成した場合、その内容が指定されたビジネス・オブジェクト形式でなければ、org.xml.sax.SAXParseException 例外が生成されます。この問題を訂正するには、ファイル内容がビジネス・オブジェクト構造と一致しているかどうか確認してください。複数のビジネス・オブジェクトがファイルに含まれている場合は、区切り文字が正しく指定されているかどうか確認してください。

### 症状:

アダプターを XML データ・ハンドラーで構成したときに、以下の例外がスローされます。

org.xml.sax.SAXParseException: この内容は末尾のセクションでは許可されていません。

### 問題:

ファイルの内容は、指定されたビジネス・オブジェクト形式ではありません。

### 解決策:

この問題を訂正するには、以下の手順を実行します。

1. ファイル内容がビジネス・オブジェクト構造と一致しているかどうか確認します。
2. 複数のビジネス・オブジェクトがコンテンツ・ファイルに含まれている場合は、区切り文字が正しく指定されているかどうか確認します。

## セルフ・ヘルプ・リソース

IBM ソフトウェア・サポートのリソースを使うと、最新のサポート情報や技術資料を入手したり、サポート・ツールやフィックスをダウンロードしたり、WebSphere Adapters の問題を回避したりすることができます。また、セルフ・ヘルプ・リソースは、アダプター関連の問題を診断する際にも役立ちますし、IBM ソフトウェア・サポートへの連絡方法に関する情報も提供します。

### サポート Web サイト

WebSphere Adapters ソフトウェアのサポート Web サイト (<http://www.ibm.com/software/integration/wbiadapters/support/>) では、WebSphere Adapters の学習、使用、およびトラブルシューティングに役立つ多数のリソースへのリンクを提供しています。以下の種類のリソースがあります。

- フラッシュ (製品に関する警告)
- 製品のインフォメーション・センター、マニュアル、IBM Redbooks®、ホワイト・ペーパーなどの技術情報
- 教育関連のオフライン
- 技術情報

## 推奨フィックス

適用したほうがよい推奨フィックスのリストは、以下のロケーションで入手できます。<http://www.ibm.com/support/docview.wss?fdoc=aimadp&rs=695&uid=swg27010397>

## 技術情報

技術情報は、Adapter for FTP に関する最新の資料を提供します。以下のトピックがあります。

- 問題とそれに対する現在使用可能な解決策
- よくある質問に対する答え
- アダプターのインストール、構成、使用法、トラブルシューティングに関する手引きとなる情報
- *IBM* ソフトウェア・サポート・ハンドブック

WebSphere Adapters の技術情報のリストを参照するには、以下のアドレスにアクセスしてください。

[http://www.ibm.com/support/search.wss?tc=SSMKUK&rs=695&rank=8  
&dc=DB520+D800+D900+DA900+DA800+DB560&dtm](http://www.ibm.com/support/search.wss?tc=SSMKUK&rs=695&rank=8&dc=DB520+D800+D900+DA900+DA800+DB560&dtm)

## IBM Support Assistant のプラグイン

Adapter for FTP では、IBM Support Assistant のプラグインを用意しています。これは、無料のローカル・ソフトウェア保守性ワークベンチです。IBM Support Assistant のインストール方法および使用法については、以下のアドレスにアクセスしてください。

<http://www.ibm.com/software/support/isa/>



---

## 第 8 章 参照情報

ユーザーの作業をサポートするための参照情報として、外部サービス・ウィザードによって生成されるビジネス・オブジェクトに関する詳細情報や、アダプター・プロパティに関する情報 (双方向変換をサポートするアダプター・プロパティなど) を提供しています。また、アダプターのメッセージや関連製品情報についても示しています。

---

### ビジネス・オブジェクトの情報

ビジネス・オブジェクト定義ファイル内のアプリケーション固有情報およびビジネス・オブジェクト名を調査することによって、ビジネス・オブジェクトの目的を特定できます。アプリケーション固有情報は、FTP サーバー で実行できる操作を決定します。名前は、通常実行される操作およびビジネス・オブジェクトの構造を反映します。

### ビジネス・オブジェクトの構造

アダプターでは、3 種類のビジネス・オブジェクト構造をサポートしています。非構造化データを渡すために使用される汎用ビジネス・オブジェクト。データおよび接続固有の情報に対して実行されるアクションを含むビジネス・グラフを備えた汎用ビジネス・オブジェクト。コンテンツ固有のビジネス・オブジェクトである、ユーザー定義型。非常に限定的なビジネス・オブジェクト構造 (Customer、Order などのビジネス・オブジェクト) をサポートします。

ビジネス・グラフは、オプションになり、外部サービス・ウィザードで選択できません。

FTPFileBG、FTPFile、および UnstructuredContent 汎用ビジネス・オブジェクト定義は、自動的に生成されます。外部サービス作成時に選択したカスタム複合タイプに応じて、対応するビジネス・オブジェクトまたはオブジェクトの定義も生成されます。例えば、オプションのビジネス・グラフを含む Customer を選択すると、ビジネス・オブジェクト CustomerWrapperBG および CustomerWrapper が生成されます。

### FTPFileBG

FTPFileBG ビジネス・オブジェクトは、汎用ビジネス・オブジェクトの一種で、動詞 (データに対して実行されるアクション) および FTPFile ビジネス・オブジェクトが子として含まれています。以下の図に、この関係を示します。

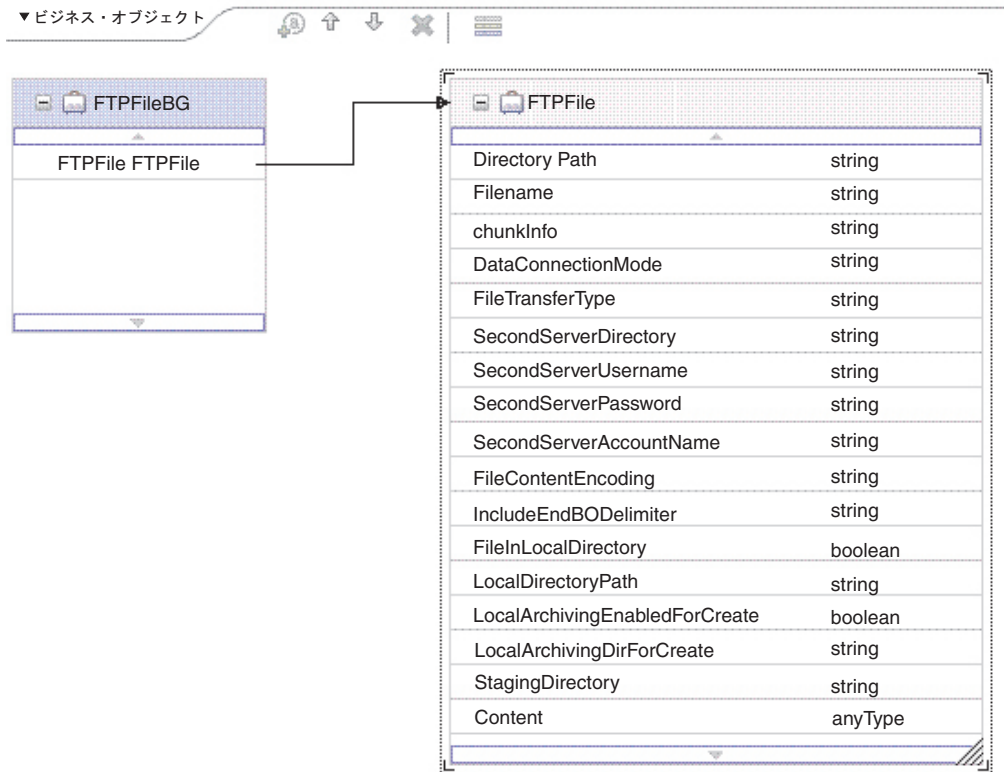


図 68. FTPFileBG ビジネス・オブジェクト

## FTPFile

FTPFile ビジネス・オブジェクトには、すべての必要な接続情報と、UnstructuredContent ビジネス・オブジェクトが子として含まれています。以下の図に、この関係を示します。

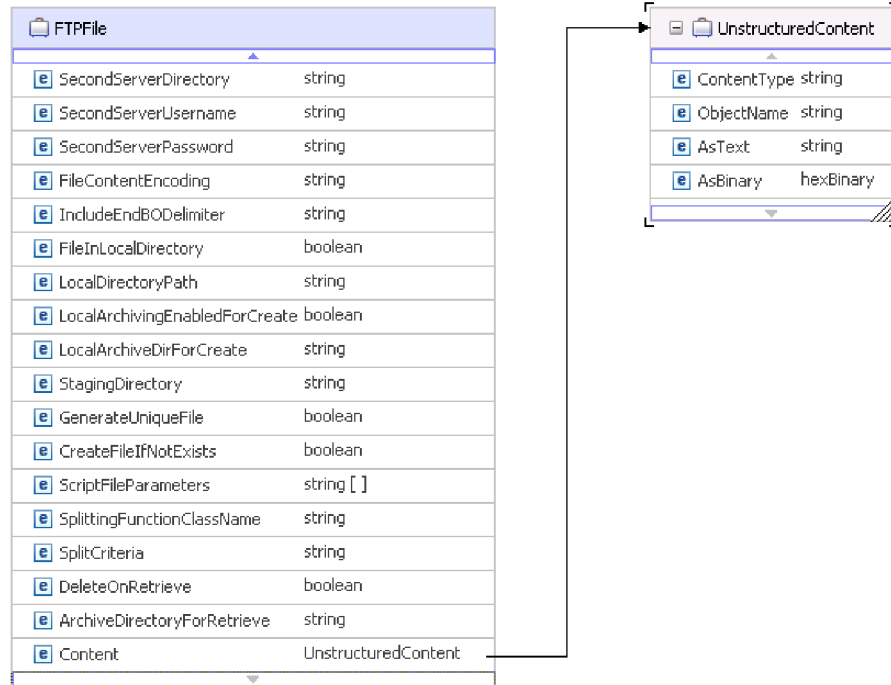


図 69. *FTPFile* ビジネス・オブジェクト

## CustomerWrapperBG

CustomerWrapperBG は、動詞 (データに対して実行されるアクション) および CustomerWrapper ビジネス・オブジェクトを子として含むビジネス・オブジェクトです。以下の図に、この関係を示します。

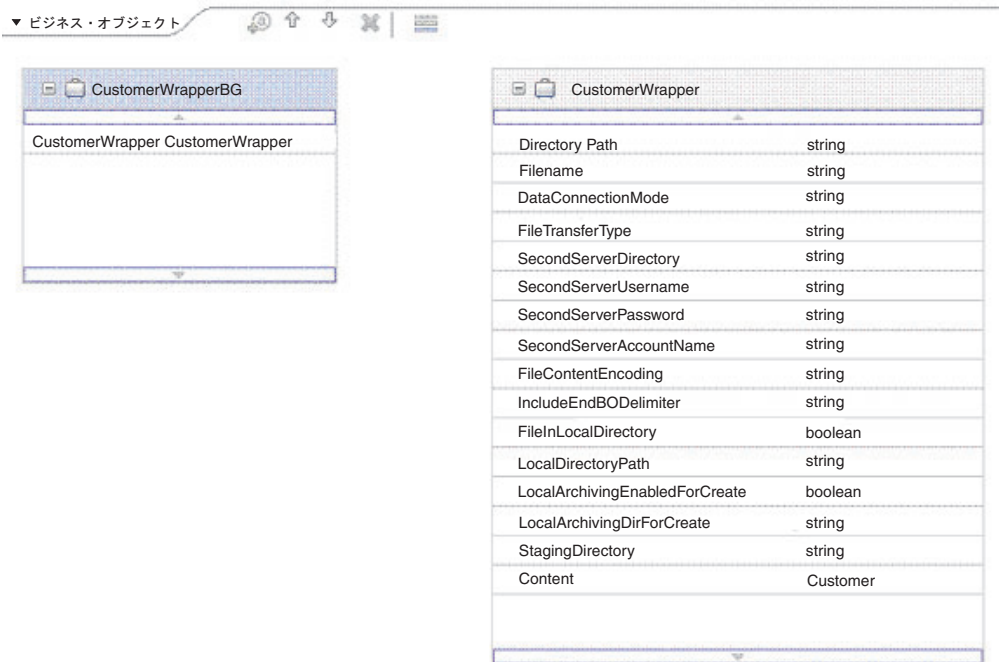


図 70. *CustomerWrapperBG* ビジネス・オブジェクト

## CustomerWrapper

*CustomerWrapper* ビジネス・オブジェクトは、すべての必要な接続情報およびコンテンツ固有の *Customer* ビジネス・オブジェクトを子として含むビジネス・オブジェクトです。以下の図に、この関係を示します。

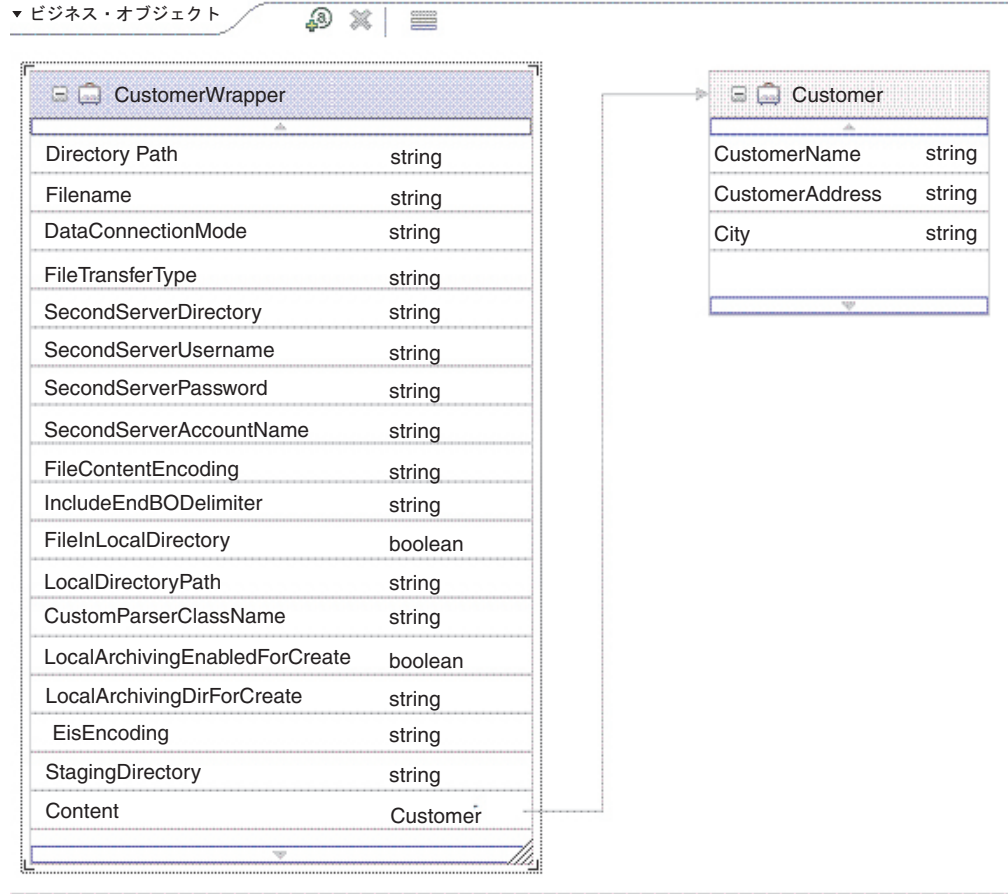


図 71. CustomerWrapper ビジネス・オブジェクト

## 命名規則

外部サービス・ウィザードがビジネス・オブジェクトを生成する際、ビジネス・オブジェクトの構築に使用する FTP サーバー 内のオブジェクト名に基づいて、ビジネス・オブジェクトに名前を提供します。ビジネス・オブジェクト・エディターを使用して、ユーザー定義オブジェクトを作成します。

外部サービス・ウィザードは、オブジェクト名を大/小文字混合に変換します。これは、スペースまたは下線などのあらゆる分離文字を除去し、各単語の先頭文字を大文字にすることを意味します。例えば、外部サービス・ウィザードが CUSTOMER\_ADDRESS という名前の FTP サーバー・オブジェクトを使用してビジネス・オブジェクトを生成する場合、CustomerAddress という名前のビジネス・オブジェクトを生成します。

生成したビジネス・オブジェクト名で、ビジネス・オブジェクトの構造を示すことができます。ただし、ビジネス・オブジェクト名には、アダプターを意味する値は含まれません。これは、ビジネス・オブジェクト名を変更しても、ビジネス・オブジェクトの動作は変わらないことを意味します。

**重要:** ビジネス・オブジェクトを名前変更する場合、WebSphere Integration Developer のリファクタリング機能を使用して、すべてのビジネス・オブジェクト依存関係を確実に更新してください。リファクタリングを使用したビジネス・オブジェクトの名前変更についての説明は、以下のリンクを参照してください:

<http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.wbit.help.refactor.doc/topics/trenameboatt.html>

**注:** ビジネス・グラフの生成はオプションで、WebSphere Process Server の場合のみサポートされています。

## ビジネス・オブジェクト属性プロパティー

ビジネス・オブジェクト・アーキテクチャーは、属性に適用されるさまざまなプロパティーを定義します。このセクションでは、アダプターがこれらのプロパティーを解釈する方法について説明します。

以下の表に、これらのプロパティーと、アダプターがプロパティーを解釈する方法を示します。

表7. ビジネス・オブジェクト属性プロパティー

プロパティー	説明
Cardinality	単純属性の場合、1 が使用されます。コンテナ属性の場合、メソッド要件に従って n が使用されます。
Foreign Key	アダプターには、Foreign Key を表す特定の要素はありません。
Key	アダプターには、Key を表す特定の要素はありません。
Name	このプロパティーは、属性の固有の名前 (単純属性の場合)、またはビジネス・オブジェクトの名前 (子ビジネス・オブジェクトの場合) を表します。
必須	このプロパティーは、属性に値を入れる必要があるかどうかを指定します。
Type	属性タイプは、単純または複合です。単純タイプは、ブール、ストリング、LongText、整数、浮動小数点、倍精度浮動小数点、およびバイト [ ] です。標準的な複合タイプは、別のビジネス・オブジェクトの名前です。

## ビジネス・オブジェクトの操作サポート

操作は、アダプターによってビジネス・オブジェクトに対して実行されるアクションの名前です。あらゆるビジネス・オブジェクトは、関連付けられた操作を保有します。通常、操作の名前は、ビジネス・オブジェクトに対するアクションのタイプを示します。

以下の表は、アダプターがサポートする操作を定義しています。

表8. ビジネス・オブジェクトのサポートされる操作

操作	結果
Create	指定されたファイル名のファイルが、指定されたディレクトリーに、要求によって送信された内容で作成される。
Append	要求の内容がファイルの終わりに付加される。
Retrieve	要求で指定したファイルの内容を戻す。
Delete	要求で指定されたディレクトリーからファイルを削除する。
Overwrite	ディレクトリーのファイルに要求で指定された内容を上書きする。

表 8. ビジネス・オブジェクトのサポートされる操作 (続き)

操作	結果
Exists	要求のファイルが指定のディレクトリーまたはサブディレクトリーに存在する場合は、成功応答を返す。
List	指定されたディレクトリーに存在するすべてのファイル名を返す。
ServerToServerFileTransfer	1 つの FTP サーバーから別の FTP サーバーへファイルを転送する。
ExecuteFTPScript	指定されたディレクトリーで FTP スクリプト・ファイルを実行する。

## カスタム・ビジネス・オブジェクト

カスタム・ビジネス・オブジェクトを使用する場合、外部サービス・ウィザードを実行する前に、WebSphere Integration Developer ビジネス・オブジェクト・ウィザードを使用して、事前定義ビジネス・オブジェクトを作成する必要があります。ウィザードによって作成されるビジネス・オブジェクト定義は、ローカル・システムに xsd ファイルとして保管されます。外部サービス・ウィザードでビジネス・オブジェクトを作成する場合、ビジネス・オブジェクト・ウィザードで作成された事前定義ビジネス・オブジェクトが検索され、モジュール固有のデータがそのオブジェクトに取り込まれます。

事前定義ビジネス・オブジェクトの作成方法については、WebSphere Integration Developer の資料を参照してください。

---

## カスタム・ファイル分割

分割ロジックを含むカスタム・クラスを実装することができます。アダプターは、このクラス用の Java™ インターフェースを提供します。インターフェースの詳細を以下に示します。

```
public interface SplittingFunctionalityInterface extends Iterator{
    public int getTotalBOs(String filename) throws SplittingException;
    public void setBODetails(String filename, int currentPosition, int totalBOs,
        boolean includeEndBODelimiter) throws SplittingException;
    public void setSplitCriteria(String splitCriteria);
    public void setEncoding(String encoding);
    public void setLogUtils(LogUtils logUtils);
    public boolean isSplitBySize()
}
```

- `public int getTotalBOs(String filename) throws SplittingException`

このメソッドは、filename で指定されるイベント・ファイル内に存在するビジネス・オブジェクトの総数を返します。

- `public void setSplitCriteria(String splitCriteria)`

このメソッドは、splitCriteria を取ります。これは、イベント・ファイル内に存在するビジネス・オブジェクトの数に基づいています。各ビジネス・オブジェクトは、next() 呼び出しで返されます。

- `public void setLogUtils(LogUtils logUtils)`

このメソッドを使用して LogUtils オブジェクトを設定します。これは、ユーザーがトレースおよびログ・メッセージをファイルに書き込むために使用するクラスです。

- `public void setEncoding(String encoding)`

このメソッドを使用して、イベント・ファイル内容のエンコードを設定します。ファイル内容の読み取り時に、このエンコードが使用されます。このエンコードは、`SplitCriteria` にも使用されます。

- `public void setBODetails(String filename, int currentPosition, int totalBOs, boolean includeEndBODelimiter) throws SplittingException`

このメソッドを使用して、現在のビジネス・オブジェクト番号を設定し、`next()` 呼び出しが行われるたびに `currentPosition` に設定されているビジネス・オブジェクト番号が返されるようにします。このメソッドは `includeEndBODelimiter` パラメーターも取ります。このパラメーターを `true` に設定すると、ビジネス・オブジェクトの内容の末尾に `SplitCriteria` が組み込まれます。このメソッドは、常に `next()` 呼び出しの前に呼び出す必要があります。これは、`next()` メソッドが、このメソッドで設定したビジネス・オブジェクトのビジネス・オブジェクトの内容を返すようにするためです。

- イテレーターには、`hasNext()`、`next`、および `remove()` という 3 つのメソッドがあり、これらも実装する必要があります。`next()` メソッドは、`setBODetails()` で設定されたビジネス・オブジェクト位置のビジネス・オブジェクト内容を (`byte[]` として) 返します。ビジネス・オブジェクトの位置が設定されていない場合は失敗します。`hasNext()` メソッドは、`setBODetails()` で設定されたビジネス・オブジェクトの位置が存在するかどうかを示します。`hasNext()` 呼び出しの前に、`setBODetails()` メソッドを呼び出す必要があります。`remove()` メソッドは、イベント・パーシスタンス・テーブルから削除されるビジネス・オブジェクト項目ごとに呼び出されます。このメソッドでイベント・ファイルを削除しないでください。使用されているリソースのみをクリーンアップしてください。
- `public boolean isSplitBySize()`

このメソッドは、イベント・ファイルの構文解析を、サイズまたは区切り文字のいずれに基づいて行うかを示します。

---

## Outbound 構成プロパティー

WebSphere Adapter for FTP には、複数のカテゴリーに分けられた Outbound 接続構成プロパティーがあります。これらは、オブジェクトおよびサービスの生成時または作成時に、外部サービス・ウィザード を使って設定します。WebSphere Integration Developer または WebSphere Process Server の管理コンソールを使ってモジュールを WebSphere Process Server にデプロイした後に、リソース・アダプター・プロパティーおよび管理接続ファクトリー・プロパティーを変更することができます。ただし、外部サービス・ウィザード の接続プロパティーは、デプロイメント後に変更することはできません。

### プロパティー関連の情報ガイド

WebSphere Adapter for FTP の構成に使用されるプロパティー (リソース・アダプター・プロパティー、管理接続ファクトリー・プロパティー) の詳細が、構成プロパティーの各トピックごとに、表形式で記載されています。これらの表を使いやすくするために、参照する機会が多い各行の情報は、以下のような形式で表されています。



構成プロパティの表で示される各行の意味について、以下の表で説明します。

行	説明
必須	<p>必須フィールド (プロパティ) には、アダプターが機能するための値を設定する必要があります。場合によっては、外部サービス・ウィザードが必須プロパティについてデフォルト値を提供します。</p> <p>外部サービス・ウィザード上の必須フィールドからデフォルト値を除去しても、そのデフォルト値は変更されません。必須フィールドに値がまったく含まれていない場合、外部サービス・ウィザードは割り当てられたデフォルト値を使用して、フィールドを処理します。このデフォルト値は、管理コンソールにも表示されます。</p> <p>使用可能な値は「はい」および「いいえ」です。</p> <p>他のプロパティが特定の値を保有する場合のみ、必須になるプロパティもあります。この場合、表でこの依存関係を示します。例えば以下のとおりです。</p> <ul style="list-style-type: none"> <li>• EventQueryType プロパティが Dynamic に設定されている場合、「はい」</li> <li>• Oracle データベースの場合、「はい」</li> </ul>
使用可能な値	<p>プロパティで選択できる値をリストおよび説明します。</p>
デフォルト	<p>外部サービス・ウィザードによって設定された事前定義値。プロパティが必須の場合、デフォルト値を受け入れるか、または固有の値を指定します。プロパティにデフォルト値がない場合、表にはデフォルト値なしと記載されます。</p> <p>「なし」という単語は、受け入れ可能なデフォルト値で、デフォルト値がないことを意味するわけではありません。</p>
計測単位	<p>プロパティを計測する単位を指定します (キロバイト、秒など)。</p>
プロパティ・タイプ	<p>プロパティ・タイプを記述します。有効なプロパティ・タイプには以下が含まれます。</p> <ul style="list-style-type: none"> <li>• Boolean</li> <li>• String</li> <li>• Integer</li> </ul>
使用法	<p>プロパティに適用される可能性がある使用条件または使用制限について記述します。例として、制約の記載方法を示します。</p> <p>WebSphere Application Server バージョン 6.40、およびそれより前のバージョンでは、パスワードについて以下の制限があります。</p> <ul style="list-style-type: none"> <li>• 大文字である必要があります</li> <li>• 長さが 8 文字である必要があります</li> </ul> <p>WebSphere Application Server の 6.40 より後のバージョンでは、パスワードの制限が以下のように変更されました。</p> <ul style="list-style-type: none"> <li>• 大文字小文字を区別しません</li> <li>• 長さが 40 文字まで可能です。</li> </ul> <p>このセクションでは、このプロパティに影響がある、またはこのプロパティに影響を受ける他のプロパティをリストし、条件関係の性質を記述します。</p>
例	<p>例として、サンプル・プロパティ値を提供します。</p> <p>"言語を JA (日本語) に設定した場合は、コード・ページ番号は 8000 に設定されます"</p>

行	説明
グローバル化	プロパティーが国際化されている場合、各国語をサポートしているため、自国語で値を設定できます。  有効な値は、「はい」 および 「いいえ」です。
BIDI 対応	双方向 (BIDI) 処理でプロパティーをサポートするかどうかを示します。双方向処理は、同じファイル内で左から右のセマンティック・コンテンツ (ヘブライ語やアラビア語など) と右から左のセマンティック・コンテンツ (URL またはファイル・パスなど) の両方を含むデータを処理するタスクのことです。  有効な値は、「はい」 および 「いいえ」です。

## アダプター・タイプ・プロパティー

アダプター・タイプ・プロパティーは、外部サービス・ウィザードにアダプターの詳細情報を提供します。これらのプロパティーは、デプロイメント前に 外部サービス・ウィザードを使用して構成するか、デプロイメント後に WebSphere Application Server 管理コンソールを使用して構成します。

**注:** 双方向スクリプトを使用してこれらのアダプター・タイプ・プロパティーのいずれかを設定する場合、そのプロパティーについて入力された双方向スクリプトの形式を識別できる値を設定する必要があります。

アダプター・タイプ・プロパティーおよびその目的を以下の表に示します。各プロパティーの完全な説明は、表に続くセクションで説明します。続くセクションのプロパティーの詳細表を読む方法については、『プロパティー詳細についてのガイド』を参照してください。

表9. アダプター・タイプ・プロパティー

プロパティー名		説明
ウィザード内	管理コンソール内	
141 ページの『「説明」プロパティー (Description)』	Description	アダプターの説明。
141 ページの『「表示名」プロパティー (DisplayName)』	DisplayName	アダプター表示名。
141 ページの『「ID」プロパティー (ID)』	ID	アダプター・タイプの ID。
141 ページの『「ベンダー」プロパティー (Vendor)』	Vendor	アダプターを提供するベンダーの名前。

表9. アダプター・タイプ・プロパティ (続き)

『「バージョン」プロパティ (Version)』	Version	アダプターのバージョン。
--------------------------	---------	--------------

### 「説明」プロパティ (Description)

アダプターの説明。

表10. 「説明」プロパティ特性

必須	はい
デフォルト	IBM WebSphere Adapter for FTP
プロパティ・タイプ	String

### 「表示名」プロパティ (DisplayName)

アダプター表示名。

表11. 「表示名」プロパティ特性

必須	はい
デフォルト	IBM WebSphere Adapter for FTP
プロパティ・タイプ	String

### 「ID」プロパティ (ID)

アダプター・タイプの ID。

表12. 「ID」プロパティ特性

必須	はい
デフォルト	FTP
プロパティ・タイプ	String

### 「ベンダー」プロパティ (Vendor)

アダプターを提供するベンダーの名前。

表13. 「ベンダー」プロパティ特性

必須	はい
デフォルト	IBM
プロパティ・タイプ	String

### 「バージョン」プロパティ (Version)

アダプターのバージョン。

表14. 「バージョン」プロパティ特性

必須	はい
----	----

表 14. 「バージョン」プロパティー特性 (続き)

デフォルト	6.1.0
プロパティー・タイプ	String

## リソース・アダプター・プロパティー

リソース・アダプター・プロパティーは、ビジネス・オブジェクトのネーム・スペースの指定など、アダプターの一般的な操作を制御します。リソース・アダプター・プロパティーは、アダプターの構成時に、外部サービス・ウィザードを使用して設定します。アダプターのデプロイ後、管理コンソールを使用して、プロパティーを変更できます。

以下に示すロギングおよびトレースのプロパティーは、バージョン 6.1.0 には必要なくなりましたが、旧バージョンとの互換性を維持するためにサポートされています。

- LogFileMaxSize
- LogFileName
- LogNumberOfFiles
- TraceFileMaxSize
- TraceFileName
- TraceNumberOfFiles

以下の表は、リソース・アダプター・プロパティーおよびその目的をリストしています。各プロパティーの完全な説明は、表に続くセクションで説明します。続くセクションのプロパティーの詳細表を読む方法については、『プロパティー詳細についてのガイド』を参照してください。

表 15. Adapter for FTP 用のリソース・アダプター・プロパティー

プロパティー名		説明
ウィザード内	管理コンソール内	
アダプター ID	AdapterID	CEI イベントおよび PMI イベントのアダプター・インスタンスをロギングおよびトレースを基準にして識別します。
143 ページの『EISEncoding (EISEncoding)』	EISEncoding	FTP サーバーのエンコード。
(なし)	enableHASupport	このプロパティーは変更しないでください。
(なし)	LogFileMaxSize	旧バージョンとの互換性を維持するためにサポートされています。
(なし)	ログ・ファイル名	旧バージョンとの互換性を維持するためにサポートされています。
(なし)	ログ・ファイル数	旧バージョンとの互換性を維持するためにサポートされています。
(なし)	TraceFileMaxSize	旧バージョンとの互換性を維持するためにサポートされています。
(なし)	トレース・ファイル名	旧バージョンとの互換性を維持するためにサポートされています。

表 15. Adapter for FTP 用のリソース・アダプター・プロパティ (続き)

プロパティ名		説明
ウィザード内	管理コンソール内	
(なし)	トレース・ファイル数	旧バージョンとの互換性を維持するためにサポートされています。

## ロギングおよびトレースで使用するアダプター ID (AdapterID)

このプロパティは、アダプターの特定のデプロイメント (インスタンス) を識別する場合に使用します。

表 16. 「ロギングおよびトレースで使用するアダプター ID」の詳細

必須	はい
デフォルト	CWYFT_FTPFile
プロパティ・タイプ	String
使用法	このプロパティを使用して、PMI イベントのアダプター・インスタンスを識別します。アダプターのインスタンスを複数デプロイする場合は、このプロパティをアダプターのインスタンスごとに固有な値に設定します。  Inbound 処理の場合、このプロパティはリソース・アダプター・プロパティから取り出します。Outbound 処理の場合、管理接続ファクトリー・プロパティから取り出します。
グローバル化	はい
BIDI 対応	いいえ

## EISEncoding (EISEncoding)

このプロパティは、FTP サーバーのエンコードを指定します。FTP サーバーとの通信中の制御接続のエンコードを設定します。FTP サーバーのディレクトリーまたはファイル名に国際化された文字が含まれている場合、このプロパティを設定します。

表 17. EISEncoding 特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
例	UTF-8、ISO-8859-1

## 高可用性サポートを使用可能にする (enableHASupport)

このプロパティは変更しないでください。必ず true に設定しておく必要があります。

## ログ・ファイルの最大サイズ (LogFileMaxSize)

このプロパティでは、ログ・ファイルのサイズをキロバイト単位で指定します。

表 18. 「ログ・ファイルの最大サイズ」の詳細

必須	いいえ
デフォルト	0
プロパティー・タイプ	Integer
使用法	ログ・ファイルがその最大サイズに到達すると、アダプターは新規のログ・ファイルの使用を開始します。ファイル・サイズを 0 と指定するか、最大サイズを指定しなかった場合、そのファイルに最大サイズはありません。
グローバル化	はい
BIDI 対応	いいえ

### ログ・ファイル名 (LogFilename)

このプロパティーでは、ログ・ファイルの絶対パス名を指定します。

表 19. 「ログ・ファイル名」の詳細

必須	いいえ
デフォルト	デフォルト値なし
プロパティー・タイプ	String
使用法	このプロパティーは非推奨です。
グローバル化	はい
BIDI 対応	はい

### ログ・ファイルの数 (LogNumberOfFiles)

このプロパティーでは、ログ・ファイルの数を指定します。

表 20. 「ログ・ファイルの数」の詳細

必須	いいえ
デフォルト	1
プロパティー・タイプ	Integer
使用法	ログ・ファイルが最大サイズに到達すると、アダプターは別のログ・ファイルを使い始めます。値を指定しなかった場合、アダプターはログ・ファイルを 1 つ作成します。
グローバル化	はい
BIDI 対応	いいえ

### トレース・ファイルの最大サイズ (TraceFileMaxSize)

このプロパティーでは、トレース・ファイルのサイズをキロバイト単位で指定します。

表 21. 「トレース・ファイルの最大サイズ」の詳細

必須	いいえ
デフォルト	0

表 21. 「トレース・ファイルの最大サイズ」の詳細 (続き)

プロパティ・タイプ	Integer
使用法	値を指定しなかった場合は、トレース・ファイルに最大サイズが設定されません。
グローバル化	はい
BIDI 対応	いいえ

### トレース・ファイル名 (TraceFilename)

このプロパティでは、トレース・ファイルの絶対パスを指定します。

表 22. 「トレース・ファイル名」の詳細

必須	いいえ
デフォルト	デフォルト値なし
計測単位	キロバイト
プロパティ・タイプ	String
使用法	このプロパティは非推奨です。
グローバル化	はい
BIDI 対応	はい

### トレース・ファイルの数 (TraceNumberOfFiles)

このプロパティでは、使用するトレース・ファイルの数を指定します。トレース・ファイルが最大サイズに到達すると、アダプターは別のトレース・ファイルを使い始めます。

表 23. 「トレース・ファイルの数」の詳細

必須	いいえ
デフォルト	1
プロパティ・タイプ	Integer
使用法	値を指定しなかった場合、アダプターはトレース・ファイルを 1 つ使用します。
グローバル化	はい
BIDI 対応	いいえ

## 管理 (J2C) 接続ファクトリー・プロパティ

管理接続ファクトリー・プロパティは、FTP サーバーとの Outbound 接続インスタンスを作成するために、アダプターがランタイムに使用します。

管理接続ファクトリー・プロパティを設定するには、外部サービス・ウィザードを使用します。変更する場合には、WebSphere Integration Developer のアセンブリ・エディターを使用することもできますし、デプロイメント後に WebSphere Process Server 管理コンソールを通じて行うこともできます。

以下の表は、管理接続ファクトリー・プロパティをリストしています。各プロパティの完全な説明は、表に続くセクションで説明します。続くセクションのプロパティの詳細表を読む方法については、『プロパティ詳細についてのガイド』を参照してください。

注: 外部サービス・ウィザードは、これらのプロパティを管理接続ファクトリー・プロパティとして参照し、WebSphere Process Server 管理コンソールは、(j2C) 接続ファクトリー・プロパティとして参照します。

表 24. 管理接続ファクトリー・プロパティ

プロパティ名		説明
ウィザード内	管理コンソール内	
147 ページの『「カスタム・パーサー・クラス名」プロパティ (CustomParserClassName)』	CustomParserClassName	ls -l の出力を解析するために使用されるカスタム・パーサーの完全修飾クラス名を指定します。
148 ページの『「デフォルト・ターゲット・ファイル名」プロパティ (Filename)』	Filename	Outbound Create 操作時に使用するファイル名を指定します。
148 ページの『「ディレクトリ」プロパティ (OutputDirectory)』	OutputDirectory	FTP サーバーの出力ディレクトリーを指定します。
148 ページの『「FTP サーバーで使用されるエンコード」プロパティ (EISEncoding)』	EISEncoding	FTP サーバーのエンコードを指定します。
149 ページの『「ホスト名」プロパティ (HostName)』	HostName	FTP サーバーのホスト名を指定します。
149 ページの『「ホスト名」プロパティ (SecondServerHostName)』	HostName	2 次 FTP サーバーのホスト名を指定します。
149 ページの『「ホスト名」プロパティ (SocksProxyHost)』	SocksProxyHost	プロキシ・サーバーとして使用されるワークステーションの名前を指定します。
149 ページの『「パスワード」プロパティ (Password)』	Password	FTP サーバーへの接続特権、および FTP 操作の実行特権を持つパスワードを指定します。
150 ページの『「パスワード」プロパティ (SecondServerPassword)』	SecondServerPassword	サーバー間のファイル転送の Outbound 操作時にファイルの転送先となる 2 次 FTP サーバーのパスワードを指定します。
150 ページの『「パスワード」プロパティ (SocksProxyPassword)』	SocksProxyPassword	プロキシ・サーバーを認証するためのパスワードを指定します。
150 ページの『「ポート番号」プロパティ (PortNumber)』	PortNumber	FTP サーバーのポート番号を指定します。



表 24. 管理接続ファクトリー・プロパティ (続き)

150 ページの『「ポート番号」プロパティ (SecondServerPortNumber)』	SecondServerPortNumber	2 次 FTP サーバーのポート番号を指定します。
151 ページの『「ポート番号」プロパティ (SocksProxyPort)』	SocksProxyPort	プロキシ・サーバーのポート番号を指定します。
151 ページの『「プロトコル」プロパティ (Protocol)』	Protocol	通常またはセキュア FTP 接続のいずれかを指定します。
151 ページの『「プロトコル」プロパティ (SecondServerProtocol)』	SecondServerProtocol	2 次サーバーに対して通常またはセキュア FTP 接続のいずれかを指定します。
152 ページの『「2 次サーバー・ディレクトリー」プロパティ (SecondServerDirectory)』	SecondServerDirectory	Outbound の ServerToServerFileTransfer 操作の実行対象となる、2 次 FTP サーバーのディレクトリー・パスを指定します。
152 ページの『「シーケンス・ファイル」プロパティ (FileSequenceLog)』	FileSequenceLog	Outbound の Create 処理用のシーケンス番号が保管されるファイルの絶対パスを指定します。
153 ページの『「ステージング・ディレクトリー」プロパティ (StagingDirectory)』	StagingDirectory	ファイルが最初に作成される先のディレクトリーを指定します。
153 ページの『「ユーザー名」プロパティ (SecondServerUserName)』	SecondServerUserName	サーバー間のファイル転送の Outbound 操作時にファイルの転送先となる 2 次 FTP サーバーのユーザー名を指定します。
153 ページの『「ユーザー名」プロパティ (SocksProxyUserName)』	SocksProxyUserName	プロキシ・サーバーを認証するために使用するユーザー名を指定します。
153 ページの『「ユーザー名」プロパティ (UserName)』	Username	ユーザーの名前を指定します。

### 「カスタム・パーサー・クラス名」プロパティ (CustomParserClassName)

ls -l の出力を解析するために使用されるカスタム・パーサーの完全修飾クラス名。ls -l の出力が標準出力から外れる場合にのみ使用されます。

表 25. 「カスタム・パーサー・クラス名」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	いいえ

## 「デフォルト・ターゲット・ファイル名」プロパティ (Filename)

Outbound Create 操作時に使用するファイル名。この値と FtpUrl の組み合わせでシーケンスが決まります。

例:

FtpUrl = ftp://localhost:21/out および Filename = Customer.txt の場合、出力ファイルは Customer.1.txt、Customer.2.txt、Customer.3.txt などのようになります。

表 26. 「デフォルト・ターゲット・ファイル名」プロパティ特性

必須	はい
デフォルト	はい
プロパティ・タイプ	String
国際化されるかどうか	いいえ

## 「ディレクトリー」プロパティ (OutputDirectory)

Outbound 操作が実行される FTP サーバーの出力ディレクトリー。

表 27. 「ディレクトリー」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

## 「FTP サーバーで使用されるエンコード」プロパティ (EISEncoding)

FTP サーバーのエンコード。この値は、FTP サーバーとの制御接続のエンコードを設定する場合に使用します。

- アダプター・レベルの EISEncoding と MCF レベルの EISEncoding の両方を設定しない場合 (両方とも null)、FTP サーバーとの通信中は制御接続で何も設定されません。
- アダプター・レベルの EISEncoding を設定し、MCF レベルの EISEncoding を設定しない場合、アダプター・レベルの値は、FTP サーバーとの通信中に制御接続で設定されます。これは、複数の MCF を使用し、同じエンコードが設定されている場合に便利です。この場合は、すべての接続で制御接続のエンコードが同じになるよう、値をアダプター・レベルで設定します。
- アダプター・レベルの EISEncoding を設定しない場合で、MCF レベルの EISEncoding を設定する場合、MCF レベルの値は、FTP サーバーとの通信中に制御接続で設定されます。値は MCF レベルであるため、これが適用されるのは、その MCF のみです。
- アダプター・レベルの EISEncoding と MCF レベルの EISEncoding の両方を設定する場合、MCF レベルの値が優先されます。

この属性には、Java がサポートする任意のエンコードを指定できます。

表 28. 「FTP サーバーで使用されるエンコード」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	いいえ

### 「ホスト名」プロパティ (HostName)

Outbound 操作時に接続の確立先となる FTP サーバーのホスト名。

表 29. 「ホスト名」プロパティ特性

必須	はい
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

### 「ホスト名」プロパティ (SecondServerHostName)

Outbound 操作時に接続の確立先となる 2 次 FTP サーバーのホスト名。

表 30. 「ホスト名」プロパティ特性

必須	はい
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

### 「ホスト名」プロパティ (SocksProxyHost)

アダプターの要求を FTP サーバーに送信する際に経由するプロキシ・サーバーとして使用されるワークステーションのホスト名。

表 31. 「ホスト名」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

### 「パスワード」プロパティ (Password)

FTP サーバーへの接続特権、および FTP 操作の実行特権を持つユーザーのパスワード。FtpUrl プロパティで指定した URL にパスワードが含まれている場合は、この属性に値を指定する必要はありません。

表 32. 「パスワード」プロパティ特性

必須	いいえ
デフォルト	なし

表 32. 「パスワード」プロパティ特性 (続き)

プロパティ・タイプ	String
国際化されるかどうか	はい

### 「パスワード」プロパティ (SecondServerPassword)

サーバー間のファイル転送の Outbound 操作時にファイルの転送先となる 2 次 FTP サーバーのパスワード。

表 33. 「パスワード」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

### 「パスワード」プロパティ (SocksProxyPassword)

プロキシ・サーバーを認証するためのパスワード。

表 34. 「パスワード」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

### 「ポート番号」プロパティ (PortNumber)

Outbound 操作時に接続の確立先となる FTP サーバーのポート番号。

表 35. 「ポート番号」プロパティ特性

必須	はい
デフォルト	21
プロパティ・タイプ	Integer
国際化されるかどうか	いいえ

### 「ポート番号」プロパティ (SecondServerPortNumber)

Outbound 操作時に接続の確立先となる 2 次 FTP サーバーのポート番号。

表 36. 「ポート番号」プロパティ特性

必須	はい
デフォルト	21
プロパティ・タイプ	Integer
国際化されるかどうか	いいえ

## 「ポート番号」プロパティ (SocksProxyPort)

アダプターの要求を FTP サーバーに送信する際に経由するプロキシ・サーバーのポート番号。

表 37. 「ポート番号」プロパティ特性

必須	いいえ
デフォルト	1080
プロパティ・タイプ	Integer
国際化されるかどうか	いいえ

## 「プロトコル」プロパティ (Protocol)

確立する接続が通常の FTP 接続か、またはセキュア FTP 接続かを定めるプロトコル。

例:

通常接続: FTP

セキュア接続: FTPS

表 38. 「プロトコル」プロパティ特性

必須	はい
デフォルト	FTP
プロパティ・タイプ	String
国際化されるかどうか	いいえ

## 「プロトコル」プロパティ (SecondServerProtocol)

確立する 2 次接続が通常の FTP 接続か、またはセキュア FTP 接続かを定めるプロトコル。

例:

通常接続: FTP

セキュア接続: FTPS

表 39. 「プロトコル」プロパティ特性

必須	はい
デフォルト	FTP
プロパティ・タイプ	String
国際化されるかどうか	いいえ

## 「2 次サーバー・ディレクトリー」プロパティー (SecondServerDirectory)

Outbound の ServerToServerFileTransfer 操作の実行対象となる、2 次 FTP サーバーのディレクトリー・パス。

ディレクトリー・パスを指定する際の構文: ftp://  
[UserId:password@]FTPserver[:port]Directory  
ForSecondServer

以下の情報も指定することができます。

- 2 次 FTP サーバーへの接続特権、および FTP 操作の実行特権を持つユーザー名およびパスワード。ここで指定しない場合は、SecondServerUsername および SecondServerPassword プロパティーで指定する必要があります。
- FTP ポート。ここで指定しない場合、アダプターはデフォルトの FTP ポートを使用します。
- リモート・イベント・ディレクトリー。ここで指定しない場合、アダプターは FTP サーバーで接続が確立されたディレクトリーにファイルを転送します。

表 40. 「2 次サーバー・ディレクトリー」プロパティー特性

必須	いいえ
デフォルト	なし
プロパティー・タイプ	String
国際化されるかどうか	はい

## 「シーケンス・ファイル」プロパティー (FileSequenceLog)

Outbound の Create 処理用のシーケンス番号が保管されるファイルの絶対パスを指定します。

FileSequenceLog プロパティーが指定された場合、アダプターは固有のシーケンス番号を生成して、Create 操作の処理時にファイル名に挿入します。

シーケンス番号は、複数のアダプターが再始動するたびに増え続けます。

シーケンス番号は、以下の形式でファイル名に挿入されます。

filename.number.extension

例: Customer.3.txt

FileSequenceLog プロパティーが指定されないか、無効な値を含む場合、シーケンス番号は生成されません。

表 41. 「シーケンス・ファイル」プロパティー特性

必須	いいえ
デフォルト	なし
プロパティー・タイプ	String
国際化されるかどうか	いいえ

## 「ステージング・ディレクトリー」プロパティ (StagingDirectory)

Outbound Create 操作時にファイルが最初に作成される先のディレクトリー。作成後に、ファイルは DirectoryPath プロパティで指定されたディレクトリーに移動されます。ステージング・ディレクトリーは、Append 操作および Overwrite 操作でも使用されます。その場合、指定されたファイルは、まず、StagingDirectory (存在する場合) にコピーされ、内容が付加または上書きされてから、元の指定されたディレクトリーに戻されます。StagingDirectory が存在しない場合、この操作は実際の必要なディレクトリー内で実行されます。ファイルの付加および更新時にファイルへの書き込みが行われているにもかかわらず、他のユーザーがそのファイルを読み取ってしまった場合に、ファイルの書き込み競合が発生することがあります。ステージング・ディレクトリーで作業すると、この競合が避けられるという利点があります。

表 42. 「ステージング・ディレクトリー」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

## 「ユーザー名」プロパティ (SecondServerUserName)

サーバー間のファイル転送の Outbound 操作時にファイルの転送先となる 2 次 FTP サーバーのユーザー名。

表 43. 「ユーザー名」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

## 「ユーザー名」プロパティ (SocksProxyUserName)

プロキシ・サーバーを認証するためのユーザー名。

表 44. 「ユーザー名」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

## 「ユーザー名」プロパティ (UserName)

FTP サーバーへの接続特権、および FTP 操作の実行特権を持つユーザーの名前。FtpUrl プロパティで指定した URL にユーザー名が含まれている場合は、この属性に値を指定する必要はありません。

表 45. 「ユーザー名」プロパティ特性

必須	いいえ
----	-----

表 45. 「ユーザー名」プロパティ特性 (続き)

デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

## 対話仕様プロパティ

対話仕様プロパティは、操作の対話を制御します。外部サービス・ウィザードは、アダプターの構成時に対話仕様プロパティを設定します。これらのプロパティには、変更可能なものもありますが、すべてを変更することはできません。しかし、Outbound 操作のプロパティのいくつかは、ユーザーが変更できます。アセンブリー・エディターを使用して、インポートのメソッド・バインディングに存在するこれらのプロパティを変更します。

以下の表は、対話仕様プロパティをリストしています。各プロパティの完全な説明は、表に続くセクションで説明します。続くセクションのプロパティの詳細表を読む方法については、『プロパティ詳細についてのガイド』を参照してください。

表 46. 対話仕様プロパティ

プロパティ名		説明
ウィザード内	管理コンソール内	
156 ページの『「retrieve 操作のリモート・アーカイブ・ディレクトリー」プロパティ (ArchiveDirectory)ForRetrieve』	ArchiveDirectory ForRetrieve	アダプターは、ファイルを Retrieve 操作中に削除される前に、このフォルダーにオプションでアーカイブする。
156 ページの『「ファイルが存在しない場合に新規ファイルを作成する」プロパティ (CreateFileIfNotExists)』	CreateFileIfNotExists	ファイルが FTP サーバーに存在しない場合、Append および Overwrite 操作中にこのプロパティが true に設定されていると、アダプターはファイルを作成する。
157 ページの『「FTP サーバー接続モード」プロパティ (DataConnectionMode)』	DataConnectionMode	FTP サーバーがファイル転送時に使用するデータ接続モード。
(なし)	DeleteOnRetrive	このプロパティが true に設定されている場合、アダプターは、ファイルを取得した後、FTP サーバーからファイルを削除する。
158 ページの『「FTP システム上のリモート・ディレクトリー」プロパティ (DirectoryPath)』	DirectoryPath	Outbound 操作を実行する必要がある FTP サーバー上のディレクトリーの絶対パス。
158 ページの『「ファイル内容のエンコード」プロパティ (FileContentEncoding)』	FileContentEncoding	ファイルへの書き込み中に使用されるエンコード。
158 ページの『「ローカル・ディレクトリー内のファイル」プロパティ (FileInLocalDirectory)』	FileInLocalDirectory	Create 操作時に true に設定されている場合、アダプター・マシンのローカル・ディレクトリー・パスからファイル内容が取り出される。



表 46. 対話仕様プロパティ (続き)

159 ページの『「デフォルト・ターゲット・ファイル名」プロパティ (Filename)』	Filename	DirectoryPath プロパティで指定されたディレクトリ内のファイルの名前。
159 ページの『「ファイル転送タイプ」プロパティ (FileTransferType)』	FileTransferType	Outbound 操作時に使用されるファイル転送タイプ。
(なし)	GenerateUniqueFile	アダプターは、このプロパティが true に設定されている場合、固有ファイル名を作成する。
160 ページの『「ホスト名」プロパティ (SecondServerHostName)』	SecondServerHostName	2 次 FTP サーバーのホスト名。
160 ページの『「ファイル内容にビジネス・オブジェクト区切り文字を組み込む」プロパティ (IncludeEndBODelimiter)』	IncludeEndBODelimiter	この値がファイルの内容に付加される。
160 ページの『「create 操作のローカル・アーカイブ・ディレクトリ」プロパティ (LocalArchiveDirForCreate)』	LocalArchiveDirForCreate	Create 操作時に LocalArchivingEnabledForCreate が true に設定されている場合、このディレクトリのローカル・ワークステーションにファイルが保存される。
160 ページの『「create 操作のローカル・ディレクトリへのファイルのアーカイブ」プロパティ』	LocalArchivingEnabledForCreate	true に設定されている場合、ファイルは Create 操作時にローカル・ワークステーションに保管される。
161 ページの『「ローカル・ディレクトリ」プロパティ (LocalDirectoryPath)』	LocalDirectoryPath	このディレクトリからファイルが取り出される。
161 ページの『「ポート番号」プロパティ (SecondServerPortNumber)』	SecondServerPortNumber	2 次 FTP サーバーのポート番号。
161 ページの『「プロトコル」プロパティ (SecondServerProtocol)』	SecondServerProtocol	接続を通常の FTP、セキュア FTP のいずれにするのかを決める。
162 ページの『「スクリプト・ファイル・パラメータ (Script File Parameters)」プロパティ (ScriptFileParameters)』	ScriptFileParameters	FTP スクリプト・ファイルに必要なパラメータ。
156 ページの『「ディレクトリ」プロパティ (SecondServerDirectory)』	SecondServerDirectory	ServerToServerFileTransfer 操作時の 2 次 FTP サーバーのディレクトリ・パス。
162 ページの『「パスワード」プロパティ (SecondServerPassword)』	SecondServerPassword	ServerToServerFileTransfer 操作時の 2 次 FTP サーバーのパスワード。
162 ページの『「ユーザー名」プロパティ (SecondServerUsername)』	SecondServerUsername	ServerToServerFileTransfer 操作時の 2 次 FTP サーバーのユーザー名。

表 46. 対話仕様プロパティ (続き)

163 ページの『「ファイル内容を分割するための基準の指定」プロパティ (SplitCriteria)』	SplitCriteria	イベント・ファイル内のビジネス・オブジェクトを分割する区切り文字。
163 ページの『「分割関数クラス名」プロパティ (SplittingFunctionClassName)』	SplittingFunctionClassName	ファイル分割を有効にするために使用するクラス・ファイルの完全修飾クラス名。
164 ページの『「ステージング・ディレクトリー」プロパティ (StagingDirectory)』	StagingDirectory	ファイルはこのディレクトリーに最初に作成される。

### 「retrieve 操作のリモート・アーカイブ・ディレクトリー」プロパティ (ArchiveDirectory ForRetrieve)

Outbound Retrieve 操作時に、アダプターはファイルが削除される前に、このフォルダーにオプションでファイルをアーカイブします。アーカイブ・ディレクトリーが存在する必要があります。

表 47. 「retrieve 操作のリモート・アーカイブ・ディレクトリー」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

### 「ファイルが存在しない場合に新規ファイルを作成する」プロパティ (CreateFileIfNotExists)

Outbound Append および Overwrite 操作中に、ファイルが FTP サーバーに存在しない場合、このプロパティが true に設定されていると、アダプターはファイルを作成します。このプロパティが false でファイルが存在しない場合、アダプターはエラーを送信します。

表 48. 「ファイルが存在しない場合に新規ファイルを作成する」プロパティ特性

必須	いいえ
デフォルト	false
プロパティ・タイプ	Boolean
国際化されるかどうか	いいえ

### 「ディレクトリー」プロパティ (SecondServerDirectory)

サーバー間のファイル転送の Outbound 操作の実行対象の 2 次 FTP サーバーのディレクトリー。

ディレクトリー・パスを指定する構文: ftp://  
[UserId:password@]FTPserver[:port]DirectoryForSecondServer

以下の情報も指定することができます。

- 2 次 FTP サーバーへの接続特権、および FTP 操作の実行特権を持つユーザー名およびパスワード。ここで指定しない場合は、SecondServerUsername および SecondServerPassword プロパティで指定する必要があります。
- FTP ポート。ここで指定しない場合、アダプターはデフォルトの FTP ポートを使用します。
- リモート・イベント・ディレクトリー。ここで指定しない場合、アダプターは FTP サーバーで接続が確立されたディレクトリーにファイルを転送します。

表 49. 「ディレクトリー」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

### 「FTP サーバー接続モード」プロパティ (DataConnectionMode)

FTP サーバーがファイル転送時に使用するデータ接続モード。active または passive のいずれかの設定値を指定します。この値は、ファイル転送が行われる場合にのみ使用されます。このプロパティは、サーバー間のファイル転送の Outbound 操作の実行時は使用されません。

表 50. 「FTP サーバー接続モード」プロパティ特性

必須	いいえ
デフォルト	active
プロパティ・タイプ	String
使用可能な値	active または passive
国際化されるかどうか	いいえ

### 「デフォルト・オブジェクト名」プロパティ (DefaultObjectName)

データ形式変換に必要な ASI 情報にアクセスするために、データ形式変更フレームワークによって使用されるビジネス・オブジェクトの名前。

アダプターは、RetrieveContentType 値に対応する DataBinding クラス名をビジネス・オブジェクトの注釈から取得します。

表 51. 「デフォルト・オブジェクト名」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい
例	アダプターが使用する汎用ビジネス・オブジェクトは FTPFile です。

### DeleteOnRetrieve

Outbound Retrieve 操作時に、このプロパティが true に設定されている場合、アダプターは、ファイルを取得した後、FTP サーバーからファイルを削除します。

表 52. 「retrieve 操作後のファイルの削除」プロパティ特性

必須	いいえ
デフォルト	false
プロパティ・タイプ	Boolean
国際化されるかどうか	いいえ

### 「FTP システム上のリモート・ディレクトリー」プロパティ (DirectoryPath)

ExecuteFTPScript 以外のすべての操作用に Outbound 操作を実行する必要がある FTP サーバー上のディレクトリーの絶対パス、または ExecuteFTPScript 操作のみローカル・アダプター・ワークステーション上のディレクトリー・パス。このディレクトリーが存在していなければなりません。

表 53. 「FTP システム上のリモート・ディレクトリー」プロパティ特性

必須	はい
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

### 「ファイル内容のエンコード」プロパティ (FileContentEncoding)

ファイルへの書き込み中に使用されるエンコード。このプロパティを指定しない場合、アダプターは特定のエンコードを使用せずに読み取りを行おうとします。Java がサポートする任意のエンコード・セットを指定できます。

表 54. 「ファイル内容のエンコード」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	いいえ

### 「ローカル・ディレクトリー内のファイル」プロパティ (FileInLocalDirectory)

Outbound の create 操作時に、このプロパティが true に設定されていると、ファイルの内容はビジネス・オブジェクト内で使用できません。ファイルは、アダプター・ワークステーションのローカル・ディレクトリーから取り出されます。Outbound の retrieve 操作時に、このプロパティが true に設定されていると、ファイルの内容はビジネス・オブジェクトの一部として J2EE アプリケーションに送信されません。ファイルは、アダプター・ワークステーションのローカル・ディレクトリーに保管されます。

表 55. 「ローカル・ディレクトリー内のファイル」プロパティ特性

必須	いいえ
デフォルト	false

表 55. 「ローカル・ディレクトリー内のファイル」プロパティー特性 (続き)

プロパティー・タイプ	Boolean
国際化されるかどうか	いいえ

### 「デフォルト・ターゲット・ファイル名」プロパティー (Filename)

Outbound 操作を実行する DirectoryPath プロパティーで指定されたディレクトリー内のファイルの名前。この値は、LIST 以外のすべての Outbound 操作に必要です。

表 56. 「デフォルト・ターゲット・ファイル名」プロパティー特性

必須	いいえ
デフォルト	なし
プロパティー・タイプ	String
国際化されるかどうか	はい

### 「ファイル転送タイプ」プロパティー (FileTransferType)

Outbound 操作時に使用されるファイル転送タイプ。ASCII または binary のいずれかの設定値を指定します。

表 57. 「ファイル転送タイプ」プロパティー特性

必須	いいえ
デフォルト	binary
プロパティー・タイプ	String
国際化されるかどうか	いいえ

### 「固有ファイルの生成」プロパティー (GenerateUniqueFile)

Outbound Create、Append、および Overwrite 操作時に、このプロパティーが true に設定されている場合、アダプターは固有ファイル名を作成します。このプロパティーが true に設定されている場合、アダプターは、Filename プロパティーに設定されているあらゆる値を無視します。

また、Append および Overwrite 操作時に、このプロパティーおよび CreateFileIfNotExists プロパティーの両方が true に設定されている場合にも、アダプターは固有ファイル名を作成します。

**注:** アダプターでは、GenerateUniqueFile と StagingDirectory の両方のオプションを同時にはサポートしていません。

表 58. 「固有ファイルの生成」プロパティー特性

必須	いいえ
デフォルト	false
プロパティー・タイプ	Boolean
国際化されるかどうか	いいえ
制約事項	この機能を使用するには、FTP サーバーが RFC1123 をサポートしている必要があります。

## 「ホスト名」プロパティ (SecondServerHostName)

Outbound 操作時に接続の確立先となる 2 次 FTP サーバーのホスト名。

表 59. 「ホスト名」プロパティ特性

必須	はい
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

## 「ファイル内容にビジネス・オブジェクト区切り文字を組み込む」プロパティ (IncludeEndBODelimiter)

この値がファイルの内容に付加される。Outbound の create、append、および overwrite 操作時に使用されます。

表 60. 「ファイル内容にビジネス・オブジェクト区切り文字を組み込む」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

## 「create 操作のローカル・アーカイブ・ディレクトリー」プロパティ (LocalArchiveDirForCreate)

Outbound の create 操作時に、ファイルの内容が、ビジネス・オブジェクトの一部として提供され、LocalArchivingEnabledForCreate が true に設定されている場合は、ファイルはローカル・ワークステーションのこのディレクトリーに保管されます。

表 61. 「create 操作のローカル・アーカイブ・ディレクトリー」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

## 「create 操作のローカル・ディレクトリーへのファイルのアーカイブ」プロパティ

Outbound の create 操作時に、ファイルの内容が、J2EE アプリケーションからビジネス・オブジェクトの一部として提供され、このプロパティが true に設定されている場合は、ファイルはローカル・ワークステーションの LocalArchiveDirForCreate ディレクトリーに保管されてから、Outbound 操作が実行されます。

表 62. 「create 操作のローカル・ディレクトリーへのファイルのアーカイブ」プロパティ特性

必須	いいえ
----	-----

表 62. 「create 操作のローカル・ディレクトリーへのファイルのアーカイブ」プロパティ特性 (続き)

デフォルト	false
プロパティ・タイプ	Boolean
国際化されるかどうか	いいえ

### 「ローカル・ディレクトリー」プロパティ (LocalDirectoryPath)

Outbound の create 操作時に、FileInLocalDirectory プロパティが true に設定されていると、ファイルの内容はビジネス・オブジェクト内で使用できません。その代わりに、このディレクトリーからファイルが取り出されます。Outbound の retrieve 操作時に、FileInLocalDirectory プロパティが true に設定されていると、ファイルの内容はビジネス・オブジェクトの一部として送信できません。ファイルはこのディレクトリーに保管されます。

表 63. 「ローカル・ディレクトリー」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

### 「ポート番号」プロパティ (SecondServerPortNumber)

Outbound 操作時に接続の確立先となる 2 次 FTP サーバーのポート番号。

表 64. 「ポート番号」プロパティ特性

必須	はい
デフォルト	21
プロパティ・タイプ	Integer
国際化されるかどうか	いいえ

### 「プロトコル」プロパティ (SecondServerProtocol)

確立する接続が通常の FTP 接続か、またはセキュア FTP 接続かを定めるプロトコル。

例:

通常接続: FTP

セキュア接続: FTPS

表 65. 「プロトコル」プロパティ特性

必須	はい
デフォルト	FTP
プロパティ・タイプ	String
国際化されるかどうか	いいえ

## 「コンテンツ・タイプの取得 (Retrieve Content Type)」プロパティ ー (RetrieveContentType)

ネイティブ・フォーマットからビジネス・オブジェクトへの変換中に、正しいデータ・ハンドラーを呼び出すためにデータ形式変更フレームワークに送信される、取得済みファイルの内容/MIME タイプ。 PassThrough 時に、この値は NULL に設定されます。

表 66. 「コンテンツ・タイプの取得 (Retrieve Content Type)」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	いいえ
例	text/xml、text/delimited、text/namevalue

## 「スクリプト・ファイル・パラメーター (Script File Parameters)」 プロパティ (ScriptFileParameters)

Outbound ExecuteFTPScript 操作時に、FTP スクリプト・ファイルに必要なパラメーターがこのプロパティに設定されます。ランタイム時に、アダプターはパラメーターをこれらの値に置換します。

表 67. 「スクリプト・ファイル・パラメーター (Script File Parameters)」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

## 「パスワード」プロパティ (SecondServerPassword)

サーバー間のファイル転送の Outbound 操作時にファイルの転送先となる 2 次 FTP サーバーのパスワード。

表 68. 「パスワード」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

## 「ユーザー名」プロパティ (SecondServerUsername)

サーバー間のファイル転送の Outbound 操作時にファイルの転送先となる 2 次 FTP サーバーのユーザー名。

表 69. 「ユーザー名」プロパティ特性

必須	いいえ
デフォルト	なし



表 69. 「ユーザー名」プロパティ特性 (続き)

プロパティ・タイプ	String
国際化されるかどうか	はい

## 「ファイル内容を分割するための基準の指定」プロパティ (SplitCriteria)

SplittingFunctionClassName プロパティの値に基づいた各種の値を指定する。

- SplittingFunctionClassName プロパティで、区切り文字に基づいてファイルを分割することを指定する場合、SplitCriteria にはイベント・ファイル内のビジネス・オブジェクトを分割する区切り文字を指定します。
- SplittingFunctionClassName をサイズに基づいた分割を行う値に設定する場合、SplitCriteria プロパティには、バイト単位のサイズを表す有効な数値を指定します。
  - イベント・ファイルのサイズがこの値より大きい場合、アダプターはファイルをこのサイズのチャンクに分割し、各チャンクが送られます。
  - イベント・ファイルのサイズがこの値より小さい場合、イベント・ファイル全体が送られます。SplitCriteria=0 の場合、チャンクへの分割は無効です。

表 70. 「ファイル内容を分割するための基準の指定」プロパティ特性

必須	いいえ
デフォルト	0
プロパティ・タイプ	String
国際化されるかどうか	はい

## 「分割関数クラス名」プロパティ (SplittingFunctionClassName)

ファイル分割を有効にするために使用するクラス・ファイルの完全修飾クラス名を指定する。以下の 2 つの値が必要です。

- 区切り文字に基づいてイベント・ファイルを分割する  
com.ibm.j2ca.extension.utils.filesplit.SplitByDelimiter クラス。
- イベント・ファイル・サイズに基づいてイベント・ファイルを分割する  
com.ibm.j2ca.extension.utils.filesplit.SplitBySize クラス。

区切り文字またはファイル・サイズは、SplitCriteria プロパティで指定します。

RetrieveContentType プロパティが null に設定されている場合、ファイル・サイズに基づいて分割を実行するクラス名に自動的に設定されます。

表 71. 「分割関数クラス名」プロパティ特性

必須	いいえ
デフォルト	com.ibm.j2ca.extension.utils.filesplit.SplitBySize
プロパティ・タイプ	String
国際化されるかどうか	いいえ

## 「ステージング・ディレクトリー」プロパティ (StagingDirectory)

Outbound の create 操作時に、ファイルは最初にこのディレクトリー内に作成されます。ファイルの作成が完了したら、ファイルは DirectoryPath プロパティで指定されたディレクトリーにコピーされます。このステージング・ディレクトリーは、Append 操作および Overwrite 操作でも使用されます。ここでは、指定されたファイルは StagingDirectory (存在する場合) にコピーされます。付加された内容または上書きされた内容は、この後、元の指定されたディレクトリーに戻されます。StagingDirectory が指定されていない場合、操作は実際の必要なディレクトリー内で実行されます。

注: アダプターでは、StagingDirectory と GenerateUniqueFile の両方のオプションを同時にはサポートしていません。

表 72. 「ステージング・ディレクトリー」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

---

## Inbound 構成プロパティ

WebSphere Adapter for FTP には、複数のカテゴリーに分けられた Inbound 接続構成プロパティがあります。これらは、オブジェクトおよびサービスの生成時または作成時に、外部サービス・ウィザード を使って設定します。WebSphere Integration Developer または WebSphere Process Server の管理コンソールを使ってモジュールをデプロイした後に、リソース・アダプター・プロパティおよびアクティベーション・スペック・プロパティを変更することができます。ただし、外部サービス・ウィザード の接続プロパティは、デプロイメント後に変更することはできません。

### プロパティ関連の情報ガイド

WebSphere Adapter for FTPの構成に使用されるプロパティ (リソース・アダプター・プロパティ、管理接続ファクトリー・プロパティ) の詳細が、構成プロパティの各トピックごとに、表形式で記載されています。これらの表を使いやすくするために、参照する機会が多い各行の情報は、以下のような形式で表されています。

構成プロパティの表で示される各行の意味について、以下の表で説明します。

行	説明
必須	<p>必須フィールド (プロパティ) には、アダプターが機能するための値を設定する必要があります。場合によっては、外部サービス・ウィザードが必須プロパティについてデフォルト値を提供します。</p> <p>外部サービス・ウィザード上の必須フィールドからデフォルト値を除去しても、そのデフォルト値は変更されません。必須フィールドに値がまったく含まれていない場合、外部サービス・ウィザードは割り当てられたデフォルト値を使用して、フィールドを処理します。このデフォルト値は、管理コンソールにも表示されます。</p> <p>使用可能な値は「はい」および「いいえ」です。</p> <p>他のプロパティが特定の値を保有する場合のみ、必須になるプロパティもあります。この場合、表でこの依存関係を示します。例えば以下のとおりです。</p> <ul style="list-style-type: none"> <li>• EventQueryType プロパティが Dynamic に設定されている場合、「はい」</li> <li>• Oracle データベースの場合、「はい」</li> </ul>
使用可能な値	プロパティで選択できる値をリストおよび説明します。
デフォルト	<p>外部サービス・ウィザードによって設定された事前定義値。プロパティが必須の場合、デフォルト値を受け入れるか、または固有の値を指定します。プロパティにデフォルト値がない場合、表にはデフォルト値なしと記載されます。</p> <p>「なし」という単語は、受け入れ可能なデフォルト値で、デフォルト値がないことを意味するわけではありません。</p>
計測単位	プロパティを計測する単位を指定します (キロバイト、秒など)。
プロパティ・タイプ	<p>プロパティ・タイプを記述します。有効なプロパティ・タイプには以下が含まれます。</p> <ul style="list-style-type: none"> <li>• Boolean</li> <li>• String</li> <li>• Integer</li> </ul>
使用法	<p>プロパティに適用される可能性がある使用条件または使用制限について記述します。例として、制約の記載方法を示します。</p> <p>WebSphere Application Server バージョン 6.40、およびそれより前のバージョンでは、パスワードについて以下の制限があります。</p> <ul style="list-style-type: none"> <li>• 大文字である必要があります</li> <li>• 長さが 8 文字である必要があります</li> </ul> <p>WebSphere Application Server の 6.40 より後のバージョンでは、パスワードの制限が以下のように変更されました。</p> <ul style="list-style-type: none"> <li>• 大文字小文字を区別しません</li> <li>• 長さが 40 文字まで可能です。</li> </ul> <p>このセクションでは、このプロパティに影響がある、またはこのプロパティに影響を受ける他のプロパティをリストし、条件関係の性質を記述します。</p>
例	<p>例として、サンプル・プロパティ値を提供します。</p> <p>"言語を JA (日本語) に設定した場合は、コード・ページ番号は 8000 に設定されます"</p>

行	説明
グローバル化	プロパティーが国際化されている場合、各国語をサポートしているため、自国語で値を設定できます。  有効な値は、「はい」 および 「いいえ」です。
BIDI 対応	双方向 (BIDI) 処理でプロパティーをサポートするかどうかを示します。双方向処理は、同じファイル内で左から右のセマンティック・コンテンツ (ヘブライ語やアラビア語など) と右から左のセマンティック・コンテンツ (URL またはファイル・パスなど) の両方を含むデータを処理するタスクのことです。  有効な値は、「はい」 および 「いいえ」です。

## アダプター・タイプ・プロパティー

アダプター・タイプ・プロパティーは、外部サービス・ウィザードにアダプターの詳細情報を提供します。これらのプロパティーは、デプロイメント前に 外部サービス・ウィザードを使用して構成するか、デプロイメント後に WebSphere Application Server 管理コンソールを使用して構成します。

**注:** 双方向スクリプトを使用してこれらのアダプター・タイプ・プロパティーのいずれかを設定する場合、そのプロパティーについて入力された双方向スクリプトの形式を識別できる値を設定する必要があります。

アダプター・タイプ・プロパティーおよびその目的を以下の表に示します。各プロパティーの完全な説明は、表に続くセクションで説明します。続くセクションのプロパティーの詳細表を読む方法については、『プロパティー詳細についてのガイド』を参照してください。

表 73. アダプター・タイプ・プロパティー

プロパティー名		説明
ウィザード内	管理コンソール内	
167 ページの『「説明」プロパティー (Description)』	Description	アダプターの説明。
167 ページの『「表示名」プロパティー (DisplayName)』	DisplayName	アダプター表示名。
167 ページの『「ID」プロパティー (ID)』	ID	アダプター・タイプの ID。
167 ページの『「ベンダー」プロパティー (Vendor)』	Vendor	アダプターを提供するベンダーの名前。
167 ページの『「バージョン」プロパティー (Version)』	Version	アダプターのバージョン。

## 「説明」プロパティ (Description)

アダプターの説明。

表 74. 「説明」プロパティ特性

必須	はい
デフォルト	IBM WebSphere Adapter for FTP
プロパティ・タイプ	String

## 「表示名」プロパティ (DisplayName)

アダプター表示名。

表 75. 「表示名」プロパティ特性

必須	はい
デフォルト	IBM WebSphere Adapter for FTP
プロパティ・タイプ	String

## 「ID」プロパティ (ID)

アダプター・タイプの ID。

表 76. 「ID」プロパティ特性

必須	はい
デフォルト	FTP
プロパティ・タイプ	String

## 「ベンダー」プロパティ (Vendor)

アダプターを提供するベンダーの名前。

表 77. 「ベンダー」プロパティ特性

必須	はい
デフォルト	IBM
プロパティ・タイプ	String

## 「バージョン」プロパティ (Version)

アダプターのバージョン。

表 78. 「バージョン」プロパティ特性

必須	はい
デフォルト	6.1.0
プロパティ・タイプ	String

## リソース・アダプター・プロパティー

リソース・アダプター・プロパティーは、ビジネス・オブジェクトのネーム・スペースの指定など、アダプターの一般的な操作を制御します。リソース・アダプター・プロパティーは、アダプターの構成時に、外部サービス・ウィザードを使用して設定します。アダプターのデプロイ後、管理コンソールを使用して、プロパティーを変更できます。

以下に示すロギングおよびトレースのプロパティーは、バージョン 6.1.0 には必要なくなりましたが、旧バージョンとの互換性を維持するためにサポートされています。

- LogFileMaxSize
- LogFileName
- LogNumberOfFiles
- TraceFileMaxSize
- TraceFileName
- TraceNumberOfFiles

以下の表は、リソース・アダプター・プロパティーおよびその目的をリストしています。各プロパティーの完全な説明は、表に続くセクションで説明します。続くセクションのプロパティーの詳細表を読む方法については、『プロパティー詳細についてのガイド』を参照してください。

表 79. Adapter for FTP 用のリソース・アダプター・プロパティー

プロパティー名		説明
ウィザード内	管理コンソール内	
アダプター ID	AdapterID	CEI イベントおよび PMI イベントのアダプター・インスタンスをロギングおよびトレースを基準にして識別します。
169 ページの『EISEncoding (EISEncoding)』	EISEncoding	FTP サーバーのエンコード。
(なし)	enableHASupport	このプロパティーは変更しないでください。
(なし)	LogFileMaxSize	旧バージョンとの互換性を維持するためにサポートされています。
(なし)	ログ・ファイル名	旧バージョンとの互換性を維持するためにサポートされています。
(なし)	ログ・ファイル数	旧バージョンとの互換性を維持するためにサポートされています。
(なし)	TraceFileMaxSize	旧バージョンとの互換性を維持するためにサポートされています。
(なし)	トレース・ファイル名	旧バージョンとの互換性を維持するためにサポートされています。
(なし)	トレース・ファイル数	旧バージョンとの互換性を維持するためにサポートされています。

## ロギングおよびトレースで使用するアダプター ID (AdapterID)

このプロパティーは、アダプターの特定のデプロイメント (インスタンス) を識別する場合に使用します。

表 80. 「ロギングおよびトレースで使用するアダプター ID」の詳細

必須	はい
デフォルト	CWYFT_FTPFile
プロパティー・タイプ	String
使用法	このプロパティーを使用して、PMI イベントのアダプター・インスタンスを識別します。アダプターのインスタンスを複数デプロイする場合は、このプロパティーをアダプターのインスタンスごとに固有な値に設定します。  Inbound 処理の場合、このプロパティーはリソース・アダプター・プロパティーから取り出します。Outbound 処理の場合、管理接続ファクトリー・プロパティーから取り出します。
グローバル化	はい
BIDI 対応	いいえ

## EISEncoding (EISEncoding)

このプロパティーは、FTP サーバーのエンコードを指定します。FTP サーバーとの通信中の制御接続のエンコードを設定します。FTP サーバーのディレクトリーまたはファイル名に国際化された文字が含まれている場合、このプロパティーを設定します。

表 81. EISEncoding 特性

必須	いいえ
デフォルト	なし
プロパティー・タイプ	String
例	UTF-8、ISO-8859-1

## 高可用性サポートを使用可能にする (enableHASupport)

このプロパティーは変更しないでください。必ず true に設定しておく必要があります。

## ログ・ファイルの最大サイズ (LogFileMaxSize)

このプロパティーでは、ログ・ファイルのサイズをキロバイト単位で指定します。

表 82. 「ログ・ファイルの最大サイズ」の詳細

必須	いいえ
デフォルト	0
プロパティー・タイプ	Integer
使用法	ログ・ファイルがその最大サイズに到達すると、アダプターは新規のログ・ファイルの使用を開始します。ファイル・サイズを 0 と指定するか、最大サイズを指定しなかった場合、そのファイルに最大サイズはありません。

表 82. 「ログ・ファイルの最大サイズ」の詳細 (続き)

グローバル化	はい
BIDI 対応	いいえ

### ログ・ファイル名 (LogFilename)

このプロパティーでは、ログ・ファイルの絶対パス名を指定します。

表 83. 「ログ・ファイル名」の詳細

必須	いいえ
デフォルト	デフォルト値なし
プロパティー・タイプ	String
使用法	このプロパティーは非推奨です。
グローバル化	はい
BIDI 対応	はい

### ログ・ファイルの数 (LogNumberOfFiles)

このプロパティーでは、ログ・ファイルの数を指定します。

表 84. 「ログ・ファイルの数」の詳細

必須	いいえ
デフォルト	1
プロパティー・タイプ	Integer
使用法	ログ・ファイルが最大サイズに到達すると、アダプターは別のログ・ファイルを使い始めます。値を指定しなかった場合、アダプターはログ・ファイルを 1 つ作成します。
グローバル化	はい
BIDI 対応	いいえ

### トレース・ファイルの最大サイズ (TraceFileMaxSize)

このプロパティーでは、トレース・ファイルのサイズをキロバイト単位で指定します。

表 85. 「トレース・ファイルの最大サイズ」の詳細

必須	いいえ
デフォルト	0
プロパティー・タイプ	Integer
使用法	値を指定しなかった場合は、トレース・ファイルに最大サイズが設定されません。
グローバル化	はい
BIDI 対応	いいえ



## トレース・ファイル名 (TraceFilename)

このプロパティーでは、トレース・ファイルの絶対パスを指定します。

表 86. 「トレース・ファイル名」の詳細

必須	いいえ
デフォルト	デフォルト値なし
計測単位	キロバイト
プロパティー・タイプ	String
使用法	このプロパティーは非推奨です。
グローバル化	はい
BIDI 対応	はい

## トレース・ファイルの数 (TraceNumberOfFiles)

このプロパティーでは、使用するトレース・ファイルの数を指定します。トレース・ファイルが最大サイズに到達すると、アダプターは別のトレース・ファイルを使い始めます。

表 87. 「トレース・ファイルの数」の詳細

必須	いいえ
デフォルト	1
プロパティー・タイプ	Integer
使用法	値を指定しなかった場合、アダプターはトレース・ファイルを 1 つ使用します。
グローバル化	はい
BIDI 対応	いいえ

## アクティベーション・スペック・プロパティー

アクティベーション・スペック・プロパティーは、メッセージ・エンドポイント用の Inbound イベント処理の構成情報を保持するプロパティーです。

アクティベーション・スペック・プロパティーは、エンドポイント・アクティベーション時に使用され、アダプターに適確なイベント・リスナーを通知します。Inbound 処理中に、アダプターはこれらのイベント・リスナーを使用して、エンドポイントに転送する前にイベントを受信します (メッセージ・ドリブン Bean)。

アクティベーション・スペック・プロパティーを設定するには、外部サービス・ウィザードを使用します。変更する場合には、WebSphere Integration Developer のアセンブリー・エディターを使用することもできますし、デプロイメント後に WebSphere Process Server 管理コンソールを通じて行うこともできます。

以下の表は、アクティベーション・スペック・プロパティーをリストしています。各プロパティーの完全な説明は、表に続くセクションで説明します。続くセクションのプロパティーの詳細表を読む方法については、『プロパティー詳細についてのガイド』を参照してください。

表 88. アクティベーション・スペック・プロパティ

プロパティ名		説明
ウィザード内	管理コンソール内	
176 ページの『イベントを一度のみ送達する (AssuredOnceDelivery)』	AssuredOnceDelivery	アダプターにより、1 回のイベント送達を確保する機能が提供されるかどうかを指定します。
176 ページの『「自動作成イベント・テーブル」プロパティ (CreateTable)』	CreateTable	イベント・パーシスタンス・テーブルを作成するかどうかアダプターに通知する。
176 ページの『「テーブル作成」プロパティ (CreateTable)』	CreateTable	true に設定すると、イベント・テーブルおよび関連した索引が作成される。
177 ページの『「カスタム・パーサー・クラス名」プロパティ (CustomParserClassName)』	CustomParserClassName	ls -l の出力を解析するために使用されるカスタム・パーサーの完全修飾クラス名。
177 ページの『「データベース・パスワード」プロパティ (DatabasePassword)』	DatabasePassword	データ・ソースから JDBC データベース接続を取得するためにイベント・パーシスタンスが使用するパスワード。
177 ページの『「データベース・スキーマ名」プロパティ (SchemaName)』	SchemaName	イベント・パーシスタンスによって使用されるデータベースのスキーマ名。
177 ページの『「データベース・ユーザー名 (Database Username)」プロパティ (DatabaseUsername)』	DatabaseUsername	データ・ソースから JDBC データベース接続を取得するためにイベント・パーシスタンスが使用するユーザー名。
178 ページの『「FTP サーバー接続モード」プロパティ (DataConnectionMode)』	DataConnectionMode	FTP サーバーがファイル転送時に使用するデータ接続モード。
(なし)	DefaultObjectName	旧バージョンとの互換性を維持するためにサポートされています。
178 ページの『送達タイプ (DeliveryType)』	DeliveryType	イベントがアダプターによってエクスポートに配信される順序を指定します。
178 ページの『「FTP サーバーで使用されるエンコード」プロパティ (EISEncoding)』	EISEncoding	FTP サーバーのエンコード。
(なし)	EventContentType	旧バージョンとの互換性を維持するためにサポートされています。
179 ページの『「イベント・リカバリー DataSource (JNDI) 名」プロパティ (DataSourceJNDIName)』	DataSourceJNDIName	JDBC データベース接続を取得するためにイベント・パーシスタンスで使用されるデータ・ソースの JNDI 名。
179 ページの『「イベント・リカバリー・テーブル名」プロパティ (EventTableName)』	TableName	アダプターがイベント・パーシスタンスのために使用するテーブルの名前。

表 88. アクティベーション・スペック・プロパティ (続き)

180 ページの『処理するイベント・タイプ (EventTypeFilter)』	EventTypeFilter	どのイベントをアダプターが配信するかをアダプターに示す、区切り文字で区切られているイベント・タイプのリスト。
180 ページの『「ローカル・アーカイブ用の障害ファイル拡張子」プロパティ (FailedArchiveExt)』	FailedArchiveExt	正常に処理されなかったイベント・ファイル内のビジネス・オブジェクトをアーカイブするために使用されるファイル拡張子。
180 ページの『「ファイル内容のエンコード」プロパティ (FileContentEncoding)』	FileContentEncoding	イベント・ファイルを読み取るために使用されるエンコード。
181 ページの『「リモート・アーカイブ用のファイル拡張子」プロパティ (FTPRenameExtension)』	FTPRenameExt	アダプターがリモート FTP ファイルの名前変更使用するファイル拡張子または接尾部
181 ページの『「内容ではなくファイル名およびディレクトリのみを受け渡し」プロパティ (FilePassByReference)』	FilePassByReference	イベント・ファイルのファイル内容をエクスポートに送信しないことを指定する。
181 ページの『「ファイル転送タイプ」プロパティ (FileTransferType)』	FileTransferType	Inbound 処理時に使用されるファイル転送タイプ。
182 ページの『「将来のタイム・スタンプを持つイベントを処理しない」 (FilterFutureEvents)』	FilterFutureEvents	アダプターが各イベントのタイム・スタンプをシステム時刻と比較することによって、将来のイベントをフィルターで除去するかどうかを指定します。
182 ページの『「一度に取得するファイル数」プロパティ (FTPGetQuantity)』	FTPGetQuantity	リモート FTP URL から取得するファイルの数を決定する。
182 ページの『「各ダウンロード間のポーリング期間の数」プロパティ (FTPPollFrequency)』	FTPPollFrequency	アダプターが FTP サーバーをポーリングする頻度を決定する。
183 ページの『「ファイルのダウンロード後に実行する FTP スクリプト・ファイル」プロパティ』	FTPScriptFileExecuted AfterInbound	各 Inbound ポーリング周期の後に実行されるスクリプト・ファイルのパスを指定します。
183 ページの『「ファイルのダウンロード前に実行する FTP スクリプト・ファイル」プロパティ』	FTPScriptFileExecuted BeforeInbound	各 Inbound ポーリング周期の前に実行されるスクリプト・ファイルのパスを指定します。
183 ページの『「ホスト名」プロパティ (HostName)』	HostName	接続の確立先となる FTP サーバーのホスト名。

表 88. アクティベーション・スペック・プロパティー (続き)

183 ページの『「ファイル内容にビジネス・オブジェクト区切り文字を組み込む」プロパティー (IncludeEndBODelimiter)』	IncludeEndBODelimiter	true に設定する場合、さらに処理を行うため、ビジネス・オブジェクトの内容と一緒に区切り文字が送信されます。
184 ページの『「ローカル・アーカイブ・ディレクトリー」プロパティー (LocalArchiveDirectory)』	LocalArchiveDirectory	ローカル・アーカイブ・ディレクトリーの絶対パス。
184 ページの『「ローカル・ディレクトリー」プロパティー (LocalEventDirectory)』	LocalEventDirectory	アダプターが FTP サイトからダウンロードしたイベント・ファイルを格納するローカル・システム・ディレクトリー。
184 ページの『最大接続数 (MaximumConnections)』	MaximumConnections	アダプターが Inbound イベント送達に使用できる接続の最大数
185 ページの『最小接続数 (MinimumConnections)』	MinimumConnections	アダプターが Inbound イベント送達に使用できる接続の最小数
185 ページの『「ローカル・アーカイブ用のファイル拡張子」プロパティー (OriginalArchiveExt)』	OriginalArchiveExt	オリジナル・イベント・ファイルをアーカイブするために使用されるファイル拡張子。
185 ページの『「パスワード」プロパティー (Password)』	Password	FTP サーバーへの接続特権、および FTP 操作の実行特権を持つユーザーのパスワード。
185 ページの『「イベント DataSource に接続するのに使用されるパスワード」プロパティー (Password)』	Password	イベント・パーシスタンス時に使用されるパスワード。
186 ページの『ポーリング期間の間隔 (PollPeriod)』	PollPeriod	ポーリング期間中にアダプターが待機する時間の長さ
186 ページの『ポーリング期間内の最大イベント数 (PollQuantity)』	PollQuantity	各ポーリング期間中にアダプターがエクスポートに配信するイベント数
186 ページの『「ポート番号」プロパティー (PortNumber)』	PortNumber	FTP サーバーのポート番号。
187 ページの『「プロトコル」プロパティー (Protocol)』	Protocol	接続を通常の FTP、セキュア FTP のいずれにするのかを決める。
188 ページの『「このパターンを持つファイルの取得」プロパティー (EventFileMask)』	EventFileMask	イベント・ファイル用のフィルター。
188 ページの『接続が失敗した場合の再試行間隔 (RetryInterval)』	RetryInterval	Inbound 操作時のエラー後、新規接続を確立しようとする試行間にアダプターが待機する時間の長さ

表 88. アクティベーション・スペック・プロパティ (続き)

189 ページの『システム接続を再試行する回数 (RetryLimit)』	RetryLimit	エラーが発生したあと、アダプターが Inbound 接続の再確立を試行する回数。
187 ページの『「リモート・アーカイブ・ディレクトリー」プロパティ (FTPArchiveDirectory)』	FTPArchiveDirectory	FTP サーバー上のアーカイブ・ディレクトリーの相対パス。
188 ページの『「リモート・ディレクトリー」プロパティ (EventDirectory)』	EventDirectory	Inbound 処理のためにイベント・ファイルが取り出される FTP サーバーのリモート・ディレクトリー。
189 ページの『「ホスト名」プロパティ (SocksProxyHost)』	SocksProxyHost	プロキシ・サーバーとして使用されるマシンのホスト名。
189 ページの『「パスワード」プロパティ (SocksProxyPassword)』	SocksProxyPassword	プロキシ・サーバーを認証するためのパスワード。
189 ページの『「ポート番号」プロパティ (SocksProxyPort)』	SocksProxyPort	プロキシ・サーバーのポート番号。
190 ページの『「ユーザー名」プロパティ (SocksProxyUserName)』	SocksProxyUserName	プロキシ・サーバーを認証するためのユーザー名。
190 ページの『「ソート・イベント・ファイル」プロパティ (SortEventFiles)』	SortEventFiles	ポーリングされるイベント・ファイルのソート順を決定する。
190 ページの『「ファイル内容を分割するための基準の指定」プロパティ (SplitCriteria)』	SplitCriteria	SplittingFunctionClassName プロパティの値に基づいた各種の値を指定する。
191 ページの『「分割関数クラス名」プロパティ』	SplittingFunctionClassName	ファイル分割を有効にするために使用するクラス・ファイルの完全修飾クラス名を指定する。
191 ページの『「ポーリング時にエラーが検出された場合はアダプターを停止する (StopPollingOnError)』	StopPollingOnError	ポーリング時にアダプターがエラーを検出した場合、アダプターがイベントのポーリングを停止するかどうかを指定します。
192 ページの『「ローカル・アーカイブ用の成功ファイル拡張子」プロパティ (SuccessArchiveExt)』	SuccessArchiveExt	正常に処理されたビジネス・オブジェクトすべてをアーカイブするために使用されるファイル拡張子。
192 ページの『「ユーザー名」プロパティ (UserName)』	UserName	FTP サーバーへの接続特権、および FTP 操作の実行特権を持つユーザーの名前。
192 ページの『「イベント DataSource に接続するのに使用されるユーザー名」プロパティ (UserName)』	UserName	データベース接続を取得するためにイベント・パーシスタンスが使用するユーザー名。

## イベントを一度のみ送達する (AssuredOnceDelivery)

このプロパティーでは、Inbound イベントに対して、「イベントを一度のみ送達する」の機能を提供するかどうかを指定します。

表 89. 「イベントを一度のみ送達する」の詳細

必須	はい
使用可能な値	True False
デフォルト	True
プロパティー・タイプ	Boolean
使用法	<p>このプロパティーを True に設定すると、アダプターにより、1 回のイベント送達を確保する機能が提供されます。つまり、各イベントは 1 回のみ配信されます。値を False にすると、1 回のイベント送達を確保する機能は提供されませんが、パフォーマンスは向上します。</p> <p>このプロパティーを True に設定すると、アダプターにより、トランザクション (XID) 情報のイベント・ストアへの保管が試行されます。このプロパティーを False に設定した場合は、アダプターではこの情報の保管は行われません。</p> <p>このプロパティーは、エクスポート・コンポーネントがトランザクションの対象である場合のみ使用されます。そうでない場合は、このプロパティーの値に関係なく、トランザクションを使用することはできません。</p>
グローバル化	いいえ
BIDI 対応	いいえ

## 「自動作成イベント・テーブル」プロパティー (CreateTable)

イベント・パーシスタンス・テーブルを作成するかどうかアダプターに通知する。値が true でテーブルが存在しない場合、アダプターはテーブルを作成します。値が false の場合、アダプターはテーブルを作成しません。

表 90. 「自動作成イベント・テーブル」プロパティー特性

必須	いいえ
デフォルト	true
プロパティー・タイプ	Boolean
国際化されるかどうか	いいえ

## 「テーブル作成」プロパティー (CreateTable)

true に設定すると、イベント・テーブルおよび関連した索引が作成される。テーブル作成エラーをトラブルシューティングするには、このプロパティーを false に設定します。その後、テーブルおよび索引を手動で作成できます。

表 91. 「テーブル作成」プロパティー特性

必須	いいえ
デフォルト	true
プロパティー・タイプ	Boolean

表 91. 「テーブル作成」プロパティ特性 (続き)

国際化されるかどうか	いいえ
------------	-----

### 「カスタム・パーサー・クラス名」プロパティ (CustomParserClassName)

ls -l の出力を解析するために使用されるカスタム・パーサーの完全修飾クラス名。ls -l の出力が標準出力から外れる場合にのみ使用されます。

表 92. 「カスタム・パーサー・クラス名」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	いいえ

### 「データベース・パスワード」プロパティ (DatabasePassword)

データ・ソースから JDBC データベース接続を取得するためにイベント・パーシスタンスが使用するパスワード。

表 93. 「データベース・パスワード」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

### 「データベース・スキーマ名」プロパティ (SchemaName)

イベント・パーシスタンスによって使用されるデータベースのスキーマ名。

表 94. 「データベース・スキーマ名」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

### 「データベース・ユーザー名 (Database Username)」プロパティ (DatabaseUsername)

データ・ソースから JDBC データベース接続を取得するためにイベント・パーシスタンスが使用するユーザー名。

表 95. 「データベース・ユーザー名 (Database Username)」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String

表 95. 「データベース・ユーザー名 (Database Username)」プロパティ特性 (続き)

国際化されるかどうか	はい
------------	----

### 「FTP サーバー接続モード」プロパティ (DataConnectionMode)

FTP サーバーがファイル転送時に使用するデータ接続モード。active または passive のいずれかの設定値を受け入れます。

表 96. 「FTP サーバー接続モード」プロパティ特性

必須	いいえ
デフォルト	active
プロパティ・タイプ	String
国際化されるかどうか	いいえ

### 送達タイプ (DeliveryType)

このプロパティでは、イベントがアダプターによってエクスポートに配信される順序を指定します。

表 97. 「送達タイプ」の詳細

必須	いいえ
使用可能な値	ORDERED UNORDERED
デフォルト	ORDERED
プロパティ・タイプ	String
使用法	以下の値がサポートされています。 <ul style="list-style-type: none"> <li>• ORDERED: アダプターは、一度に 1 つのイベントをエクスポートに配信します。</li> <li>• UNORDERED: アダプターは、一度にすべてのイベントをエクスポートに配信します。</li> </ul>
グローバル化	いいえ
BIDI 対応	いいえ

### 「FTP サーバーで使用されるエンコード」プロパティ (EISEncoding)

FTP サーバーのエンコード。この値は、FTP サーバーとの制御接続のエンコードを設定する場合に使用します。

- アダプター・レベルの EISEncoding とアクティベーション・スペック・レベルの EISEncoding の両方を設定しない場合 (両方とも null)、FTP サーバーとの通信中は制御接続で何も設定されません。
- アダプター・レベルの EISEncoding を設定し、アクティベーション・スペック・レベルの EISEncoding を設定しない場合、アダプター・レベルの値は、FTP サーバーとの通信中に制御接続で設定されます。これは、複数のアクティベーション・スペックを使用し、同じエンコードが設定されている場合に便利です。この場合は、すべての接続で制御接続のエンコードが同じになるよう、値をアダプター・レベルで設定します。



- アダプター・レベルの EISEncoding を設定しないで、アクティベーション・スペック・レベルの EISEncoding を設定する場合、アクティベーション・スペック・レベルの値は、FTP サーバーとの通信中に制御接続で設定されます。値はアクティベーション・スペック・レベルであるため、これが適用されるのは、そのアクティベーション・スペックのみです。
- アダプター・レベルの EISEncoding とアクティベーション・スペック・レベルの EISEncoding の両方を設定する場合、アクティベーション・スペック・レベルの値が優先されます。

この属性には、Java がサポートする任意のエンコードを指定できます。

表 98. 「FTP サーバーで使用されるエンコード」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティタイプ	String
国際化されるかどうか	はい

### 「イベント・リカバリー DataSource (JNDI) 名」プロパティ (DataSourceJNDIName)

JDBC データベース接続を取得するためにイベント・パーシスタンスで使用されるデータ・ソースの JNDI 名。データ・ソースは、WebSphere Process Server 内に作成する必要があります。データ・ソースの作成中に指定されるデータベース名は、既に存在しているデータベースでなければなりません。

表 99. 「イベント・リカバリー DataSource (JNDI) 名」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティタイプ	String
国際化されるかどうか	はい

### 「イベント・リカバリー・テーブル名」プロパティ (EventTableName)

アダプターがイベント・パーシスタンスのために使用するテーブルの名前。複数のアクティベーション・スペックを使用する場合、この値はアクティベーション・スペックごとに固有でなければなりません。同じアダプター、または異なるアダプターのほかのインスタンスが同じテーブル名を使用することはできません。データベースにこのテーブルが存在しない場合は、アダプターがこのテーブルを作成します。

表 100. 「イベント・リカバリー・テーブル名」プロパティ特性

必須	いいえ
デフォルト	FTPTABLE
プロパティタイプ	String
国際化されるかどうか	はい

## 処理するイベント・タイプ (EventTypeFilter)

このプロパティには、どのイベントをアダプターが配信するかをアダプターに示す、区切り文字で区切られているイベント・タイプのリストが入っています。

表 101. 「処理するイベント・タイプ」の詳細

必須	いいえ
使用可能な値	ビジネス・オブジェクト・タイプのコンマ (,) 区切りのリスト
デフォルト	null
プロパティ・タイプ	String
使用法	イベントは、ビジネス・オブジェクト・タイプ別にフィルタリングされます。このプロパティを設定すると、アダプターは、リスト内に存在するイベントのみを配信するようになります。値が null の場合は、フィルターが適用されず、すべてのイベントはエクスポートに配信されることを示しています。
例	Customer ビジネス・オブジェクトおよび Order ビジネス・オブジェクトに関連するイベントのみを受信するには、次の値を指定します。  Customer,Order
グローバル化	いいえ
BIDI 対応	いいえ

## 「ローカル・アーカイブ用の障害ファイル拡張子」プロパティ (FailedArchiveExt)

正常に処理されなかったイベント・ファイル内のビジネス・オブジェクトをアーカイブするために使用されるファイル拡張子。このプロパティは、LocalArchiveDirectory が有効で存在する場合にのみ使用されます。

表 102. 「ローカル・アーカイブ用の障害ファイル拡張子」プロパティ特性

必須	いいえ
デフォルト	fail
プロパティ・タイプ	String
国際化されるかどうか	はい

## 「ファイル内容のエンコード」プロパティ (FileContentEncoding)

EndBODelimiter プロパティに基づいたイベント・ファイルの読み取りで使用するエンコード、およびストリングから byte[] への変換時に使用されるエンコード。指定しない場合、アダプターは特定のエンコードを使用せずに読み取りを行おうとします。Java がサポートする任意のエンコード・セットを指定できます。

表 103. 「ファイル内容のエンコード」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	いいえ

## 「リモート・アーカイブ用のファイル拡張子」プロパティ (FTPRenameExtension)

コネクタがリモート FTP ファイルをポーリングした後、アダプターがリモート FTP ファイルの名前変更使用するファイル拡張子または接尾部。ファイルの名前変更を行うと、コネクタが次のポーリング周期で同じファイルをポーリングしないように設定できます。アダプターを、処理済みのイベント・ファイルの名前を変更して、そのファイルをアーカイブ・ディレクトリーに移動するように構成することができます。

表 104. 「リモート・アーカイブ用のファイル拡張子」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

## 「内容ではなくファイル名およびディレクトリーのための受け渡し」プロパティ (FilePassByReference)

イベント・ファイルのファイル内容をエクスポートに送信しないことを指定する。

true に設定する場合、ファイルにタイム・スタンプが付加され、LocalArchiveDirectory に送信されます。タイム・スタンプにより、同じ名前での別のファイルが受信される場合に起きる、エラーやファイルの上書きを回避できます。このプロパティを true に設定できるのは、LocalArchiveDirectory プロパティを設定する場合で、指定したディレクトリーが存在する場合のみです。このプロパティは、PassThrough Inbound 処理でのみ使用されます。有効にすると、ファイルはチャンクに分割されません。

表 105. 「内容ではなくファイル名およびディレクトリーのための受け渡し」プロパティ特性

必須	いいえ
デフォルト	false
プロパティ・タイプ	Boolean
国際化されるかどうか	いいえ

## 「ファイル転送タイプ」プロパティ (FileTransferType)

Inbound 処理時に使用されるファイル転送タイプ。ASCII または binary のいずれかの設定値を受け入れます。

表 106. 「ファイル転送タイプ」プロパティ特性

必須	いいえ
デフォルト	binary
プロパティ・タイプ	String
国際化されるかどうか	no

## 将来のタイム・スタンプを持つイベントを処理しない (FilterFutureEvents)

このプロパティーでは、アダプターが各イベントのタイム・スタンプをシステム時刻と比較することによって、将来のイベントをフィルターで除去するかどうかを指定します。

表 107. 「将来のタイム・スタンプを持つイベントを処理しない」の詳細

必須	はい
使用可能な値	True False
デフォルト	False
プロパティー・タイプ	Boolean
使用法	True に設定すると、アダプターは各イベントの時刻をシステム時刻と比較します。イベント時刻がシステム時刻より後の時刻である場合、そのイベントは配信されません。  False に設定すると、アダプターはすべてのイベントを配信します。
グローバル化	いいえ
BIDI 対応	いいえ

## 「一度に取得するファイル数」プロパティー (FTPGetQuantity)

リモートでのポーリングごとに、リモート FTP URL から取得するファイルの数を決定する。

表 108. 「一度に取得するファイル数」プロパティー特性

必須	はい
デフォルト	10
プロパティー・タイプ	Integer
国際化されるかどうか	いいえ

## 「各ダウンロード間のポーリング期間の数」プロパティー (FTPPollFrequency)

アダプターが FTP サーバーをポーリングする頻度 (標準のポーリング周期の数で測定される) を決定する。例えば、PollPeriod を 10000 に設定し、FTPPollFrequency を 6 に設定した場合、アダプターは LocalEventDirectory を 10 秒ごとにポーリングし、リモートの EventDirectory を 60 秒ごとにポーリングします。アダプターが FTP のポーリングを行うのは、このプロパティーに値を指定した場合のみです。PollPeriod が 0 の場合は、1 として計算します。計算により 0 と評価された場合、アダプターは FTP のポーリングを行いません。

表 109. 「各ダウンロード間のポーリング期間の数」プロパティー特性

必須	はい
デフォルト	5
プロパティー・タイプ	Integer
国際化されるかどうか	いいえ

## 「ファイルのダウンロード後に実行する FTP スクリプト・ファイル」プロパティ

各 Inbound ポーリング周期の後に実行されるスクリプト・ファイルのパスを指定します。この機能を使用して、各ポーリング周期の後に FTP サーバーで追加アクションを実行できます。例えば、ファイル・アクセス権を設定することが可能です。

表 110. 「ファイルのダウンロード後に実行する FTP スクリプト・ファイル」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

## 「ファイルのダウンロード前に実行する FTP スクリプト・ファイル」プロパティ

各 Inbound ポーリング周期の前に実行されるスクリプト・ファイルのパスを指定します。この機能を使用して、各ポーリング周期の前に FTP サーバーで追加アクションを実行できます。例えば、ファイル・アクセス権を設定することが可能です。

表 111. 「ファイルのダウンロード前に実行する FTP スクリプト・ファイル」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

## 「ホスト名」プロパティ (HostName)

Inbound 処理時に接続の確立先となる FTP サーバーのホスト名。

表 112. 「テーブル作成 (Create Table)」プロパティ特性

必須	はい
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

## 「ファイル内容にビジネス・オブジェクト区切り文字を組み込む」プロパティ (IncludeEndBODelimiter)

true に設定する場合、さらに処理を行うため、ビジネス・オブジェクトの内容と一緒に区切り文字が送信されます。このプロパティは、区切り文字に基づいてイベント・ファイルを分割する場合にのみ有効です。

表 113. 「ファイル内容にビジネス・オブジェクト区切り文字を組み込む」プロパティ特性

必須	いいえ
デフォルト	false

表 113. 「ファイル内容にビジネス・オブジェクト区切り文字を組み込む」プロパティ特性 (続き)

プロパティ・タイプ	String
国際化されるかどうか	いいえ

### 「ローカル・アーカイブ・ディレクトリー」プロパティ (LocalArchiveDirectory)

ローカル・アーカイブ・ディレクトリーの絶対パス。ディレクトリーは、有効で既に存在している必要があります。

表 114. 「ローカル・アーカイブ・ディレクトリー」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

### 「ローカル・ディレクトリー」プロパティ (LocalEventDirectory)

アダプターが FTP サイトからダウンロードしたイベント・ファイルを格納するローカル・システム・ディレクトリー。アダプターがイベントを処理できるようにするために、このプロパティに値を指定する必要があります。

表 115. 「ローカル・ディレクトリー」プロパティ特性

必須	はい
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

### 最大接続数 (MaximumConnections)

このプロパティでは、アダプターが Inbound イベント送達に使用できる接続の最大数を指定します。

表 116. 「最大接続数」の詳細

必須	いいえ
デフォルト	1
プロパティ・タイプ	Integer
使用法	正の値のみが有効です。1 より小さい正の入力値は、アダプターによって 1 と解釈されます。このプロパティに負の値または 1 を入力すると、ランタイム・エラーが発生するおそれがあります。
グローバル化	いいえ
BIDI 対応	いいえ

## 最小接続数 (MinimumConnections)

このプロパティーでは、アダプターが Inbound イベント送達に使用できる接続の最小数を指定します。

表 117. 「最小接続数」の詳細

必須	いいえ
デフォルト	1
プロパティー・タイプ	Integer
使用法	正の値のみが有効です。1 より小さい値は、アダプターによって 1 として処理されます。このプロパティーに負の値または 1 を入力すると、ランタイム・エラーが発生するおそれがあります。
グローバル化	いいえ
BIDI 対応	いいえ

## 「ローカル・アーカイブ用のファイル拡張子」プロパティー (OriginalArchiveExt)

オリジナル・イベント・ファイルをアーカイブするために使用されるファイル拡張子。このファイルには、ビジネス・オブジェクトのいずれかが失敗した場合の参照用にイベント・ファイル全体が保存されます。このプロパティーは、LocalArchiveDirectory が有効で存在する場合にのみ使用されます。

表 118. 「ローカル・アーカイブ用のファイル拡張子」プロパティー特性

必須	いいえ
デフォルト	original
プロパティー・タイプ	String
国際化されるかどうか	はい

## 「パスワード」プロパティー (Password)

FTP サーバーへの接続特権、および FTP 操作の実行特権を持つユーザーのパスワード。EventDirectory プロパティーで指定した URL にパスワードが含まれている場合は、このプロパティーに値を指定する必要はありません。

表 119. 「パスワード」プロパティー特性

必須	いいえ
デフォルト	なし
プロパティー・タイプ	String
国際化されるかどうか	はい

## 「イベント DataSource に接続するのに使用されるパスワード」プロパティー (Password)

データ・ソースからデータベース接続を取得するためにイベント・パーシスタンスで使用されるパスワード。

表 120. 「イベント DataSource に接続するのに使用されるパスワード」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

## ポーリング期間の間隔 (PollPeriod)

このプロパティでは、ポーリング期間中にアダプターが待機する時間の長さを指定します。

表 121. 「ポーリング期間の間隔」の詳細

必須	はい
使用可能な値	0 以上の整数。
デフォルト	2000
計測単位	ミリ秒
プロパティ・タイプ	Integer
使用法	ポーリング期間は一定の割合で確立されます。つまり、ポーリング周期の実行が何らかの理由で遅延すると (例えば、前のポーリング周期が完了するまでに予想より時間がかかった場合)、遅延によって失った時間を取り戻すために次のポーリング周期がすぐに開始されます。
グローバル化	いいえ
BIDI 対応	いいえ

## ポーリング期間内の最大イベント数 (PollQuantity)

このプロパティでは、各ポーリング期間中にアダプターがエクスポートに配信するイベント数を指定します。

表 122. 「ポーリング期間内の最大イベント数」の詳細

必須	はい
デフォルト	10
プロパティ・タイプ	Integer
使用法	値は 0 より大きくする必要があります。この値を大きくすると、ポーリング期間ごとに処理されるイベントの数が増加し、アダプターのパフォーマンス効率が低下するおそれがあります。この値を小さくすると、ポーリング期間ごとに処理されるイベントの数が減少し、アダプターのパフォーマンスが若干向上することがあります。
グローバル化	いいえ
BIDI 対応	いいえ

## 「ポート番号」プロパティ (PortNumber)

Inbound 処理時に接続の確立先となる FTP サーバーのポート番号。



表 123. 「ポート番号」プロパティ特性

必須	はい
デフォルト	21
プロパティ・タイプ	Integer
国際化されるかどうか	いいえ

### 「プロトコル」プロパティ (Protocol)

確立する接続が通常の FTP 接続か、またはセキュア FTP 接続かを定めるプロトコル。

例:

通常接続: FTP

セキュア接続: FTPS

表 124. 「プロトコル」プロパティ特性

必須	はい
デフォルト	FTP
プロパティ・タイプ	String
国際化されるかどうか	いいえ

### 「リモート・アーカイブ・ディレクトリー」プロパティ (FTPArchiveDirectory)

FTP サーバー上のアーカイブ・ディレクトリーの相対パス。このディレクトリーが存在していなければなりません。このプロパティを使用してアーカイブを指定する場合には、いくつかのオプションがあります。

- このプロパティに値を指定したが、FTPRenameExt プロパティに値を指定しない場合、アダプターは、イベント・ファイル名にタイム・スタンプを付加し、ファイルをこのプロパティに指定された FTP サーバーのアーカイブ・ディレクトリーに移動します。
- このプロパティと FTPRenameExt プロパティの両方に値を指定した場合、アダプターは、処理済みのイベント・ファイルの名前をタイム・スタンプと FTPRenameExt に指定された値で変更し、ファイルをこのプロパティに指定された FTP サーバーのアーカイブ・ディレクトリーに移動します。
- このプロパティにも FTPRenameExt プロパティにも値を指定しない場合、アダプターは、処理済みのイベント・ファイルをアーカイブせずに削除します。
- このプロパティには値を指定せずに、FTPRenameExt プロパティに値を指定した場合、アダプターは処理済みのイベント・ファイルの名前を変更し、タイム・スタンプと FTPRenameExt に指定された値を追加します。

表 125. 「リモート・アーカイブ・ディレクトリー」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String

表 125. 「リモート・アーカイブ・ディレクトリー」プロパティー特性 (続き)

国際化されるかどうか	はい
------------	----

### 「リモート・ディレクトリー」プロパティー (EventDirectory)

Inbound 処理のためにイベント・ファイルが取り出される FTP サーバーのリモート・ディレクトリー。

表 126. 「リモート・ディレクトリー」プロパティー特性

必須	はい
デフォルト	*.*
プロパティー・タイプ	String
国際化されるかどうか	はい

### 「このパターンを持つファイルの取得」プロパティー (EventFileMask)

イベント・ファイル用のフィルター。ファイルのフィルターは、英数字と \* および ? のワイルドカードから成る適切な修飾の表現です。

表 127. 「このパターンを持つファイルの取得」プロパティー特性

必須	はい
デフォルト	*.*
プロパティー・タイプ	String
国際化されるかどうか	はい

### 接続が失敗した場合の再試行間隔 (RetryInterval)

このプロパティーでは、アダプターが Inbound 接続に関連したエラーを検出した場合に、アダプターが新規接続を確立しようとするまで待機する時間の長さを指定します。

表 128. 再試行間隔の詳細

必須	はい
デフォルト	2000
計測単位	ミリ秒
プロパティー・タイプ	Integer
使用法	正の値のみが有効です。このプロパティーでは、アダプターが Inbound 接続に関連したエラーを検出した場合に、アダプターが新規接続を確立しようとするまで待機する時間の長さを指定します。
グローバル化	はい
BIDI 対応	いいえ

## システム接続を再試行する回数 (RetryLimit)

このプロパティでは、アダプターが Inbound 接続の再確立を試行する回数を指定します。

表 129. 「システム接続を再試行する回数」の詳細

必須	いいえ
使用可能な値	正整数
デフォルト	0
プロパティ・タイプ	Integer
使用法	正の値のみが有効です。 このプロパティでは、アダプターが Inbound 接続に関連したエラーを検出した場合に、アダプターが接続を再開しようとする回数を指定します。値を 0 にすると、再試行回数は無限になります。
グローバル化	はい
BIDI 対応	いいえ

## 「ホスト名」プロパティ (SocksProxyHost)

アダプターの要求を FTP サーバーに送信する際に経由するプロキシ・サーバーとして使用されるマシンのホスト名。

表 130. 「ホスト名」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

## 「パスワード」プロパティ (SocksProxyPassword)

プロキシ・サーバーを認証するためのパスワード。

表 131. 「パスワード」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

## 「ポート番号」プロパティ (SocksProxyPort)

アダプターの要求を FTP サーバーに送信する際に経由するプロキシ・サーバーのポート番号。

表 132. 「ポート番号」プロパティ特性

必須	いいえ
デフォルト	1080

表 132. 「ポート番号」プロパティ特性 (続き)

プロパティ・タイプ	Integer
国際化されるかどうか	いいえ

### 「ユーザー名」プロパティ (SocksProxyUserName)

プロキシ・サーバーを認証するためのユーザー名。

表 133. 「ユーザー名」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

### 「ソート・イベント・ファイル」プロパティ (SortEventFiles)

ポーリングされるイベント・ファイルのソート順を決定する。以下をサポートしています。

- ファイル名順 - ファイル名を基準に昇順にソート
- 日時順 - 最終変更日時のタイム・スタンプを基準に昇順でソート
- ソートなし - ソートされません

イベントを送信するためのイベント・ファイルの順序付けは、アクティベーション・スペックの `DeliveryType` プロパティが `ORDERED` に設定されている場合にのみ有効です。ファイル名のソートは、FTP サーバーのロケールに基づいて行われます。ロケールおよびそれに対応する規則の追跡には、`ICU4J` パッケージが使用されます。

表 134. 「ソート・イベント・ファイル」プロパティ特性

必須	いいえ
デフォルト	<空白> (= ソートなし)
プロパティ・タイプ	String
国際化されるかどうか	いいえ

### 「ファイル内容を分割するための基準の指定」プロパティ (SplitCriteria)

このプロパティには、`SplittingFunctionClassName` プロパティの値に基づいた各種の値を指定する。例: 5 KB ごとにファイルを分割するように指定するには、`SplitCriteria` プロパティを 5000 に設定します。

- `SplittingFunctionClassName` プロパティで、区切り文字に基づいてファイルを分割することを指定する場合、`SplitCriteria` にはイベント・ファイル内のビジネス・オブジェクトを分割する区切り文字を指定します。
- `SplittingFunctionClassName` をサイズに基づいた分割を行う値に設定する場合、`SplitCriteria` プロパティには、バイト単位のサイズを表す有効な数値を指定します。

- イベント・ファイルのサイズがこの値より大きい場合、アダプターはファイルをこのサイズのチャンクに分割し、各チャンクが送られます。
- イベント・ファイルのサイズがこの値より小さい場合、イベント・ファイル全体が送られます。SplitCriteria=0 の場合、チャンクへの分割は無効です。

Inbound PassThrough 時に FilePassByReference を有効にすると、イベント・ファイルは分割されません。

表 135. 「ファイル内容を分割するための基準の指定」プロパティー特性

必須	いいえ
デフォルト	0
プロパティー・タイプ	String
国際化されるかどうか	はい

### 「分割関数クラス名」プロパティー

この値には、ファイル分割を有効にするために使用するクラス・ファイルの完全修飾クラス名を指定する。以下の 2 つの値が必要です。

- 区切り文字に基づいてイベント・ファイルを分割する  
com.ibm.j2ca.extension.utils.filesplit.SplitByDelimiter クラス。
- イベント・ファイル・サイズに基づいてイベント・ファイルを分割する  
com.ibm.j2ca.extension.utils.filesplit.SplitBySize クラス。

オプションで、カスタムのファイル分割クラスを指定し、そのクラス名を SplittingFunctionClassName プロパティーに入力することによって使用することができます。

区切り文字またはファイル・サイズは、SplitCriteria プロパティーで指定します。EventContentType プロパティーが null に設定されている場合、ファイル・サイズに基づいて分割を実行するクラス名に自動的に設定されます。

表 136. 「分割関数クラス名」プロパティー特性

必須	いいえ
デフォルト	com.ibm.j2ca.extension.utils.filesplit.SplitBySize
プロパティー・タイプ	String
国際化されるかどうか	いいえ

### ポーリング時にエラーが検出された場合はアダプターを停止する (StopPollingOnError)

このプロパティーでは、ポーリング時にアダプターがエラーを検出した場合、アダプターがイベントのポーリングを停止するかどうかを指定します。

表 137. 「ポーリング時にエラーが検出された場合はアダプターを停止する」の詳細

必須	いいえ
使用可能な値	True False
デフォルト	False

表 137. 「ポーリング時にエラーが検出された場合はアダプターを停止する」の詳細 (続き)

プロパティ・タイプ	Boolean
使用法	このプロパティを True に設定した場合、アダプターはエラーを検出するとポーリングを停止します。  このプロパティを False に設定した場合、アダプターはポーリング時にエラーを検出すると例外をログに記録し、ポーリングを続行します。
グローバル化	いいえ
BIDI 対応	いいえ

### 「ローカル・アーカイブ用の成功ファイル拡張子」プロパティ (SuccessArchiveExt)

正常に処理されたビジネス・オブジェクトすべてをアーカイブするために使用されるファイル拡張子。このプロパティは、LocalArchiveDirectory が有効で存在する場合にのみ使用されます。例: 12345.order > 12345.order.success

表 138. 「ローカル・アーカイブ用の成功ファイル拡張子」プロパティ特性

必須	いいえ
デフォルト	success
プロパティ・タイプ	String
国際化されるかどうか	はい

### 「ユーザー名」プロパティ (UserName)

FTP サーバーへの接続特権、および FTP 操作の実行特権を持つユーザーの名前。EventDirectory プロパティで指定した URL にユーザー名が含まれている場合は、このプロパティに値を指定する必要はありません。

表 139. 「ユーザー名」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

### 「イベント DataSource に接続するのに使用されるユーザー名」プロパティ (UserName)

データ・ソースからデータベース接続を取得するためにイベント・パーシスタンスが使用するユーザー名。

表 140. 「イベント DataSource に接続するのに使用されるユーザー名」プロパティ特性

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
国際化されるかどうか	はい

---

## グローバリゼーション

WebSphere Adapter for FTP は、複数の言語および国/地域別環境で使用することができる、国際化されたアプリケーションです。アダプターは、文字セット・サポートおよびホスト・サーバーのロケールに基づいて、メッセージ・テキストを適切な言語で送信します。アダプターは、統合コンポーネント間の双方向スクリプト・データの変換をサポートします。

### グローバリゼーションおよび双方向変換

アダプターは、1 バイト文字セットとマルチバイト文字セットをサポートし、メッセージ・テキストを指定された言語で配信できるようにグローバル化されています。このアダプターは、双方向変換も実行します。これは、同じファイル内で左から右のセマンティック・コンテンツ（ヘブライ語やアラビア語など）と右から左のセマンティック・コンテンツ（URL またはファイル・パスなど）の両方を含むデータを処理するタスクのことです。

### グローバリゼーション

Java 仮想マシン (JVM) 内の Java ランタイム環境は、Unicode 文字コード・セットでデータを表現します。Unicode には、知られているほとんどの文字コード・セット（1 バイトとマルチバイトの両方）の文字エンコードが含まれています。

WebSphere Business Integration システムのコンポーネントは、Java で記述されています。そのため、WebSphere Business Integration のシステム・コンポーネント間でデータを転送するときは、文字変換の必要はありません。

エラー・メッセージや情報メッセージを適切な言語や個々の国や地域に合った形でログに記録するために、アダプターは、稼働しているシステムのロケールを使用します。

### 双方向変換

アラビア語やヘブライ語などの言語は右から左に記述されますが、内部に、左から右に記述されるテキストのセグメントが組み込まれているため、双方向スクリプトとなります。ソフトウェア・アプリケーションで双方向スクリプトを扱う場合は、その表示と処理のためにさまざまな規格を使用します。WebSphere Process Server および WebSphere Enterprise Service Bus は Windows 標準形式を使用しますが、WebSphere Process Server または WebSphere Enterprise Service Bus とデータを交換するエンタープライズ情報システムは別の形式を使用できます。WebSphere Adapter は、2 つのシステム間でやり取りされる双方向スクリプト・データの変換を行うことによって、トランザクションの両側でデータが正確に処理および表示されるようにします。

### 双方向形式

WebSphere Process Server および WebSphere Enterprise Service Bus では、ILYNN（暗黙、左から右、オン、オフ、公称を表す英語の頭文字）という双方向形式が使用されます。これは Windows で使用される形式です。エンタープライズ情報システムが別の形式を使用する場合、アダプターは、データを WebSphere Process Server または WebSphere Enterprise Service Bus に導入する前に形式を変換します。

双方向形式は、5つの属性で構成されます。双方向プロパティを設定する場合は、これらの各属性に値を割り当てます。属性と設定を以下の表に示します。

表 141. 双方向形式の属性

文字の位置	目的	値	説明	デフォルト設定
1	順序のスキーマ	I または V	暗黙 (論理的) または表示	I
2	方向	L R C D	左から右 右から左 コンテキスト上の左から右 コンテキスト上の右から左	L
3	対称スワッピング	Y または N	対称スワッピングのオン/オフ	Y
4	形状の指定	S N I M F B	テキストの形状を指定する テキストの形状を指定しない 初期形状指定 中間形状指定 最終形状指定 分離形状指定	N
5	数字の形状指定	H C N	ヒンディ語 コンテキスト Nominal	N

アダプターは、データを左から右の論理形式に変換してから WebSphere Process Server または WebSphere Enterprise Service Bus に送信します。

#### 双方向プロパティの使用

複数の双方向プロパティを使用して、コンテンツ・データとメタデータの両方の変換を制御できます。特殊な双方向プロパティを設定して、コンテンツ・データまたはメタデータを双方向変換から除外したり、変換時に特殊な処理が必要なデータを識別することができます。

以下の表に、4つのタイプの双方向プロパティを示します。

表 142. 双方向プロパティのタイプ

プロパティ・タイプ	データ形式変更
EIS	コンテンツ・データ (エンタープライズ情報システムによって送信されるデータ) の形式を制御します。
メタデータ	メタデータ (コンテンツ・データについての情報を提供するデータ) の形式を制御します。
スキップ	変換から除外するコンテンツまたはメタデータを識別します。



表 142. 双方向プロパティのタイプ (続き)

プロパティ・タイプ	データ形式変更
特殊フォーマット	変換処理時に異なる取り扱いが必要な特定のテキスト (ファイル・パスや URL など) を識別します。コンテンツ・データとメタデータのいずれに設定してもかまいません。

双方向変換を制御するプロパティは、以下の 3 つの領域で設定できます。

- **リソース・アダプター・プロパティ:** これらのプロパティは、デフォルト構成を保管します。これには、アダプター・インスタンスで双方向変換を実行するかどうかを制御する TurnBiDiOff プロパティが含まれます。これらのプロパティを構成するには、サーバーの管理コンソールを使用します。
- **管理 (J2C) 接続ファクトリー・プロパティ:** これらのプロパティは、エンタープライズ情報システムとの Outbound 接続インスタンスを作成するためにランタイムに使用されます。管理接続ファクトリー・プロパティは作成後、デプロイメント記述子に格納されます。
- **アクティベーション・スペック・プロパティ:** これらのプロパティには、メッセージ・エンドポイントの Inbound イベント処理構成情報が保持されています。外部サービスを実行するときに設定するか、サーバーの管理コンソールを使用します。

### ビジネス・オブジェクトの注釈

一部のアダプターでは、ビジネス・オブジェクト内部の双方向プロパティに注釈を付けることができます。これにより、1 つのビジネス・オブジェクトまたはビジネス・オブジェクトの一部の変換を具体的に制御する情報を追加できます。ビジネス・オブジェクト・エディター (WebSphere Integration Developer 内のツール) を使用して、以下のレベルで注釈を追加できます。

- ビジネス・オブジェクト
- ビジネス・オブジェクト・アプリケーション固有の属性
- ビジネス・オブジェクト属性
- ビジネス・オブジェクト属性アプリケーション固有の属性

### プロパティのスコープと検索機構

アダプターの双方向プロパティに値を設定し、ビジネス・オブジェクトに適宜注釈を付けると、アダプターは双方向変換を実行します。実行時には、プロパティ設定の階層の継承と検索機構に依存するロジックを使用します。

リソース・アダプター内で定義したプロパティは階層の最上位になり、その他の領域内で定義したプロパティや、ビジネス・オブジェクト内部で注釈を付けたプロパティは、階層の下位レベルになります。このため、例えば、リソース・アダプターの EIS タイプの双方向プロパティのみに値を設定すると、Inbound (アクティベーション・スペック) トランザクションと Outbound (管理接続ファクトリー) トランザクションのいずれで発生するかにかかわらず、定義された EIS タイプの双方向プロパティを必要とする変換によってそれらの値が継承および使用されます。

ただし、リソース・アダプターとアクティベーション・スペックの両方の EIS タイプ双方向プロパティに値を設定した場合、インバウンド・トランザクションに起因する変換では、アクティベーション・スペックに設定した値が使用されます。

処理ロジックでは、変換時に使用する双方向プロパティの値を、検索機構を使用して検索します。検索機構は、変換の原因となるレベルから検索を開始し、適切なプロパティ・タイプを持つ定義済みの値を対象に、階層の上位に向かって検索を進めます。検出された最初の有効値が使用されます。階層の検索は、子から親の方向にのみ進行します。兄弟は検索の対象になりません。

## 双方向データ変換で使用可能なプロパティ

双方向データ変換プロパティは、アプリケーションまたはファイル・システム、統合ツール、およびランタイム環境の間で交換される双方向スクリプト・データが正しい形式になるように制御します。これらのプロパティが設定されると、双方向スクリプト・データは、WebSphere Integration Developer および WebSphere Process Server または WebSphere Enterprise Service Bus で正しく処理および表示されます。

### 管理 (J2C) 接続ファクトリー・プロパティ

以下の管理 (J2C) 接続プロパティは、双方向変換を制御します。

- FTPURL
- FileName
- StagingDirectory
- SecondServerUsername
- SecondServerPassword
- SecondServerDirectory
- SocksProxyUsername
- SocksProxyPassword
- FileSequenceLog

### アクティベーション・スペック・プロパティ

以下のアクティベーション・スペック・プロパティは、双方向変換を制御します。

- EventDirectory
- EventFileMask
- FTPArchiveDirectory
- LocalEventDirectory
- LocalArchiveDirectory
- FTPScriptFileExecutedBeforeInbound
- FTPScriptFileExecutedAfterInbound
- FTPRenameExt
- FailedArchiveExt
- OriginalArchiveExt

- SuccessArchiveExt
- SplitCriteria
- SocksProxyUsername
- SocksProxyPassword

## デプロイメント記述子構成プロパティ

以下のデプロイメント記述子構成プロパティは、双方向変換を制御します。

- EPDataSourceJNDIName
- EPEventTableName
- EPDatabaseUsername
- EPDatabasePassword
- EPDatabaseSchemaName

## ラッパー・ビジネス・オブジェクト・プロパティ

以下のラッパー・ビジネス・オブジェクト・プロパティは、双方向変換を制御します。

- DirectoryPath
- Filename
- ChunkInfo
- FtpServerEventDirectory
- SecondServerDirectory
- SecondServerUsername
- SecondServerPassword
- IncludeEndBODElimiter
- LocalDirectoryPath
- LocalArchiveDirForCreate
- StagingDirectory
- ScriptFileParameters
- SplitCriteria
- ArchiveDirectoryForRetrieve

---

## アダプター・メッセージ

WebSphere Adapter for FTP が発行したメッセージを、以下のロケーションに表示します。

メッセージへのリンク: <http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.wbit.610.help.messages.doc/messages.html>

表示される Web ページには、メッセージ接頭語のリストが示されています。メッセージ接頭語をクリックすると、その接頭部を持つメッセージがすべて表示されます。

- 接頭部 CWYFT を持つメッセージは、WebSphere Adapter for FTP が発行したものです。
- 接頭部 CWYBS を持つメッセージは、アダプター・ファウンデーション・クラス (すべてのアダプターが使用するもの) によって発行されたものです。

---

## 関連情報

以下のインフォメーション・センター、IBM Redbooks、および Web ページには、WebSphere Adapter for FTPの関連情報が記載されています。

### サンプルおよびチュートリアル

WebSphere Integration Developer のサンプル/チュートリアルのオンライン・ギャラリーには、WebSphere Adaptersを使用する際に役立つサンプルとチュートリアルが用意されています。サンプル/チュートリアルのオンライン・ギャラリーにアクセスするには、次のようにします。

- WebSphere Integration Developerの開始時に開くウェルカム・ページからアクセスします。WebSphere Adapter for FTPのサンプルとチュートリアルを表示するには、「取得 (**Retrieve**)」をクリックします。次に、表示されるカテゴリを参照して、必要なものを選択します。
- 以下の Web サイトにアクセスします。<http://publib.boulder.ibm.com/bpcsamp/index.html>

### 情報リソース

- WebSphere Business Process Management の情報リソースに関する Web ページ (<http://www14.software.ibm.com/webapp/wsbroker/redirect?version=pix&product=wps-dist&topic=bpmroadmaps>) には、WebSphere Adapters について理解を深めるために役立つ記事、Redbooks、資料、および教材へのリンクが用意されています。
- WebSphere Adapters のライブラリー・ページ (<http://www.ibm.com/software/integration/wbiadapters/library/infocenter/>) では、あらゆるバージョンの資料へのリンクが利用できます。

### 関連製品の情報

- WebSphere Business Process Management バージョン 6.1.0 インフォメーション・センター。WebSphere Process Server、WebSphere Enterprise Service Bus、および WebSphere Integration Developerの情報が記載されています。  
<http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp>
- WebSphere Adapters バージョン 6.0.2 インフォメーション・センター:  
[http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.wsadapters602.doc/welcome\\_top\\_wsa602.html](http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.wsadapters602.doc/welcome_top_wsa602.html)
- WebSphere Adapters バージョン 6.0 インフォメーション・センター:  
[http://publib.boulder.ibm.com/infocenter/wbihelp/v6rxmx/topic/com.ibm.wsadapters.doc/welcome\\_wsa.html](http://publib.boulder.ibm.com/infocenter/wbihelp/v6rxmx/topic/com.ibm.wsadapters.doc/welcome_wsa.html)
- WebSphere Business Integration Adapters インフォメーション・センター:  
[http://publib.boulder.ibm.com/infocenter/wbihelp/v6rxmx/index.jsp?topic=/com.ibm.wbi\\_adapters.doc/welcome\\_adapters.htm](http://publib.boulder.ibm.com/infocenter/wbihelp/v6rxmx/index.jsp?topic=/com.ibm.wbi_adapters.doc/welcome_adapters.htm)

## developerWorks® リソース

- WebSphere Adapter Toolkit
- WebSphere Business Integration ゾーン (WebSphere business integration zone)

## サポートおよび支援

- WebSphere Adapters テクニカル・サポート: <http://www.ibm.com/software/integration/wbiadapters/support/>
- WebSphere Adapters 技術情報: <http://www.ibm.com/support/search.wss?tc=SSMKUK&rs=695&rank=8&dc=DB520+D800+D900+DA900+DA800+DB560&dtm> 「**Product category**」 リストからアダプターの名前を選択して、「**Go**」をクリックしてください。



---

## 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。

〒106-8711  
東京都港区六本木 3-2-12  
日本アイ・ビー・エム株式会社  
法務・知的財産  
知的財産権ライセンス渉外

**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。** IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、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。

IBM Corporation  
Department 2Z4A/SOM1  
294 Route 100  
Somers, NY 10589-0100  
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せんが、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が、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おりません。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証できません。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。これらの名称はすべて架空のものであり、名称や住所が類似する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のめかしたり、保証することはできません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生した創作物にも、次のように、著作権表示を入れていただく必要があります。(c) (お客様の会社名) (西暦年)。このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。(c) Copyright IBM Corp. \_年を入れる\_。 All rights reserved.

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は、プログラムを使用してアプリケーション・ソフトウェアを作成する際に役立ちます。

一般使用プログラミング・インターフェースにより、お客様はこのプログラム・ツール・サービスを含むアプリケーション・ソフトウェアを書くことができます。

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。

### 警告:

診断、修正、調整情報は、変更される場合がありますので、プログラミング・インターフェースとしては使用しないでください。

---

## 商標

IBM、IBM ロゴ、developerWorks、i5/OS、Redbooks、Tivoli、ViaVoice、WebSphere、および z/OS は、International Business Machines Corporation の米国およびその他の国における商標です。

Java およびすべての Java 関連の商標およびロゴは Sun Microsystems, Inc.の米国およびその他の国における商標です。

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。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

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。



# 索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

## [ア行]

アーカイブ、イベント 18  
アクセシビリティ  
 外部サービス・ウィザード 22  
 管理コンソール 21  
 キーボード 22  
 ショートカット・キー 22  
 IBM アクセシビリティ・センター 22  
 アクティベーション・スペック・プロパティ 171  
 管理コンソールでの設定 107, 112  
 リスト 171  
 アセンブリ・エディター 85  
 アダプター実装 23  
 アダプターのアーキテクチャー 4  
 アダプターの技術情報 199  
 アダプターのパフォーマンス 115  
 アダプター・アプリケーション  
 開始 114  
 停止 114  
 アダプター・アプリケーションの開始 114  
 アダプター・アプリケーションの停止 114  
 アダプター・パターン・ウィザード 2, 47  
 アダプター・メッセージ 197  
 イベントのリカバリー 15  
 イベント・アーカイブ、MVS プラットフォームでのアーカイブ 18  
 イベント・ストア 17  
 インストール、EAR ファイルの 101  
 インターネット・プロトコル・バージョン 6.0 (IPv6) 22  
 エクスポート、モジュールの、EAR ファイルとしての 99  
 エンドポイント 85

## [カ行]

外部サービス接続プロパティ 140, 166  
外部サービス・ウィザード  
 アクセシビリティ 22  
 開始 54  
 始動 76  
 接続プロパティ 76  
 概要 1  
 カスタム・ビジネス・オブジェクト 137  
 カスタム・プロパティ  
 アクティベーション・スペック 107, 112  
 管理接続ファクトリー 105, 110

カスタム・プロパティ (続き)  
 リソース・アダプター 103, 109  
 関数セクター 13  
 管理 (J2C) 接続ファクトリー・プロパティ 145  
 管理コンソールでの設定 105, 110  
 関連情報 198  
 関連製品、情報 198  
 キーボード 22  
 技術概説 4  
 技術情報 3, 129, 199  
 技術情報、WebSphere Adapters 198  
 教育、WebSphere Adapters 198  
 区切り文字 14, 15  
 組み込みアダプター  
 アクティベーション・スペック・プロパティ、設定 107  
 管理接続ファクトリー・プロパティ、設定 105  
 使用時の考慮事項 30  
 説明 28  
 リソース・アダプター・プロパティ、設定 103  
 クラスタ環境  
 説明 31  
 デプロイ 31  
 Inbound 処理 32  
 Outbound 処理 32  
 高可用性環境  
 説明 31  
 デプロイ 31  
 Inbound 処理 32  
 Outbound 処理 32  
 構成  
 トレース 120  
 ロギング 120  
 Performance Monitoring Infrastructure (PMI) 115  
 後方互換性  
 プロジェクト 36  
 プロジェクト交換ファイル 36  
 互換性マトリックス 3

## [サ行]

再試行制限のプロパティ 189  
サポート  
 概要 120  
 セルフ・ヘルプ・リソース 129  
 テクニカル 199  
 サポートされる Outbound 操作 6  
 参照バインディング 67  
 参照バインディング、Inbound 85  
 サンプル 37  
 事前定義ビジネス・オブジェクト 137  
 実装、Java 95

- ショートカット・キー 22
- 紹介 1
- スタンドアロン参照 67
  - 接続 85
- スタンドアロン・アダプター
  - アクティベーション・スペック・プロパティ、設定 112
  - 管理接続ファクトリー・プロパティ、設定 110
  - 使用時の考慮事項 30
  - 説明 28
  - リソース・アダプター・プロパティ、設定 109
- 成果物、生成 72
- 成果物の生成 72
- 成果物の生成、Inbound 88
- 製品概要 1
- セキュア FTP 23
- セキュリティ 23
- 接続 67
- 接続プロパティの設定 57, 76
- 説明 141, 167
- セルフ・ヘルプ・リソース 129
- ソフトウェア要件 3

## [タ行]

- ターゲット・コンポーネント 93
- 対話仕様プロパティ 154
  - 変更 91
- チャンク機能 15
- チュートリアル 37
- 注釈 5
- データ変換フレームワーク: Inbound 14
- データ変換フレームワーク: Outbound 5
- テクニカル・サポート 199
- テスト環境
  - テスト、モジュールの 97
  - デプロイ 93, 96
  - モジュールの追加 96
- デバッグ
  - セルフ・ヘルプ・リソース 129
  - org.xml.sax.SAXParseException 例外 129
  - XAResourceNotAvailableException 例外 128
- デプロイメント
  - オプション 28
  - 環境 93
  - 実稼働環境に 97
  - テスト環境に 93
- トラストストア 23, 26
- トラブルシューティング
  - 概要 120
  - セルフ・ヘルプ・リソース 129
  - org.xml.sax.SAXParseException 例外 129
  - XAResourceNotAvailableException 例外 128
- トレース
  - 管理コンソールによるプロパティの構成 120
- トレース・ファイル
  - 使用可能化 120

- トレース・ファイル (続き)
  - 詳細レベル 120
  - 使用不可化 120
  - ファイル名の変更 122
  - ロケーション 122

## [ナ行]

- 認証
  - 実行時 27
  - 説明 27
- 認証別名 41

## [ハ行]

- バージョン 141, 167
- ハードウェア要件 3
- ハードウェア要件とソフトウェア要件 3
- 配線、コンポーネントの 93
- バススルー処理 15
- パターン 2, 47
- パッケージ・ファイル、アダプターの 121
- バッチ FTP モード 23
- パフォーマンスに関する統計 119
- パフォーマンスのモニター 115
- ビジネス・オブジェクト 20, 131
  - 命名規則 135
- ビジネス・オブジェクト、カスタム 137
- ビジネス・オブジェクト、事前定義 43, 45
- ビジネス・オブジェクトおよびサービスの選択: Outbound 61
- ビジネス・オブジェクト属性プロパティ 136
- ビジネス・オブジェクトの構造 131
- ビジネス・オブジェクトの選択、Inbound 81
- ビジネス・オブジェクトの操作サポート 136
- ビジネス・オブジェクトの命名規則 135
- ビジネス・グラフ 5
- ビジネス・フォールト 123
- 非推奨の機能 33
- 表示名 (DisplayName) 141, 167
- 標準規格の準拠 21
- ファイアウォール 23
- ファイル
  - SystemOut.log ログ・ファイル 122
  - trace.log トレース・ファイル 122
- ファイル分割 15
- フォールト
  - 説明 123
- プロジェクト、作成 54
- プロジェクト交換 (PI) ファイル
  - 更新する、マイグレーションせずに 36
- プロパティ
  - アクティベーション・スペック 107, 112
  - リスト 171
  - 管理 (J2C) 接続ファクトリー 105, 110

プロパティ (続き)  
構成プロパティ  
  Inbound 164  
  Outbound 138  
リソース・アダプター 103, 109  
  Inbound 構成 164  
  Outbound 構成 138  
ベンダー (Vendor) 141, 167

## [マ行]

マイグレーションに関する考慮事項 33  
前のバージョンとの互換性 33  
マトリックス、互換性 3  
メッセージ、アダプター 197  
文字セット 57, 76  
モジュール、作成 43  
モジュールの構成のためのロードマップ 39  
問題判別  
  セルフ・ヘルプ・リソース 129  
  org.xml.sax.SAXParseException 例外 129  
  XAResourceNotAvailableException 例外 128

## [ヤ行]

要件、ハードウェアおよびソフトウェア 3

## [ラ行]

ラッパー・ビジネス・オブジェクト 5  
ランタイム環境  
  デプロイ、EAR ファイルの 97  
  認証 27  
リカバリー機能 15  
リソース・アダプター・アーカイブ (RAR) ファイル  
  インストール、サーバーへの 97  
  説明 97  
リソース・アダプター・プロパティ  
  管理コンソールでの設定 103, 109  
  詳細 142, 168  
例外  
  org.xml.sax.SAXParseException 129  
  XAResourceNotAvailableException 128  
連邦情報処理標準 26  
ロギング  
  管理コンソールによるプロパティの構成 120  
ログ・アナライザー 120  
ログ・ファイル  
  使用可能化 120  
  詳細レベル 120  
  使用不可化 120  
  ファイル名の変更 122  
  ロケーション 122

## A

Adapter for FTP  
  アクセシビリティ 21  
  管理 103  
  標準規格の準拠 21  
Adapter for FTP モジュール  
  インストール、EAR ファイルの、サーバーへの 101  
  エクスポート、EAR ファイルとしての 99  
  開始 114  
  停止 114  
Append 6  
ArchiveDirectoryForRetrieve 156

## C

CEI (Common Event Infrastructure) 118  
Common Event Infrastructure (CEI) 118  
Create 6  
CreateFileIfNotExists 156  
CustomerWrapper 5  
CustomerWrapper ビジネス・オブジェクト 131  
CustomerWrapperBG 5, 131

## D

DataConnectionMode 157  
DefaultObjectName 157  
Delete 6  
DeleteOnRetrieve 157  
developerWorks 199  
developerWorks リソース、WebSphere Adapters 198  
Directory Path 158

## E

EAR ファイル  
  インストール、サーバーへの 101  
  エクスポート 99  
EmbeddedNameFunctionSelector 13  
enableHASupport プロパティ 32  
ExecuteFTPScript 6  
Exists 6

## F

FFDC (First Failure Data Capture) 123  
FileContentEncoding 158  
FileInLocalDirectory 158  
Filename 159  
FilenameFunctionSelector 13  
FileTransferType 159  
First Failure Data Capture (FFDC) 123  
FTPFile 5  
FTPFileBG 5

FTPFileBG ビジネス・オブジェクト 131

## G

GenerateUniqueFile 159

## I

IBM WebSphere Adapter Toolkit 199

ID 141, 167

Inbound イベント処理 11

Inbound 構成プロパティ 164

Inbound 処理 11

IncludeEndBODelimiter 160

IPv6 22

## J

J2EE コンポーネント 67

Java 実装 95

## L

List 6

LocalArchiveDirForCreate 160

LocalArchivingEnabledForCreate 160

LocalDirectoryPath 161

## O

org.xml.sax.SAXParseException 129

Outbound 構成プロパティ 138

Outbound 処理 4

Overwrite 6

## P

Performance Monitoring Infrastructure (PMI)

構成 115

説明 115

パフォーマンスに関する統計の表示 119

PMI (Performance Monitoring Infrastructure)

構成 115

説明 115

パフォーマンスに関する統計の表示 119

## R

RAR (リソース・アダプター・アーカイブ) ファイル

インストール、サーバーへの 97

説明 97

Redbooks、WebSphere Adapters 198

Retrieve 6

RetrieveContentType 162

## S

ScriptFileParameters 162

SecondServerDirectory 156

SecondServerHostName 160

SecondServerPassword 162

SecondServerPortNumber 161

SecondServerProtocol 161

SecondServerUsername 162

Secure Sockets Layer (SSL) 23

ServerToServerFileTransfer 6

SplitByDelimiter 15

SplitBySize 15

SplitCriteria 14, 15, 163

SplittingFunctionClassName 163

SSL 通信 23

StagingDirectory 164

SystemOut.log ファイル 122

## T

trace.log ファイル 122

## W

WebSphere Adapters、バージョン 6.0、情報 198

WebSphere Adapters、バージョン 6.0.2、情報 198

WebSphere Application Server の情報 198

WebSphere Business Integration Adapters の情報 198

WebSphere Business Process Management、バージョン 6.1.0、情報 198

WebSphere Enterprise Service Bus

情報 198

デプロイ 97

WebSphere Extended Deployment 31

WebSphere Integration Developer

開始 43, 45, 54

情報 198

テスト環境 93

WebSphere Process Server

情報 198

デプロイ 97

## X

XAResourceNotAvailableException 128

xsd ファイル 131





Printed in Japan