



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、245 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 2、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ます。

原典： WebSphere® Adapters
Version 6 Release 2
WebSphere Adapter for FTP User Guide
Version 6 Release 2

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1刷 2009.2

© Copyright International Business Machines Corporation 2006, 2008.

目次

第 1 章 WebSphere Adapter for FTP の概要	1
ハードウェア要件とソフトウェア要件	2
Adapter for FTP の技術概要	2
Outbound 処理	2
Inbound 処理	10
ビジネス・オブジェクト	19
WebSphere Application Server 環境変数	21
外部サービス・ウィザード	22
第 2 章 アダプター実装の計画	23
始める前に	23
セキュリティ	23
Secure Socket Layer の構成	23
SFTP のサポートの構成	26
連邦情報処理標準 140 に対応するアダプターの構成	26
サーバー検査	27
公開鍵認証	28
機密のロギングおよびトレースのサポート	29
ユーザー認証	29
デプロイメント・オプション	30
クラスター化された環境の WebSphere アダプター	33
カスタム・パーサー・クラスを使用したアダプターの	
カスタマイズ	35
バージョン 6.2 へのマイグレーション	36
マイグレーションに関する考慮事項	36
マイグレーションの実行	38
マイグレーションしない場合のバージョン 6.0.2.x	
またはバージョン 6.1.x プロジェクトの更新	39
第 3 章 サンプルおよびチュートリアル	41
第 4 章 デプロイメントのためのモジュールの構成	43
モジュールの構成のためのロードマップ	43
認証別名の作成	45
モジュールの作成	47
ビジネス・オブジェクトの定義	48
Outbound 処理におけるビジネス・オブジェクトの	
COBOL コピーブック・ファイルへの変換	51
Inbound 処理での COBOL コピーブック・ファイル	
のビジネス・オブジェクトへの変換	61
WebSphere Application Server 環境変数の定義	71
アダプター・パターン・ウィザードを使用した単純	
サービスの作成	73
外部サービス・ウィザードの開始	83
Outbound 処理のモジュールの構成	84
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	84

データ・タイプおよび操作名の選択	87
データ・バインディングの構成	90
データ・ハンドラーの構成	92
対話仕様プロパティの設定およびサービスの生	
成	97
Inbound 処理のモジュールの構成	100
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	100
データ・タイプおよび操作名の選択	113
データ・バインディングの構成	115
データ・ハンドラーの構成	117
サービスの生成	121

第 5 章 アセンブリー・エディターによる対話仕様プロパティの変更 123

第 6 章 モジュールのデプロイ 125

デプロイメント環境	125
テスト用のモジュールのデプロイ	125
Inbound 処理をテストするためのターゲット・コ	
ンポーネントの生成および接続	125
サーバーへのモジュールの追加	127
テスト・クライアントを使用した Outbound 処理	
用モジュールのテスト	128
実稼働のためのモジュールのデプロイ	129
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)	129
EAR ファイルとしてのモジュールのエクスポート	
	131
EAR ファイルのインストール	132

第 7 章 アダプター・モジュールの管理 135

組み込みアダプターの構成プロパティの変更	135
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	135
組み込みアダプターの Managed (J2C)	
Connection Factory プロパティの設定	137
組み込みアダプターの活動化仕様プロパティの	
設定	139
スタンドアロン・アダプターの構成プロパティの	
変更	141
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	141
スタンドアロン・アダプターの Managed (J2C)	
Connection Factory プロパティの設定	142
スタンドアロン・アダプターの活動化仕様プロ	
パティの設定	144
アダプターを使用するアプリケーションの開始	146
アダプターを使用するアプリケーションの停止	146

Performance Monitoring Infrastructure を使用したパフォーマンスのモニター	147
Performance Monitoring Infrastructure の構成	147
Common Event Infrastructure (CEI) を使用したトレースの使用可能化	150
パフォーマンスに関する統計の表示	151

第 8 章 トラブルシューティングおよびサポート 153

Log and Trace Analyzer のサポート	153
うるう年例外の取り扱い	154
ServerToServerFileTransfer	154
ロギングおよびトレースの構成	154
ロギング・プロパティの構成	154
ログ・ファイル名およびトレース・ファイル名の変更	156
First Failure Data Capture (FFDC) サポート	158
ビジネス・フォールト	158
XAResourceNotAvailableException	159
org.xml.sax.SAXParseException	160
セルフ・ヘルプ・リソース	161

第 9 章 参照情報 163

ビジネス・オブジェクトの情報	163
ビジネス・オブジェクトの構造	163
命名規則	166

ビジネス・オブジェクト属性プロパティ	167
ビジネス・オブジェクトの操作サポート	167
カスタム・ビジネス・オブジェクト	168
カスタム・ファイル分割	168
Outbound 構成プロパティ	169
アダプター・タイプ・プロパティ	171
リソース・アダプター・プロパティ	173
Managed (J2C) Connection Factory プロパティ	178
Wrapper および対話仕様プロパティ	190
Inbound 構成プロパティ	200
アダプター・タイプ・プロパティ	202
リソース・アダプター・プロパティ	204
活動化仕様プロパティ	209
グローバリゼーション	236
グローバリゼーションおよび双方向変換	236
ビジネス・オブジェクト内の双方向変換	239
双方向データ変換で使用可能なプロパティ	240
アダプター・メッセージ	242
関連情報	242

特記事項 245

プログラミング・インターフェース情報	247
商標	247

索引 249

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。

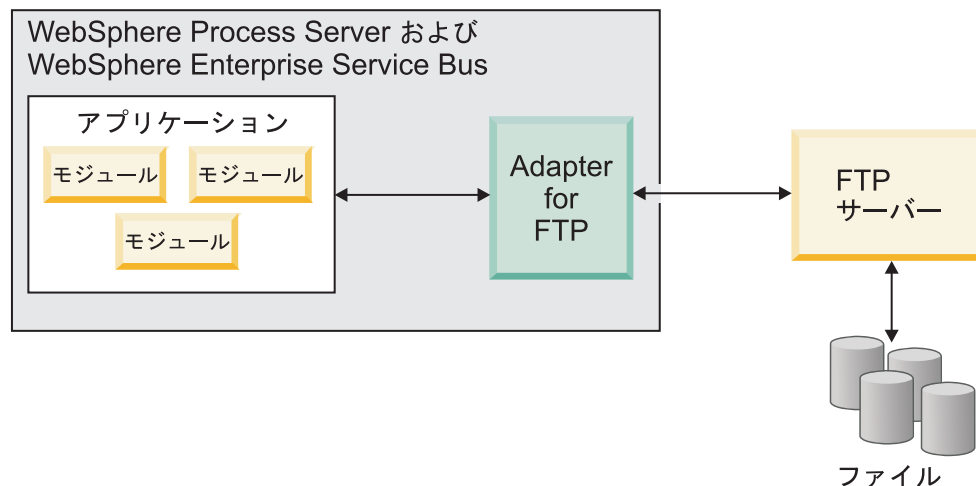


図 1. アダプターの概要

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、IBM Support 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を表示するには、<http://www.ibm.com/support/docview.wss?uid=swg27006249> を参照してください。

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。この文書を表示するには、WebSphere アダプターのサポート・ページを表示して、「追加のサポート・リンク」セクション：
<http://www.ibm.com/software/integration/wbiadapters/support/>の「関連情報」見出しの下にある「互換性マトリックス」をクリックします。
- 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 Integration Developer 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。

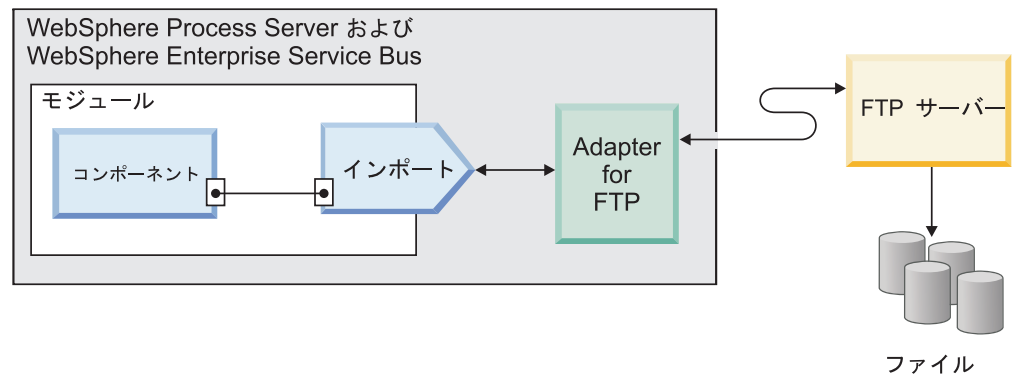


図2. Outbound 処理フロー

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 6.2 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。

- 区切り形式
- 固定幅
- XML

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、要求の一部として送信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>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・ワークステーショ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>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。作成するファイルが既に存在する場合は、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> <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。9 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。7 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>
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に、要求で送信された内容が付加される。</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>ステージング・ディレクトリーが指定された場合、付加するファイルが指定の出力ディレクトリーからステージング・ディレクトリーにコピーされ、内容がステージング・ディレクトリー内の該当ファイルに付加されます。その後で、追加されたファイルは元のディレクトリーに戻されます。</p> <p>付加するファイルが存在せず、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>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、ファイルが正常に削除されたことを示す true 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>

表 1. サポートされる Outbound 操作 (続き)

操作	結果
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 プロパティおよび SplitCriteria プロパティに基づいて、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>ファイル内容が取得された後、それが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。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> <p>アダプターは、DeleteOnRetrieve プロパティを使用してファイルを取得した後に、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。</p> <p>アダプターは、ArchiveDirectoryForDeleteOnRetrieve プロパティを使用して、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>データ変換用に Retrieve 操作を構成中に、CustomerRetrieveWrapper、CustomerRetrieveWrapperBG、OrderRetrieveWrapper、OrderRetrieveWrapperBG のようなカスタム取得ラッパーを作成し、そのラッパーを「操作」ウィンドウで出力タイプとして使用します。</p> <p>データ変換を伴わない Retrieve 操作の場合、デフォルトのラッパー RetrieveResponseWrapper が使用されます。</p> <p>注: 後方互換性には、データ変換を伴う XML の取得に、RetrieveResponseWrapper を使用できます。</p>
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>内容が上書きされたのち、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が指定のディレクトリーからステージング・ディレクトリー (指定した場合) にコピーされ、ステージング・ディレクトリー内の該当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>上書きするファイルが存在せず、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>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。同じ名前のファイルが複数のディレクトリーに存在する場合、アダプターは、呼び出し側コンポーネントに true を返すと共に、最初に検出したファイルの絶対パスを返します。</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>

表 1. サポートされる Outbound 操作 (続き)

操作	結果
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は呼び出し側コンポーネントに返される。</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>指定されたファイルは、1 つの FTP サーバー・ディレクトリーから別の FTP サーバー・ディレクトリーに転送される。ファイルが正常に転送された後、呼び出し側コンポーネントに true が返されます。</p> <p>両方の FTP サーバーが ServerToServerFileTransfer 操作をサポートし、FTP サーバーとアダプターが稼働中のワークステーションとの間で接続が確立される必要があります。</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> <p>注: ServerToServerFileTransfer 操作は、SFTP プロトコルではサポートされていません。</p>
ExecuteFTPScript	<p>FTP スクリプト・ファイルに含まれているコマンドは、アダプター・ワークステーションで実行される。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、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す。</p> <p>注: スクリプト・ファイルには、選択されたプロトコルでサポートされているコマンドが含まれます。</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。

シーケンス・ファイルは、シーケンス番号の保管に使用されるファイルです。アダプターは、現在の操作のためのこのファイルにあるシーケンス番号を取得し、その

番号を 1 増やしてファイルを更新します。シーケンス・ファイルが作成されるとき、ファイルにはデータは何も入っておらず、アダプターはシーケンス番号 1 から生成し始めます。

アダプターは要求のたびにシーケンス番号を読み取り、その番号を 1 増やして、シーケンス・ファイルを更新します。シーケンス番号は、ターゲット・フォルダーで要求ファイルを作成しているときに使用されます。例えば、非数値であったり、特殊文字で構成されていたり、ゼロまたは負であったりなど、番号が無効な場合、アダプターは、再びシーケンス番号 1 から始めます。アダプターを再始動したときは、アダプターはファイル内の既存のシーケンス番号を使用します。

注: ディレクトリーまたはファイル名にかかわらず、シーケンス番号が `Outbound Create` 操作に使用されるシーケンス・ファイルの唯一の内容です。

`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。シーケンス番号の形式は、`$FILENAME.$SEQUENCE_NUMBER.$FILE_EXT` です。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、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` プロパティーの値はローカル・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` プロパティーの値は、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。アダプターにはシーケンス・ログ・ファイルに対する書き込み権限がなければなりません。書き込み権限がなければ、`IOException` が発生します。

注: ファイル・シーケンス番号は、ファイル内のエントリーを削除するか、ファイルを削除すると、リセットできます。新しいシーケンスは 1 から始まります。`FileSequenceLog` プロパティーおよび `GenerateUniqueFilename` プロパティーの両方が有効な場合、`GenerateUniqueFilename` プロパティー値が優先され、`FileSequenceLog` プロパティーは生成されません。

ファイル・シーケンス名を生成できます。ファイル・シーケンス名を作成するには、次の項目を指定します。

1. シーケンス・ファイル。これは、シーケンス番号が保管されているファイルの絶対パスです。
2. デフォルト・ターゲット・ファイル名

アダプターは、デフォルト・ターゲット・ファイル名にシーケンス番号が付加されたファイル名を生成します。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、`Managed Connection Factory` 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。

アダプターは、前のバージョンとの互換性をサポートするために、次のステップを実行します。

1. アダプターは、シーケンス・ファイルを読み取り、path = sequenceNumber の形式のエントリーを検査します。
2. このようなエントリーがファイルに存在する場合、シーケンス・ファイルには、IBM WebSphere Adapter for FTP バージョン 6.2 でサポートされる形式でデータが含まれています。
3. アダプターは、すべてのエントリーから、有効な最大のシーケンス番号を取得します。
4. この番号は、新規ファイルの作成に使用されます。
5. アダプターは、番号を 1 増やして、その新しい番号でファイルの全体を上書きします。

注: 2 つの別個の Managed Connection Factory が同じシーケンス・ファイルにアクセスしてはなりません。また、2 つの別個のアダプター・インスタンスは、クラスターのパーツでない限り、同じシーケンス・ファイルにアクセスしてはなりません。2 つのアダプター・インスタンスがクラスターのパーツである場合は、共用シーケンス・ファイルにアクセスします。

固有ファイル名の生成

Create 操作は、次の条件の下で、固有ファイル名の生成をサポートします。

- Create 操作で、GenerateUniqueFile プロパティーが true に設定されている場合。

固有ファイル名の生成を制御するプロパティーは、次の 3 つの場所にあります。

- Managed Connection Factory プロパティー (デフォルト・ターゲット・ファイル名およびシーケンス・ファイル・プロパティー)
- 対話仕様プロパティー (デフォルト・ターゲット・ファイル名および固有ファイルの生成プロパティー)
- Wrapper ビジネス・オブジェクト

ビジネス・オブジェクトのプロパティーは、対話仕様のプロパティーより優先されます。対話仕様のプロパティーは、Managed Connection Factory プロパティーよりも優先されます。特定のオブジェクトを異って処理するのではない限り、Managed Connection Factory のプロパティーを使用して、ファイル名の生成を制御します。

FTP サーバーが RFC 1123 に指定されている STOU コマンドをサポートする場合、アダプターはこのサーバー・サポートを使用して、固有ファイル名を生成します。

FTP サーバーが STOU コマンドをサポートしない場合、Adapter for FTP は、固有ファイル名を生成し、それを FTP サーバー上に作成します。アダプターによって作成されるファイルの形式は、F の後に TP と乱数の組み合わせが付いたものになります。番号の範囲は、0 から 99999 の間です。次の例は、この形式を示します。

FTP0、FTP9、FTP729、FTP99999

注: アダプターは、GenerateUniqueFile と StagingDirectory の両方のオプションを同時にはサポートしません。

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。

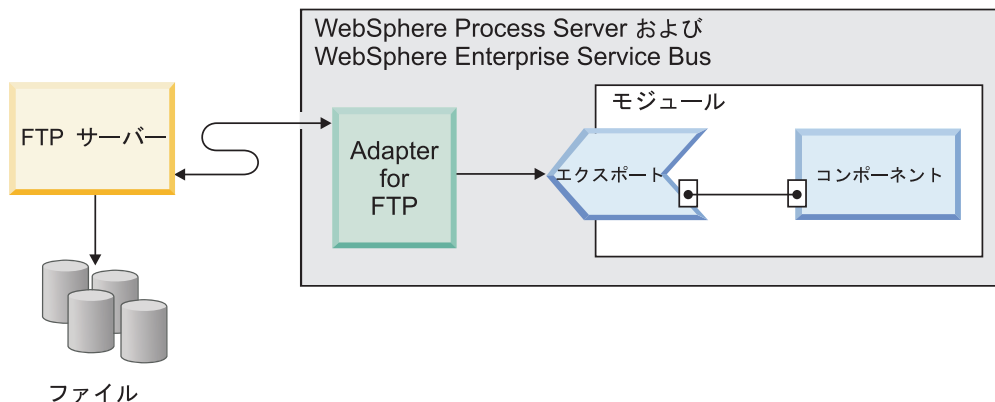


図3. Inbound 処理フロー

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・ワークステーション上のローカル・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[®] 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[®] などの UNIX[®] 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[™] 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。

ルール・ベースのイベントのフィルター処理

アダプターでは、Inbound 処理についてオプションのルール・ベースのイベント・フィルター処理をサポートしています。イベントのフィルター処理は、複数のルールに基づいて実行できます。こうしたルールを組み合わせで定義し、ブール論理でそれらをグループ化し、以下のメタデータによってイベントをフィルター処理することができます。

- FileName
- File Size
- Last Modified

例えば、「*FileName* "MatchesPattern" *.txt」の形を使用できます。*FileName* はプロパティ・タイプ、「MatchesPattern」は演算子、「*.txt」が値です。

ルールはオプションで、イベント・ファイル・マスクは必須ですが、ルールとイベント・ファイル・マスクの両方が指定された場合、イベント・ファイル・マスクよりもルールの方が優先順位は上になります。イベント・ファイル・マスクが有効になるのは、ルールが指定されていない場合に限られます。デフォルトでは、イベント・ファイル・マスクには、デフォルト値として、「*.」が設定されます。

ルール・ベースのフィルター処理では、複数ルール間の論理 OR 演算子はサポートしていません。

注: アダプターでは、EIS が MVS™ プラットフォーム上にある場合、ルール・ベースのフィルター処理はサポートしません。

表2. メタデータのフィルター・プロパティ

プロパティ	有効な演算子	値	前提条件
FileName	MATCHES_FILE_PATTERN	例: *.txt	なし
	MATCHES_REGEX	Java 正規表現	
FileSize	GT (より大)、LT (より小)、GE (より大か等しい)、LE (より小か等しい)、EQ (等しい)、NE (等しくない)	バイト単位の数値。例: FileSize LE 3628927	なし
LastModified	GT (より大)、LT (より小)、GE (より大か等しい)、LE (より小か等しい)、EQ (等しい)、NE (等しくない)	曜日または時刻。曜日値に使用できる演算子は、EQ (等しい) のみです。例:(LastModified EQ MONDAY)、(LastModified EQ 20:41:10)	なし
END-OF-RULE	END-OF-RULE	END-OF-RULE	なし

関数セクター

Inbound 処理中に、関数セクターは、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、関数セクターを選択します。アダプターには、2 つの関数セクター `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 3. `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。

関数セレクターでは、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、関数セレクターは関数名 `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、関数セレクターは、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、関数セレクターは関数名 `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、関数セレ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、関数セレ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 6.2 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。

- 区切り形式
- 固定幅
- XML

参照によるファイルの引き渡し

また、アダプターは PassByReference 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 (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。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$$
```

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。

- 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。
- イベント・ファイル内の各ビジネス・オブジェクト・レコードが有効な区切り文字で区切られていて、最後のビジネス・オブジェクト・レコードに区切り文字が無い、区切り文字が無効な場合でも、アダプターはビジネス・オブジェクト・レコードを処理できます。
- Inbound 処理で区切り文字に基づいてイベント・ファイルを分割するときに、イベント・ファイルにあるビジネス・オブジェクト・レコードが区切り文字で区切られており、この区切り文字が各レコードの終わりではなく先頭にある場合、アダプターは、区切り文字は常に各レコードの先頭にあるとみなし、それに応じて処理します。

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。

例 3:

```
ISAJohnDoe1*IBM*****USA*****  
ISAJohnDoe2*IBM*****USA*****  
ISAJohnDoe3*IBM*****USA*****
```

各レコードの先頭に区切り文字 (ISA) があるイベント・ファイルの内容です。

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、活動化仕様の 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、活動化仕様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 4. 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>NEWEVENT (0) イベントの処理の準備ができています。</p> <p>PROCESSED (1) 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>FAILED (-1) 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

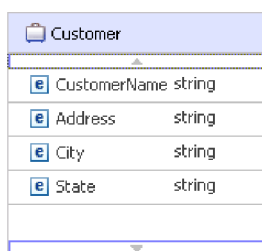
ビジネス・オブジェクトとは、データ、データ上で実行されるアクション、およびデータを処理するための追加の指示 (存在する場合) で構成される構造体のことです。データは、送り状または従業員レコードなどのビジネス・エンティティ、または非構造化テキストのいずれかを表すことができます。

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、最上位レベルの各ビジネス・オブジェクトがビ

ビジネス・グラフに含まれていますが、このビジネス・オブジェクトには、実行する操作に関する追加情報を指定するために、バージョン 6.0.2 でアプリケーションが使用できる動詞が組み込まれています。バージョン 6.1.x では、ビジネス・グラフはオプションです。ビジネス・グラフが必要になるのは、バージョン 6.1.x より前のバージョンの WebSphere Integration Developer を使用して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限られます。ビジネス・グラフが存在する場合、ビジネス・グラフは処理されますが、動詞は無視されます。

WebSphere Application Server 環境変数

外部サービス・ウィザードを使用してアダプターの Inbound または Outbound 処理を構成するときに、ローカルのファイルとディレクトリーに必要な種々の値を設定します。デプロイされたアプリケーションの中のこれらの値は、後で、WebSphere Process Server 管理コンソールから変更できます。

WebSphere Process Server バージョン 6.2 では、ディレクトリーとファイルの値をハードコーディングせずに、それらの値を WebSphere Application Server 環境変数として宣言し、外部サービス・ウィザードを実行するときに環境変数名を指定できます。アプリケーションをデプロイするときに、環境変数名は実際の値に置き換えられ、アダプターによって使用されます。プロパティー値を変更するときは、WebSphere Process Server 管理コンソールの中で、環境変数を変更するだけです。

WebSphere Application Server 環境変数は、Inbound および Outbound 構成の中で設定されたすべてのストリングのプロパティー値 (ブールまたは整数の変数ではない) に対して使用できます。

WebSphere Process Server 管理コンソールを使用して、WebSphere Application Server 環境変数を定義します。

WebSphere Application Server 環境変数を定義するときは、次の項目を指定します。

- 環境変数の名前。例えば、EVENT_DIRECTORY
- シンボル名が表す値。例えば、C:¥ftp¥event
- 環境変数のスコープ。スコープ・レベルによって、管理コンソールに表示される環境変数のレベルが決まります。スコープ・レベルとして、サーバー、ノード、またはセルのレベルがあります。
 - サーバー・スコープで、可視性が指定されたサーバーに制限されます。環境変数を定義する場合、サーバー・スコープが最も詳細に指定できるスコープです。
 - ノード・スコープで、可視性が指定されたノード上のすべてのサーバーに制限されます。これは、デフォルトのスコープです。
 - セル・スコープで、可視性が指定されたセル上のすべてのサーバーに制限されます。

WebSphere Application Server 環境変数の作成方法については、『WebSphere Application Server 環境変数の定義』のトピックを参照してください。

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。

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

セキュア通信のために、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。アダプターもまた、FTP over SSH (SFTP) をサポートします。これは、ポート 22 の SSH チャンネルで実行されるネットワーク・プロトコルです。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 Socket 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 Socket Layer (SSL) を構成すると、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できます。アダプターが Secure

Socket Layer (SSL) を使用してセキュア FTP で動作するように構成されている場合、制御接続とデータ接続の両方が暗号化されます。

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。
- 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。

注: アダプターが SSL バージョン 3.0 を使用してセキュア FTP で動作するのは、暗黙モードの場合のみです。明示モードはサポートされません。アダプターは、次のものをサポートしません。

- SSL バージョン 3.0 を使用してセキュア FTP で動作する場合は、RFC 2228 FTP コマンド
- Transport Layer Security (TLS) バージョン 1.0

注: TLS バージョン 1.0 は、非 FIPS モードではサポートされません。

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」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。

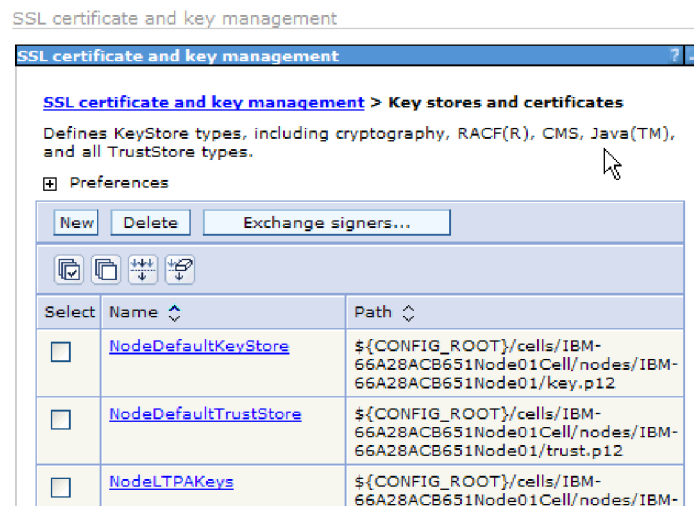


図 5. **NodeDefaultTrustStore** の選択

- f. 「追加プロパティ」の下で、「署名者証明書」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。

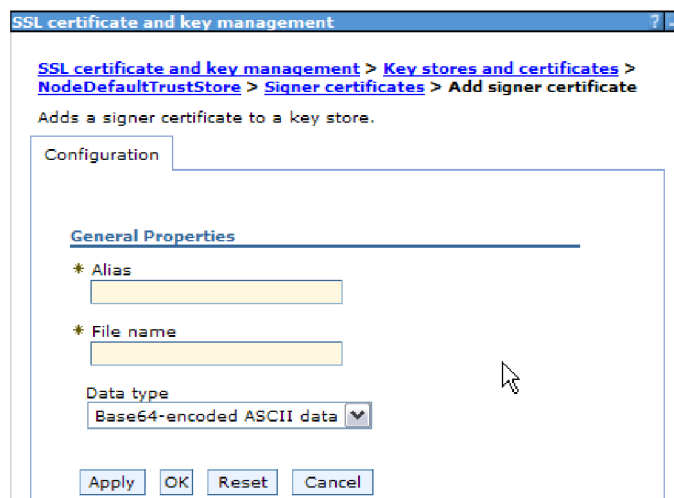


図 6. FTP サーバーの証明書用の署名者証明書プロパティを追加する

- 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。

注:

クラスター化と非クラスター化の両方の WebSphere Process Server 環境では、トラストストアが作成される際には、トラストストア情報を、アダプター固有のトラストストアではなく、WebSphere Process Server によって提供されるデフォルト・トラストストアに追加する必要があります。アダプターは、クラスター化された環境で独自のトラストストアを使用する場合、フェイルオーバー・シナリオ時に開始しません。

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・ウィザードでは、「プロトコル」を「FTPS」 (File Transfer Protocol over Secure Socket Layer (SSL))に設定します。

SFTP のサポートの構成

SSH over FTP (SFTP) は、信頼できるデータ・ストリーム経由のファイル転送メカニズムを提供するネットワーク・プロトコルです。SFTP は、ポート 22 上のセキュアな SSH チャンネル上で稼働し、ユーザー名とパスワード認証または公開鍵認証のいずれかを使用して、すべてのトラフィックを暗号化します。公開鍵認証は、コンピュータが生成した、1 つは公開で、1 つは秘密の、1 対の鍵を使用します。

このタスクを実行する理由および時期

Adapter for FTPをセキュア SSH サーバーで使用するよう構成するには、次のようにします。

このタスクの手順

1. SSH サーバーをインストールして、構成します。さまざまな SSH サーバーから選択できます。プロバイダー特有のインストール情報を使用して、選択したサーバーをインストールして、構成します。
2. Outbound または Inbound の「デプロイメント・プロパティおよびランタイム・プロパティの設定」を参照して、「SFTP - セキュア・シェル (SSH) ファイル転送プロトコル」を選択し、外部サービス・ウィザードで、SFTP サーバー接続とセキュリティ情報を指定します。

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 形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 形式で稼働するには、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。

サーバー検査

サーバー検査は、クライアントが接続を確立する前にサーバー ID を検査する方法です。

アダプターは、SFTP プロトコルが有効な場合にサーバー検査を実行します。アダプターは、接続を確立しようとしている SFTP サーバーがトラステッド・サーバーであるかどうかを検査します。

サーバー検査には、入力としてホスト鍵ファイルが必要です。ホスト鍵ファイルは、トラステッド・サーバーのホスト鍵が追加されたアダプター・ワークステーションで使用可能でなければなりません。ホスト鍵ファイル内のエントリーは、KNOWN_HOSTS ファイルの OpenSSH フォーマットに従っている必要があります。

アダプターは、サーバーが提供するホスト鍵と、ホスト鍵ファイル内のホスト鍵とを比較してサーバーを検査します。サーバーのホスト鍵がホスト鍵ファイルの中に存在する場合に限り、アダプターはサーバーに接続します。トラステッド・サーバーのホスト鍵がホスト鍵ファイルにあるホスト鍵と異なる場合、ホスト鍵ファイル内のホスト鍵のエントリーを変更して、新しいエントリーを反映させる必要があります。

サーバーがトラステッドでない (ホスト鍵がホスト鍵ファイル内にない) 場合、アダプターはサーバーに接続しません。接続要求は失敗し、非トラステッド・サーバーに対する接続が試行されたけれどもセキュリティーの理由により接続が確立されなかったことが示されます。

注: ホスト名と IP アドレスの両方を指定する場合、ホスト鍵エントリーの中に、両方をコンマで区切って指定します。

公開鍵認証

セキュア・シェルを使用するとき、公開鍵認証は、認証に使用される最もセキュアな方式の 1 つです。公開鍵認証は、コンピューターが生成した、1 つは公開で、1 つは秘密の、1 対の鍵を使用します。公開鍵は配布されて、SFTP サーバー上に置くことができます。秘密鍵は、ユーザーに固有で、共用できません。

次のプロパティーには、公開鍵認証が使用可能になっている必要があります。

- ホスト名
- ポート番号
- ユーザー名
- 秘密鍵

パスフレーズは秘密鍵をさらに保護するために使用される、オプションのプロパティーです。

鍵ペアは任意のサード・パーティー・サービスを使用して生成でき、標準の暗号化アルゴリズムのどれかを選択できます。最も一般的に使用されるアルゴリズムは RSA ですが、DSA などの他のアルゴリズムも使用できます。

注: 鍵ペアは OpenSSH フォーマットでなければなりません。

パスワード (ユーザー名とパスワード認証) と秘密鍵 (公開鍵認証) の両方の値が外部サービス・ウィザードに指定されている場合、秘密鍵プロパティー値が優先されます。そのあと、アダプターは、公開鍵認証を使用してサーバーに対して認証を試みます。

機密のロギングおよびトレースのサポート

アダプターは、ログ・ファイルとトレース・ファイルの中の重要または機密データを許可を持たない者が表示できないようにする機能を提供します。

アダプターのログ・ファイルとトレース・ファイルは、FTP サーバーからのデータを含むことができます。これには、重要な情報または機密情報が入っていることがあります。時には、これらのファイルを許可のない個人が調べる必要があり、機密のカスタマー・データが表示されることとなります。例えば、問題をトラブルシューティングするために、ログ・ファイルとトレース・ファイルを使用する必要があるサポート・スペシャリストです。

このようなシチュエーションでデータを保護するために、アダプターは `HideConfidentialTrace` プロパティを提供します。`HideConfidentialTrace` プロパティは、アダプターのログ・ファイルとトレース・ファイルの中にある機密のユーザー・データを表示させるかどうかを指定します。このプロパティが有効な場合、アダプターは、機密データを `XXX` で置き換えます。

次のタイプの情報が基本的に機密データであるとみなされ、隠蔽されます。

- ビジネス・オブジェクトの内容
- イベント・レコードの内容
- ユーザー ID
- 中間形式のビジネス・オブジェクト・データ。例えば、コンマで区切られた形のファイル

次のタイプの情報はユーザー・データであるとはみなされず、隠蔽されません。

- ビジネス・オブジェクト・スキーマ
- トランザクション ID
- イベント ID
- 呼び出しシーケンス

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・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

一環境では、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、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、アプリケーション EAR ファイルに組み込まれたアダプターやサーバーに個別にインストールされたアダプターを含め、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する必要があります。

データ・ソースを使用すると、別のアプリケーションに既に確立されている接続を使用できます。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 2 セキュリティーの Java 認証・承認サービス (JAAS、Authentication and Authorization Service) フィーチャーで作成された J2C 認証データ入力、すなわち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

アダプターをデプロイするには、2 つの方法があります。デプロイされた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、アダプターをスタンドアロン RAR ファイルとしてデプロイできます。選択するデプロイメント・オプションのタイプは、環境要件によります。

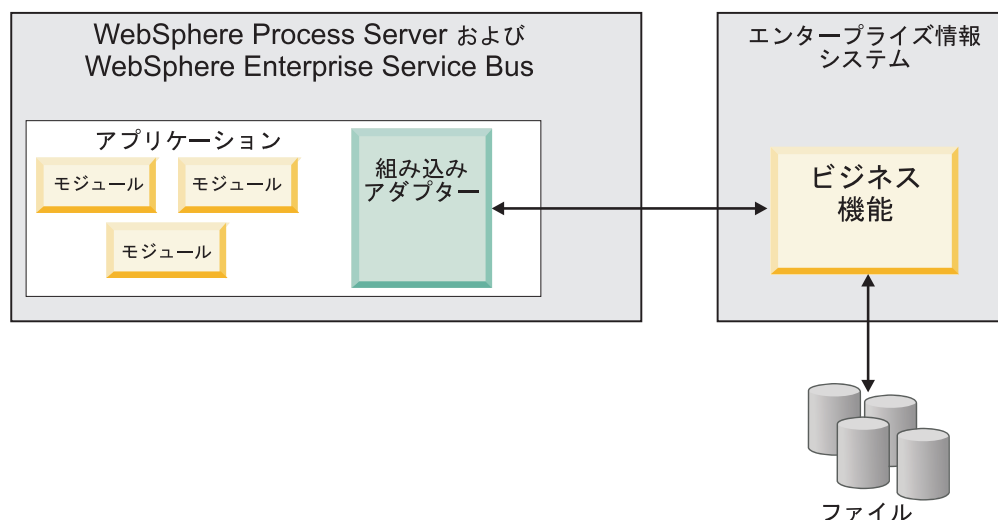
デプロイメント・オプションについて以下に説明します。

- 「**単一アプリケーションが使用するモジュールとともにデプロイする**」。アダプター・ファイルをモジュール内に組み込むと、モジュールをあらゆるアプリケーション・サーバーにデプロイすることができます。組み込みアダプターを使用するのは、組み込みアダプターを使用するモジュールが 1 つある場合か、複数のモジュールでバージョンの異なるアダプターを実行する必要がある場合です。組み込みアダプターを使用すると、他のモジュールのアダプター・バージョンを変更

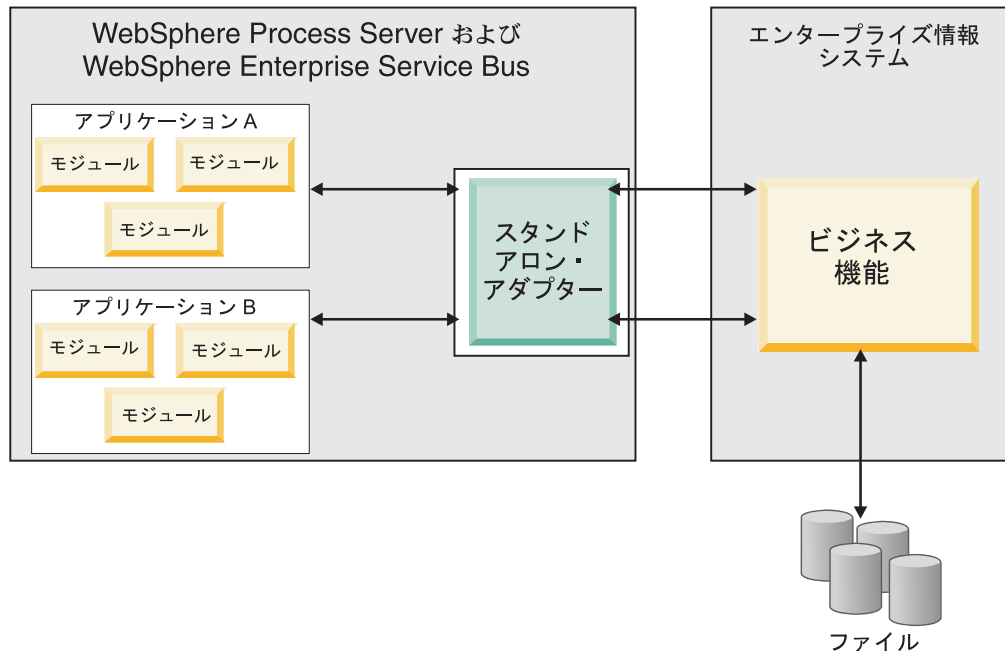
することで、それらのモジュールを不安定にするリスクを生じることなく、1つのモジュール内でアダプターをアップグレードできます。

- 「複数アプリケーションが使用するサーバー上」。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するアプリケーション・サーバーごとにモジュー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。複数のモジュールが同じバージョンのアダプターを使用可能で、アダプターを中央の場所で管理する場合は、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・アダプターの場合も、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り、必要なリソースが軽減されます。

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PLOY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PLOYされた後、サーバー・インスタンス内のすべてのデPLOY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のプロジェクトを作成しているときに、アダプターのパッケージ方法 (EAR ファイルによるバンドルか、あるいはスタンドアロン RAR ファイルとしてか) を選択できます。この選択に応じて、アダプターをランタイム環境で使用方法、および管理コンソールでのアダプターのプロパティの表示の仕方が異なります。

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は、アダプターの管理の仕方によって決まります。アダプターの 1 つのコピーのみを保持して、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が中断してもかまわない場合は、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が多くなります。

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となります。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・ローダーの分離とは、アダプターは別のアプリケーションまたはモジュールからクラスを読み込むことができないという意味です。クラス・ローダーの分離機能により、異なるアプリケーションで、類似した名前の付いた 2 つのクラスによる相互干渉が防止されます。

- 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・アダプターにはクラス・ローダーの分離が存在しないため、ある特定の Java 成果物の 1 つのバージョンのみが実行され、その成果物のバージョンや順序は特定されません。例えば、スタンドアロン・アダプターを使用する場合は、1 つのリソース・アダプター・バージョン、1 つのアダプター・ファウンデーション・クラス (AFC) バージョン、または 1 つのサード・パーティー JAR バージョンのみが存在します。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が動作しなくなることがあります。

- アダプター・ファウンデーション・クラス (AFC) には前のバージョンとの互換性がありますが、単体でデプロイされる各 RAR ファイルには、最新バージョンの AFC を入れておく必要があります。

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

クラスター化された環境の WebSphere アダプター

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 または WebSphere Enterprise Service Bus、WebSphere Application Server Network Deployment、および WebSphere Extended Deployment では、クラスター化された環境がサポートされます。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 Java 2 Platform, Enterprise Edition (J2EE) Connector Architecture (JCA) 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 を参照してください。

必要に応じて、WebSphere Extended Deployment を使用して、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様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、<http://publib.boulder.ibm.com/infocenter/wxinfo/v6r1/index.jsp> のリンクを参照してください。

クラスター化された環境では、アダプター・インスタンスは、Inbound 処理および Outbound 処理の両方を行えます。

制約事項: 各ノードが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場合、Inbound 通信時、WebSphere Adapter for FTP は、WebSphere Process Server または WebSphere Enterprise Service Busのクラスター・バックアップ・ノードとクラスター 1 次ノードの間でポーリングを切り替えることはできません。例えば、アダプターが 1 次 Windows ノードでポーリングを開始した場合、バックアップ UNIX ノードに切り替えることはできません。進行中のイベントを保管するディレクトリーに使用される Windows パスを処理できないためです。

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: クラスター化された環境では、イベント・ディレクトリーをファイル共有システム上に配置し、いずれのクラスター・マシンに対してもローカルにならないようにしてください。

モジュールをクラスターにデプロイすると、JCA コンテナにより、enableHASupport リソース・アダプター・プロパティが検査されます。enableHASupport プロパティ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 要求について 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は、クラスター化された環境にモジュールをデプロイすることにより、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 要求が、顧客の住所などの同じレコードを処理しようとした場合、WebSphere Application Server Network Deployment のワークロード管理機能により、その要求は、受信された順に使用可能なアダプター・インスタンスの間で分配されます。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 を参照してください。

カスタム・パーサー・クラスを使用したアダプターのカスタマイズ

WebSphere Adapter for FTP は Apache Commons Net API v1.4.1 を使用して FTP サーバーに接続します。Commons Net API は、カスタム・パーサー・クラス名プロパティーを通してインターフェースを提供します。そのインターフェースをインプリメントすると、長いリスト (ls -l) 出力を解析できます。

これは、ls -l 出力が標準出力から外れている FTP サーバーにアダプターを接続しなければならない場合に役に立ちます。このインターフェースをインプリメントするクラスを使用することで、アダプターはこのような FTP サーバーを扱うことができます。

アダプターは、このインターフェースの基本的なインプリメンテーションを提供します。クラス名は com.ibm.j2ca.ftp.util.FTPLongListEntryParser です。このクラスは、Windows[®] 2000 サーバーで稼働する FTP サーバーでテストされています。

カスタム・パーサー・クラス名プロパティーは、活動化仕様および Managed Connection Factory の中に存在し、このクラスの完全修飾名を取ります。

次のメソッドは Java[™] インターフェースに存在します。

```
package org.apache.commons.net.ftp;
public interface FTPFileEntryParser{
FTPFile parseFTPEntry(String listEntry);
String readNextEntry(BufferedReader reader) throws IOException;
List preparse(List original);
}
```

これらの各メソッドの詳細は、Apache Commons Net 1.4.1 の資料の中の <http://commons.apache.org/net/api/index.html>を参照してください。

バージョン 6.2 へのマイグレーション

WebSphere Adapter for FTP のバージョン 6.2 へのマイグレーションを行うことにより、アダプターの前のバージョンから自動的にアップグレードします。さらに、アダプターの前のバージョンを組み込んだアプリケーションをマイグレーションできるため、このアプリケーションは、バージョン 6.2 が備えている機能や処理能力を活用できます。

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 6.2 には、既存のアダプター・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にあたっては、既存のアプリケーションに影響する可能性があるいくつかの要因を考慮する必要があります。

前のバージョンとの互換性

WebSphere Adapter for FTP バージョン 6.2 は、アダプターのバージョン 6.0.2.x およびバージョン 6.1.x 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・バインディングを扱うことができます。

WebSphere Adapter for FTP バージョン 6.2 は、バージョン 6.0.2.x およびバージョン 6.1.x と完全に互換性があるため、WebSphere Adapter for FTP バージョン 6.0.2.x バージョン 6.1.0.x を使用していたアプリケーションは、バージョン 6.2 へのアップグレード後も変更なしで稼働します。ただし、バージョン 6.2 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使用する場合は、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、アダプターのバージョン 6.0.2.x またはバージョン 6.1.x をバージョン 6.2 で置き換え (アップグレードし)、バージョン 6.2 のフィーチャーと機能をアプリケーションでできるようにします。

注: マイグレーション・ウィザードは、コンポーネントを作成することもなければ、バージョン 6.2のアダプターを処理するためにマッパーやメディアーターなどの既存のコンポーネントを変更することはありません。バージョン 6.2 にアップグレードし、いずれかのアプリケーションがバージョン 6.1.x以前のアダプターを組み込んでいる場合、アプリケーションでバージョン 6.2のフィーチャーおよび機能を活用したければ、こうしたアプリケーションを変更する必要があります。

単一モジュール内で成果物のバージョンに関する不整合がある場合、そのモジュール全体が不整合であるというマークが付けられ、マイグレーションに選択できなくなります。バージョンの不整合は、プロジェクト破損の症状である可能性があるため、ワークスペース・ログに記録されます。

WebSphere Integration Developerのバージョンが WebSphere Adapter for FTPのバージョンと一致しない場合、それらの製品は一緒に稼働することはできません。

以下の製品の組み合わせが使用される場合、外部サービス・ウィザードは実行できません。

表 5. サポートされない製品の組み合わせ

WebSphere Integration Developer	WebSphere Adapter for FTP
バージョン 6.2	バージョン 6.1.x
バージョン 6.1.x	バージョン 6.0.2.x
バージョン 6.0.2.x	バージョン 6.2
バージョン 6.0.2.x	バージョン 6.1.x

アップグレードかアップグレード後にマイグレーションかの決定

デフォルトでは、マイグレーション・ウィザードはアダプターをアップグレードし、アプリケーション成果物をマイグレーションするので、アプリケーションはバージョン 6.2のアダプターのフィーチャーおよび機能を使用できます。プロジェクトを選択してアダプターのアップグレードを選択すると、ウィザードは、マイグレーションのために、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.x または バージョン 6.1.x からバージョン 6.2 にアップグレードする場合で、アダプター成果物をマイグレーションしない場合は、マイグレーション・ウィザードの該当するページでアダプター成果物の選択を解除してください。

アダプター成果物を選択しないでマイグレーション・ウィザードを実行すると、アダプターのインストールおよびアップグレードが行われますが、成果物はマイグレーションされないため、アプリケーションはバージョン 6.2 のアダプターの機能を利用できません。

最初にテスト環境でマイグレーション・ウィザードを実行

アダプターのマイグレーションでは、WebSphere Adapter for FTP のバージョン 6.2 を使用するアプリケーションを変更しなければならない場合があるため、アプリケーションを実稼働環境にデプロイする前に、まずマイグレーションを開発環境で実行して、アプリケーションをテスト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。以前のバージョンの WebSphere Adapter for FTP の機能のうち、バージョン 6.1.x および バージョン 6.2 で非推奨になった機能を以下に示します。

- 活動化仕様プロパティ `EventContentType` および `DefaultObjectName`
- `FTPURL Managed Connection Factory` プロパティ
- `FTPFileDataBinding` データ・バインディング
- 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

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 ファイルをバージョン 6.2 にマイグレーションできます。ツールが終了したらマイグレーションは完了するため、プロジェクトで作業したり、モジュールをデプロイしたりできます。

始める前に

『マイグレーションに関する考慮事項』の情報を見直します。

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 または 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: 以下の手順では、WebSphere Integration Developer の Java EE パースペクティブでコネクタ・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説明します。

このタスクの手順

1. 既存のプロジェクトの場合は PI (プロジェクト交換) ファイルを、デプロイ済みアプリケーションの場合は EAR (エンタープライズ・アーカイブ) ファイルを、それぞれワークスペースにインポートします。
2. Java EE 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、「更新」 → 「コネクタ・プロジェクトの更新」を選択します。

注: また、以下の方法でアダプター・マイグレーション・ウィザードを起動できます。

- Java EE パースペクティブで、プロジェクトを右クリックし、「更新」 → 「アダプター成果物のマイグレーション」を選択します。
- 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」を選択して問題を解消します。

Java EE パースペクティブにあるときに、コネクタ・プロジェクトのコンテキスト・メニューからマイグレーション・ウィザードを起動する場合、デフォルトでは、依存関係にある成果物プロジェクトがすべて選択されます。依存関係のある成果物プロジェクトを選択解除すると、そのプロジェクトはマイグレーションされません。あとで、選択解除したプロジェクトをマイグレーションすることもできます。以前にマイグレーション済みのプロジェクト、現行バージョンのプロジェクト、エラーのあるプロジェクトはマイグレーションの対象外であり、選択されません。

4. 「ようこそ」ページに表示されたタスクおよび警告を確認して、「次へ」をクリックします。

5. 「このバージョンのターゲット・アダプターでサポートされないプロパティは、マイグレーション中に除去されます。」というメッセージが表示された、警告のウィンドウが開きます。「OK」をクリックして、次のステップに進みます。
6. 「変更内容の確認」ウィンドウで、マイグレーションする成果物の「+」記号をクリックすると、成果物がマイグレーションでどのように変更されるかをそれぞれ確認できます。
7. 「終了」をクリックして、マイグレーションを実行します。

マイグレーション・プロセスを実行する前に、ウィザードは、マイグレーションの影響を受けるすべてのプロジェクトをバックアップします。プロジェクトは、ワークスペース内の一時フォルダーにバックアップされます。何らかの原因でマイグレーションが失敗したり、マイグレーションが完了する前にマイグレーションを取り消したりした場合、ウィザードは、マイグレーション済みのプロジェクトを削除して、一時フォルダーに保管されているプロジェクトと置き換えます。

マイグレーションが正常に終了したときに、バックアップされていたプロジェクトはすべて削除されます。

8. 問題ビューを参照して、マイグレーション・ウィザードからのメッセージ (先頭に CWPAD というストリングがあるメッセージ) の有無を確認します。
9. 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 ファイルを新規に作成し、WebSphere Process Server または WebSphere Enterprise Service Busにデプロイできます。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、この資料で EAR ファイルについて説明してい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 ファイルは、バージョン 6.2 へ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。

マイグレーションしない場合のバージョン 6.0.2.x またはバージョン 6.1.x プロジェクトの更新

アダプターをバージョン 6.0.2.x またはバージョン 6.1.x からバージョン 6.2 にアップグレードする一方で、アダプター・プロジェクトの成果物をマイグレーションしないことを選択できます。

このタスクを実行する理由および時期

アダプターの内部名はバージョン 6.2 で変更されたため、WebSphere Integration Developer バージョン 6.2 でアダプター・ウィザードを使用するには、新しい名前を使用するように、バージョン 6.0.2.x またはバージョン 6.1.x プロジェクトの成果物を更新する必要があります。バージョン 6.0.2.x またはバージョン 6.1.x 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。次に、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内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. Java EE パースペクティブで、プロジェクト名を右クリックし、「更新」 → 「コネクター・プロジェクトの更新」をクリックします。アダプター・マイグレーション・ウィザードが開きます。
3. 「ようこそ」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の選択を解除し、「終了」をクリックします。
5. 「このバージョンのターゲット・アダプターでサポートされないプロパティは、マイグレーション中に除去されます。」というメッセージが表示された、警告のウィンドウが開きます。「OK」をクリックして、次のステップに進みます。
6. 「クイック・フィックス」ウィンドウで、「参照しているアダプターの名前を変更する (Rename the referenced adapter)」というフィックスが選択されていることを確認し、「OK」をクリックします。
7. エラーが表示されたままの場合は、「プロジェクト」 → 「クリーン」をクリックし、更新直後のプロジェクトを選択して「OK」をクリックします。

結果

これで、プロジェクトを WebSphere Adapter for FTP バージョン 6.2で使用す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 Adapters を使用する際に役立つように、サンプルおよびチュートリアルがビジネス・プロセス・マネージメントのサンプルおよびチュートリアルの Web サイトに掲載しています。

サンプルおよびチュートリアルには、次のいずれかの方法でアクセスできます。

- WebSphere Integration Developer を始動すると表示される「ようこそ」ページ。WebSphere Adapter for FTP のサンプルおよびチュートリアルを表示するには、「取得」をクリックします。表示されたカテゴリーをブラウズして、選択を行います。
- Web 上のビジネス・プロセス・マネージメントのサンプルおよびチュートリアル <http://publib.boulder.ibm.com/bpcsamp/index.html> サイト。

第 4 章 デプロイメントのため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、作成の対象となるビジネス・オブジェクトと、その作成を行うシステムを指定します。

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。

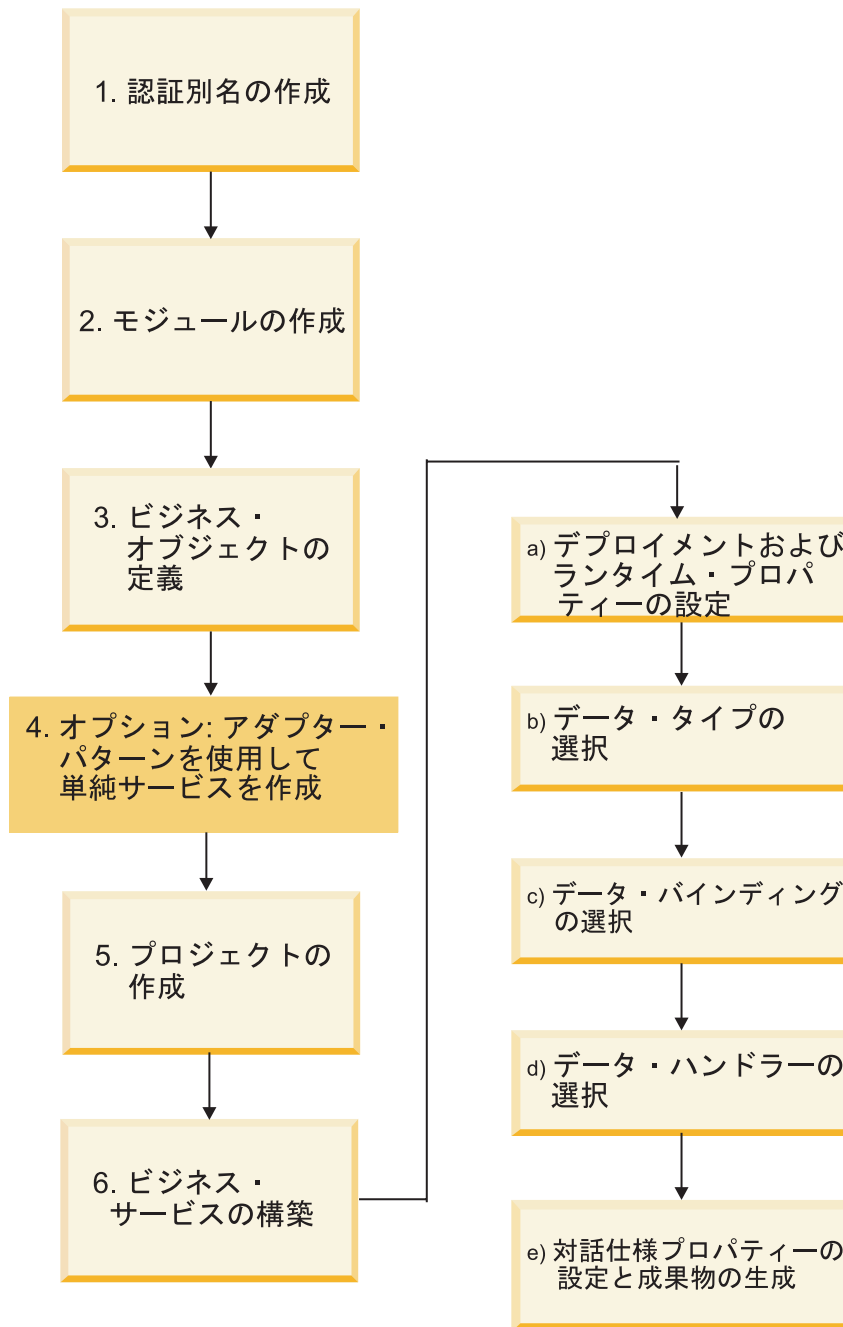


図7. モジュールの構成のためのロードマップ

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、73 ページの『アダプター・パターン・ウィザードを使用した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・プロパティーおよびランタイム・プロパティーを指定します。
 - 接続プロパティー
 - セキュリティー・プロパティー
 - 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ー値を選択し、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。アダプターは、アダプター・プロパティーに保管されたユーザー ID とパスワードを使用する代わりに、これを使用して FTP サーバーに接続できます。

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。また、FTP サーバーに接続するために使用するユーザー名とパスワードを知っておく必要があります。

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

認証別名を使用すると、アダプター構成プロパティ内に平文でパスワードを保管する (他のユーザーに見られる可能性があります) 必要がなくなります。

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. WebSphere Integration Developer の Business Integration パースペクティブで、「サーバー」タブをクリックします。
- b. 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。
- c. 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。
- d. 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」の下で、「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。

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は、指定したノード名および認証別名が含まれます。例えば、ノード widNode で ProductionServerAlias という名前で別名を作成すると、フルネームは widNode/ProductionServerAlias となります。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」ウィンドウの「ビジネス・インテグレーション」ワークスペースのどこかを右クリックし、「新規」→「モジュール」を選択します。

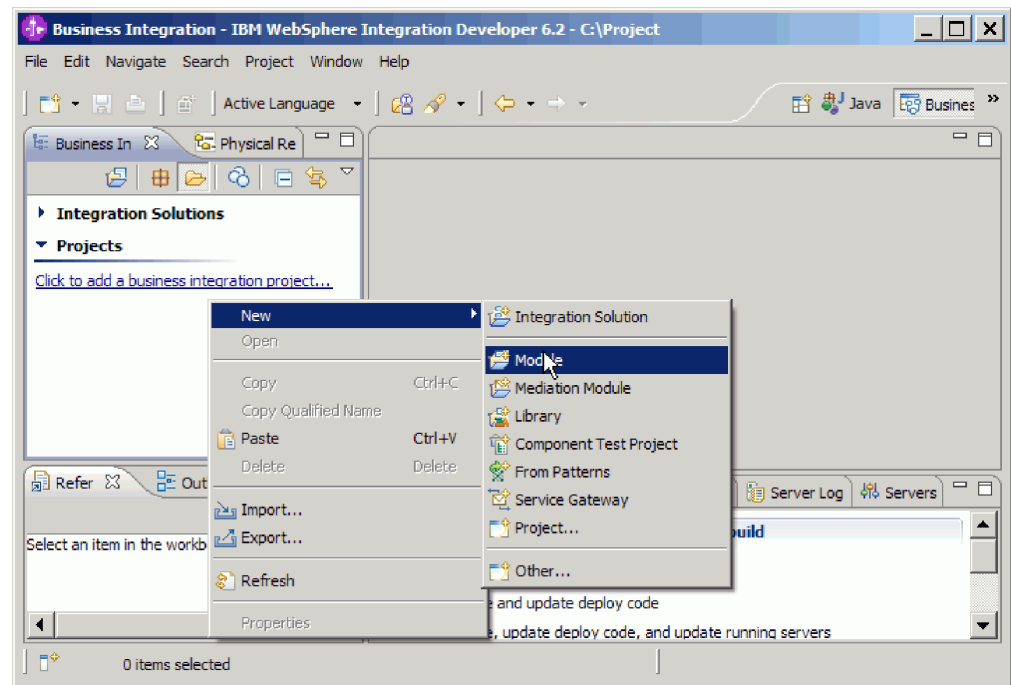


図 8. 「ビジネス・インテグレーション」パースペクティブからの新規モジュールの作成

2. 「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。

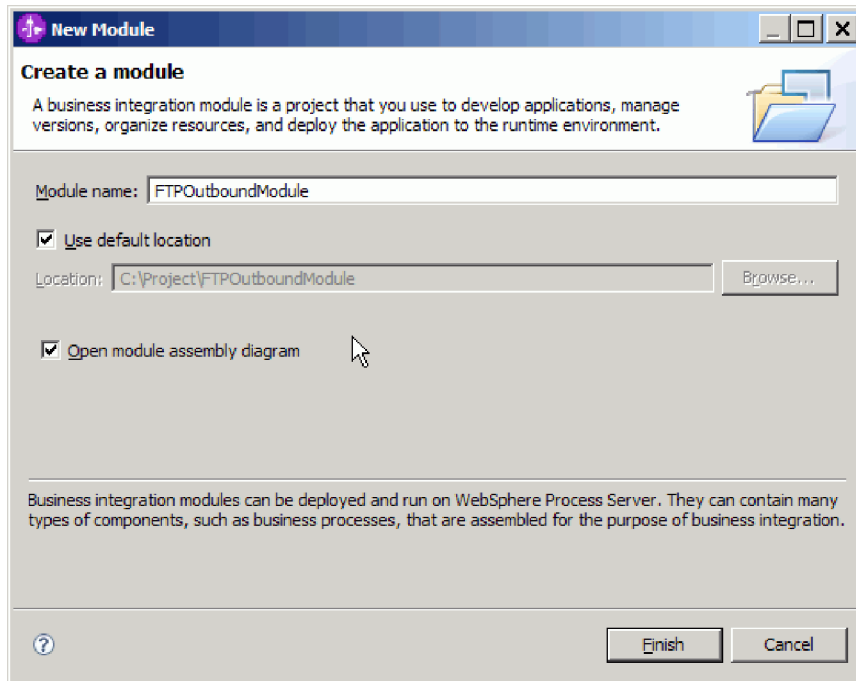


図9. 「新規モジュール」ウィンドウ

3. 「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」フォルダーを右クリックして、「新規」 > 「ビジネス・オブジェクト」を選択します。

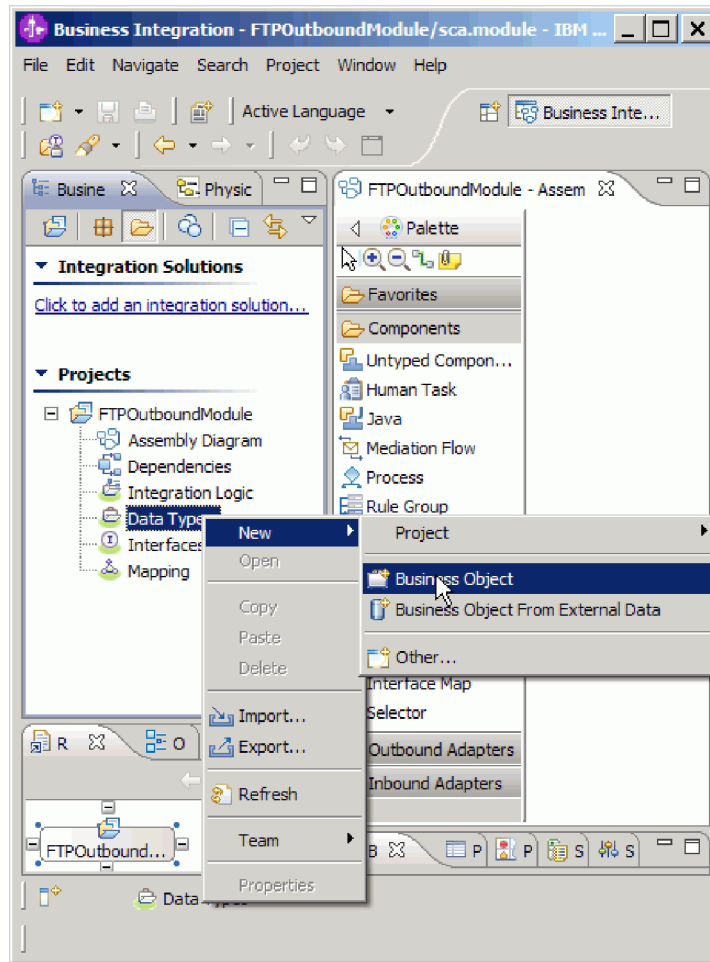


図 10. 新規ビジネス・オブジェクトの作成ビュー

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。

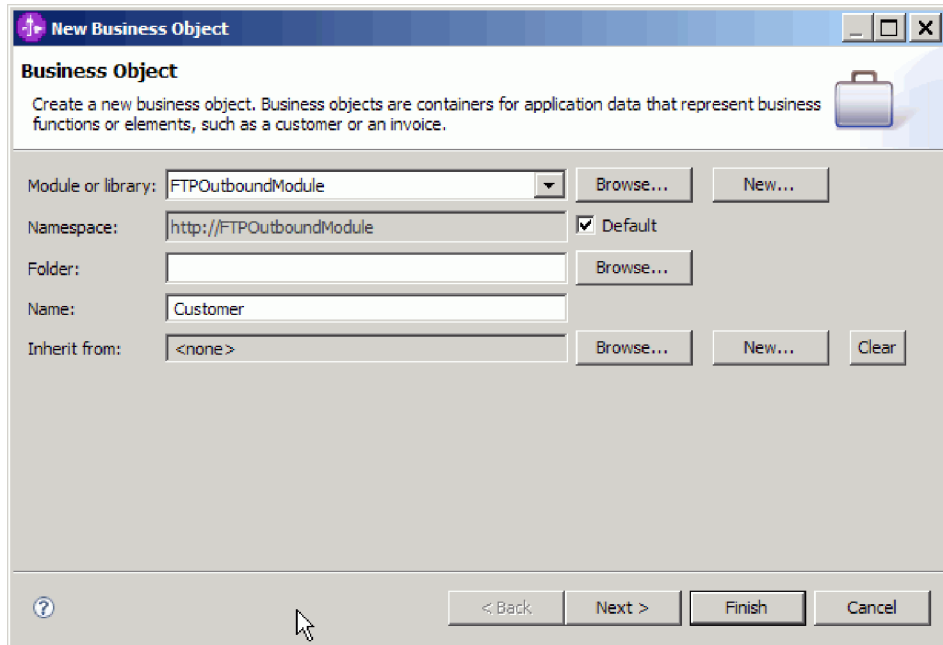


図 11. 「ビジネス・オブジェクト」ウィンドウ

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. 「ビジネス・オブジェクトにフィールドを追加 (Add a field to a business object)」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。

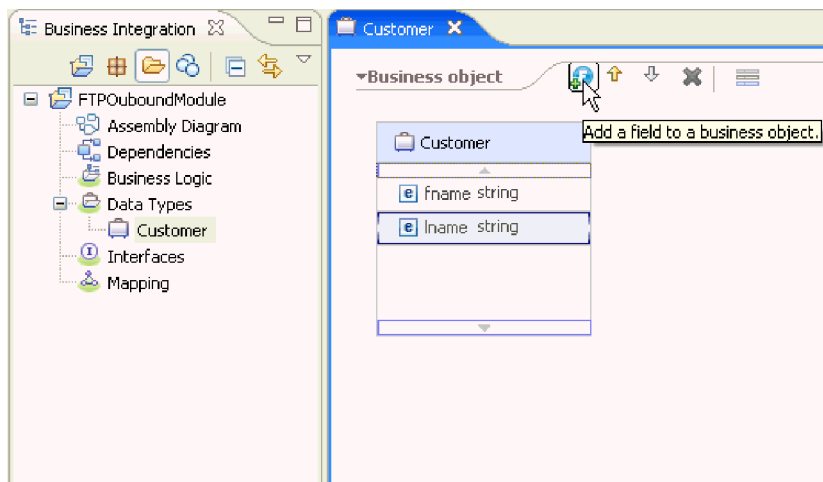


図 12. 「ビジネス・オブジェクト・フィールドの追加 (Add Business object fields)」アイコン

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。

Outbound 処理におけるビジネス・オブジェクトの COBOL コピーブック・ファイルへの変換

WebSphere Integration Developer で外部データ・ウィザードを使用して、COBOL のプログラム・ソース・ファイルからビジネス・オブジェクト定義を生成します。これらのビジネス・オブジェクト定義は、Outbound 処理で使用されます。

始める前に

このタスクを実行する前に、次のことを確認してください。

1. WebSphere Integration Developer でモジュールを作成済みです。
2. COBOL プログラム・ソース・ファイル (.ccp ファイル) は、ワークステーションのローカル・ディレクトリーにあります。
3. Wrapper ビジネス・オブジェクト定義を作成する場合は、アダプター RAR ファイルをワークスペースにインポート済みです。

このタスクを実行する理由および時期

外部データ・ウィザードを使用して、COBOL プログラム・ソース・ファイルのビジネス・オブジェクト定義を生成します。ビジネス・オブジェクト定義を生成した後、再度外部データ・ウィザードを実行して、生成したビジネス・オブジェクトから Wrapper ビジネス・オブジェクト定義を生成できます。

このタスクの手順

1. COBOL プログラム・ソース・ファイルのビジネス・オブジェクトを生成します。
 - a. ウィンドウの「ビジネス・インテグレーション」セクションで、モジュールを右クリックし、「新規」 → 「外部データからのビジネス・オブジェクト」を選択します。
 - b. 「外部データからのビジネス・オブジェクト」ウィンドウで、「言語」を展開して、「COBOL」を選択します。

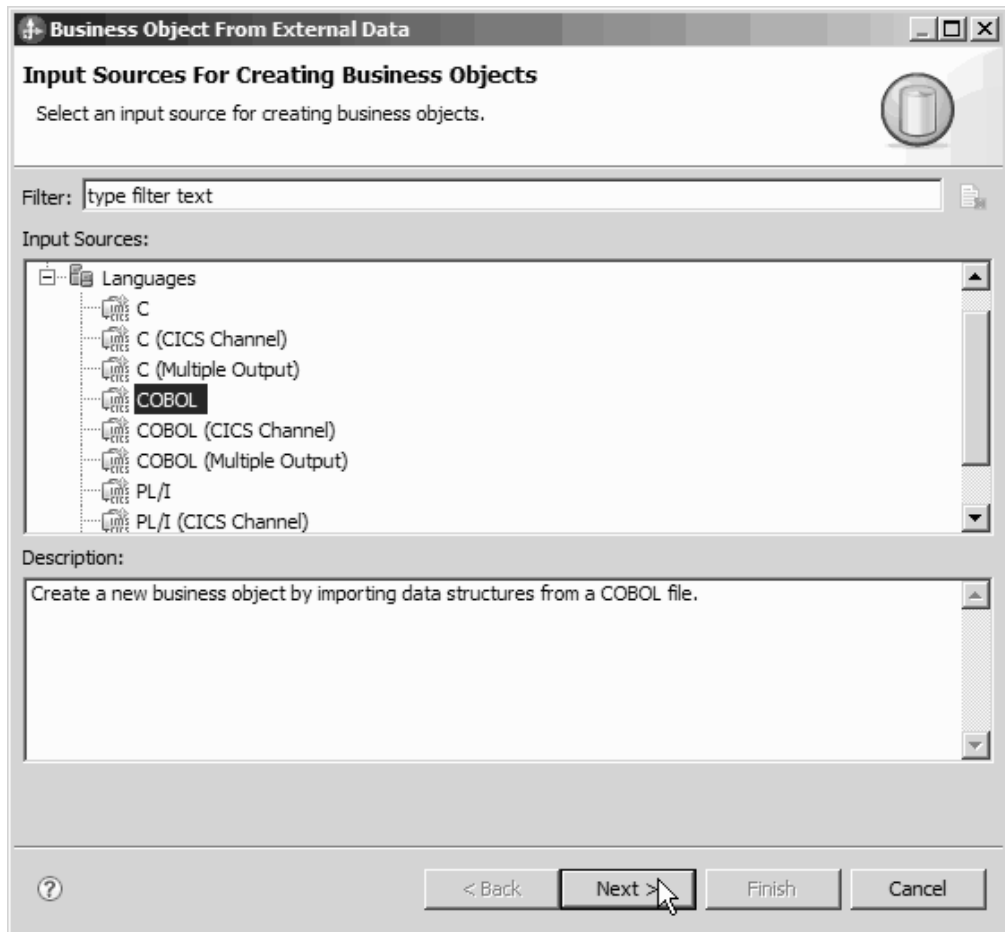


図 13. 「外部データからのビジネス・オブジェクト」ウィンドウ

- c. 「次へ」をクリックします。
- d. 「ビジネス・オブジェクトのマッピング詳細」ウィンドウで、「参照」をクリックして、.ccp ファイルを選択します (この例では、taderc99.ccp を使用)。

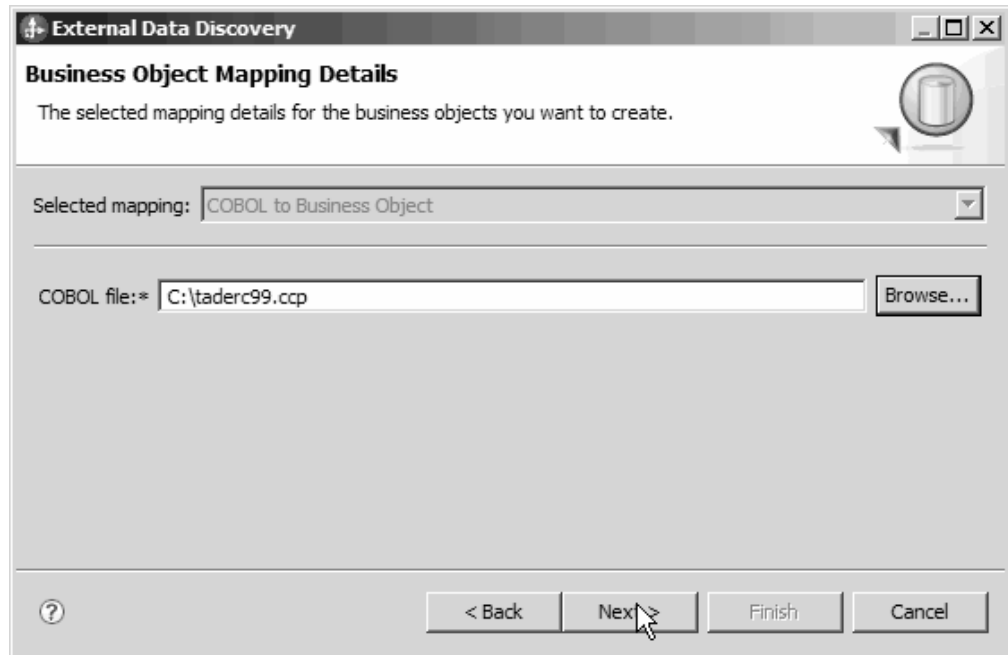


図 14. 「ビジネス・オブジェクトのマッピング詳細 (Business Object Mapping Details)」ウィンドウ

- e. 「次へ」をクリックします。
- f. 「データ構造の選択 (Select Data Structures)」ウィンドウで、「検索 (Find)」をクリックします。DFHCOMMAREA という新規ビジネス・オブジェクトが表示されます。

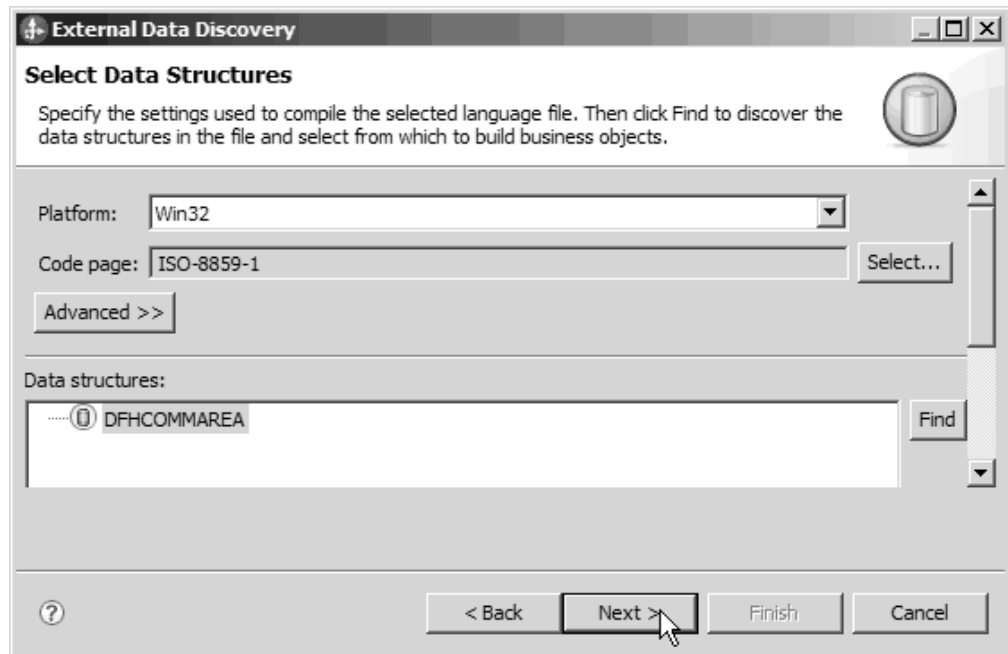


図 15. 「データ構造の選択 (Select Data Structures)」ウィンドウ

- g. 「DFHCOMMAREA」を選択し、「次へ」をクリックします。

h. 「終了」をクリックします。

DFHCOMMAREA という名前のビジネス・オブジェクトがモジュールの中に作成されます。

2. オプション: Wrapper ビジネス・オブジェクト定義を生成します。 Wrapper ビジネス・オブジェクト定義は、追加の機能で既存のビジネス・オブジェクト定義をラップします。 Wrapper ビジネス・オブジェクト定義の生成は、オプションです。 Wrapper ビジネス・オブジェクト定義を生成するオプションは、アダプター RAR ファイルがワークスペースにインポートされている場合に限り表示されません。

注: wrapper ビジネス・オブジェクト定義を生成したい場合は、外部サービス・ウィザードの実行前に生成する必要があります。

- a. ウィンドウの「ビジネス・インテグレーション」セクションで、モジュールを右クリックし、「新規」 → 「外部データからのビジネス・オブジェクト」を選択します。
- b. 「外部データからのビジネス・オブジェクト」ウィンドウで、「アダプター」を展開して、wrapper ビジネス・オブジェクトの生成対象となるアダプター・コネクタ・プロジェクトを選択します。

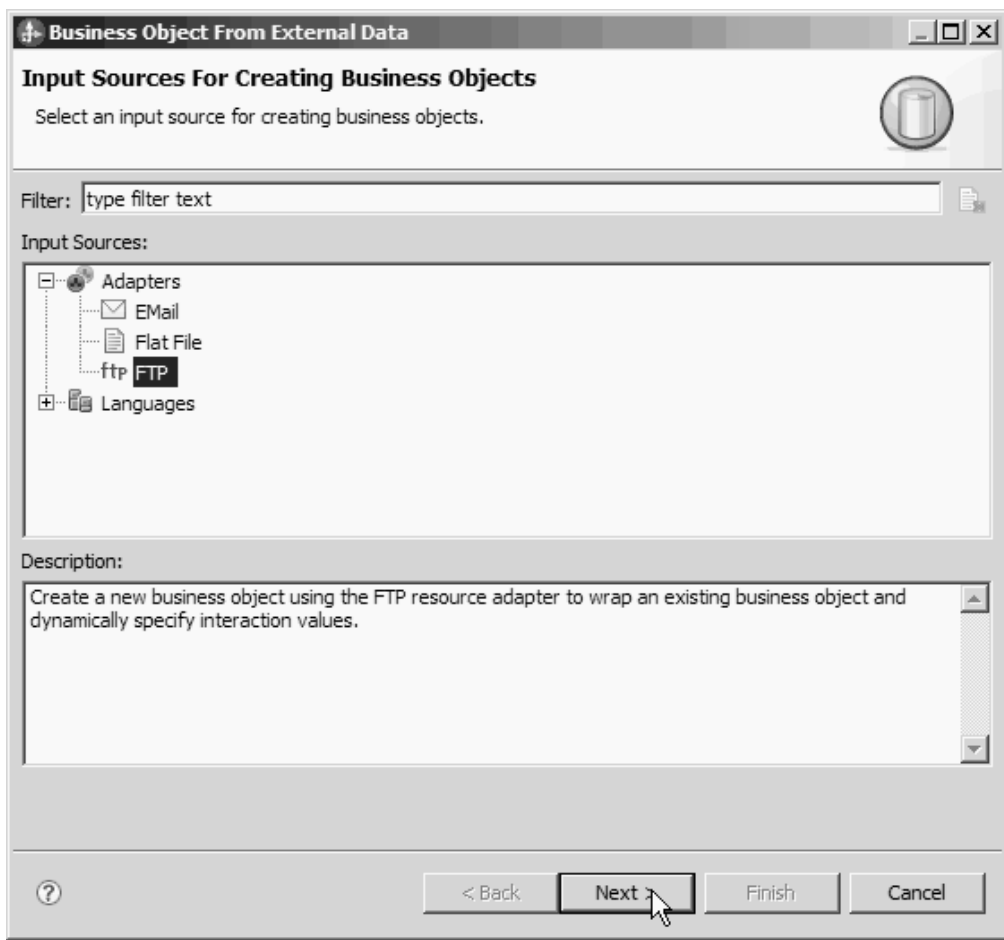


図 16. 「外部データからのビジネス・オブジェクト」ウィンドウ

c. 「次へ」をクリックします。

- d. 「ビジネス・オブジェクト・プロパティ」ウィンドウで、「参照」をクリックし、データ・タイプに DFHCOMMAREA を選択します。
- e. ビジネス・グラフを生成するには、「ビジネス・オブジェクトごとにビジネス・グラフを生成」チェック・ボックスを選択します。取得ラッパーを生成するには、「複数のビジネス・オブジェクトを取得するための取得コンテナを生成」チェック・ボックスを選択します。

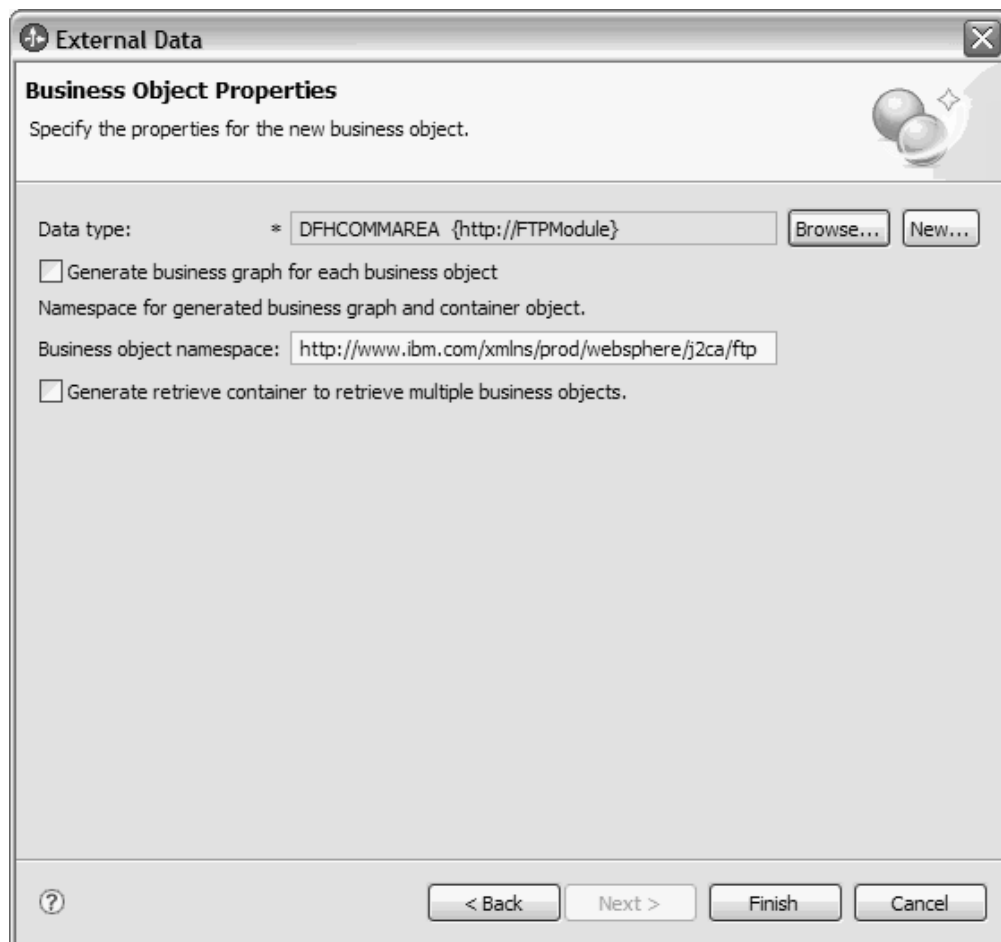


図 17. 「ビジネス・オブジェクト・プロパティ」ウィンドウ

- f. 「終了」をクリックします。

現行モジュールの、DFHCOMMAREARWrapper という Wrapper ビジネス・オブジェクトと、DFHCOMMAREARWrapperBG というビジネス・グラフが、「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。「ビジネス・オブジェクトごとにビジネス・グラフを生成」と「複数のビジネス・オブジェクトを取得するための取得コンテナを生成」を選択すると、現行モジュールの、DFHCOMMAREARRetrieveWrapper というビジネス・オブジェクトと、DFHCOMMAREARRetrieveWrapperBG というビジネス・グラフも、「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。

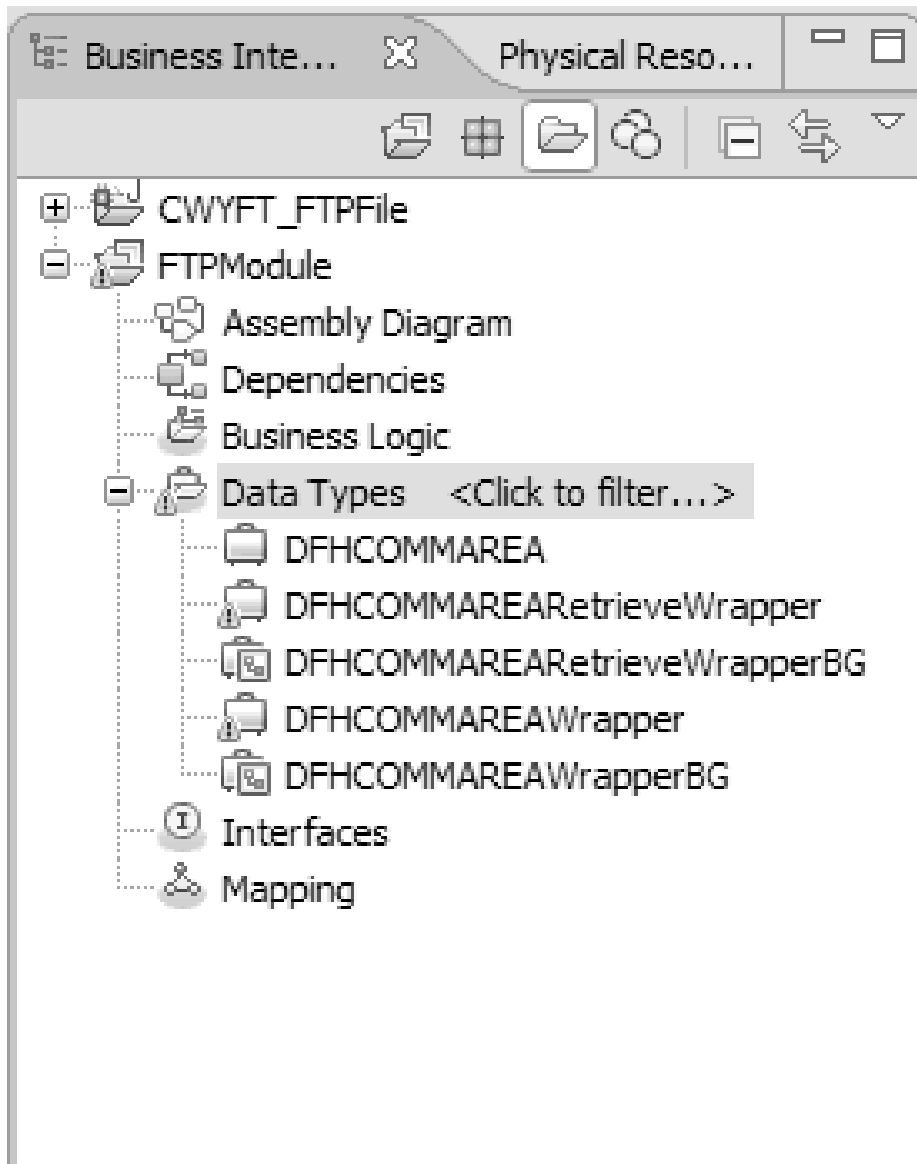


図 18. 「ビジネス・インテグレーション」ウィンドウにリストされた Wrapper ビジネス・オブジェクトとビジネス・グラフ

3. COBOL コピーブック Outbound モジュールの必要な成果物を生成します。この例は、Create 操作の構成を示しています。
 - a. ウィンドウの「ビジネス・インテグレーション」セクションで、モジュールを右クリックし、「新規」 → 「外部サービス」を選択します。
 - b. 「アダプター」を選択して、「次へ」をクリックします。
 - c. 「アダプターの選択」ウィンドウで、FTP アダプター・コネクター・プロジェクトを選択して、「次へ」をクリックします。

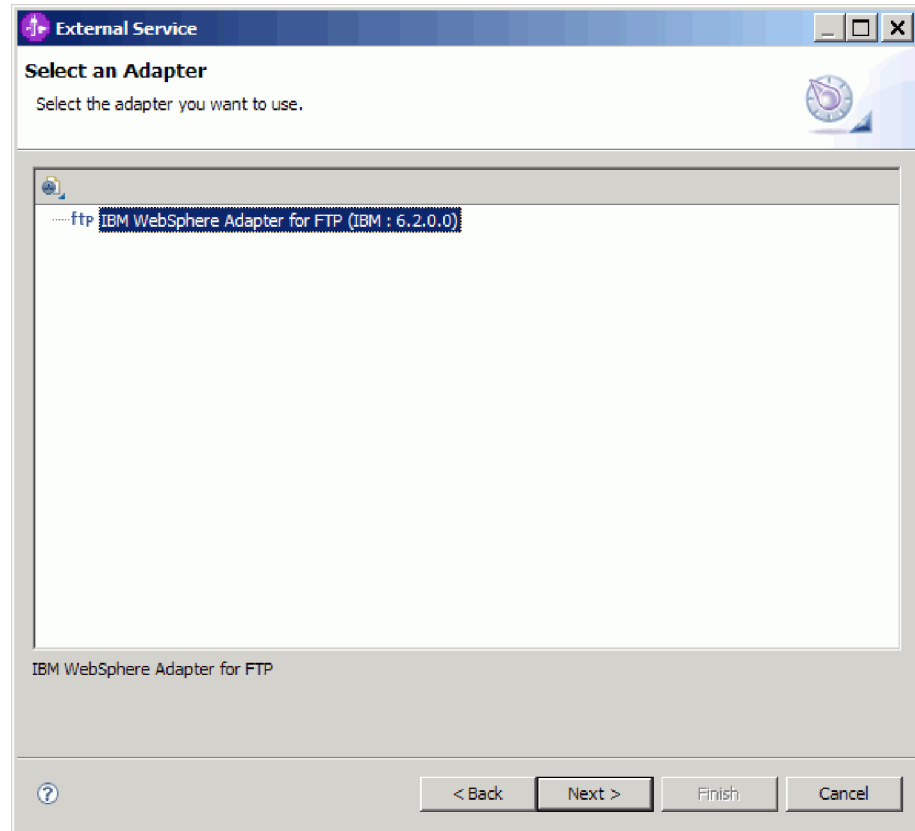


図 19. 「アダプターの選択」ウィンドウ

- d. 「処理方向」ウィンドウで、「**Outbound**」を選択します。
- e. 「次へ」をクリックします。
- f. 「サービス構成プロパティ」ウィンドウの「データ・フォーマット・オプション」リストで、「**COBOL、C、または PL/I のデータ・バインディングを使用 (Use COBOL, C or PL/I data binding)**」を選択します。

注: これはデータ・バインディングではなく、データ・バインディング生成プログラムです。ツールは、現行モジュールの中に適切なデータ・バインディング・コードを生成します。

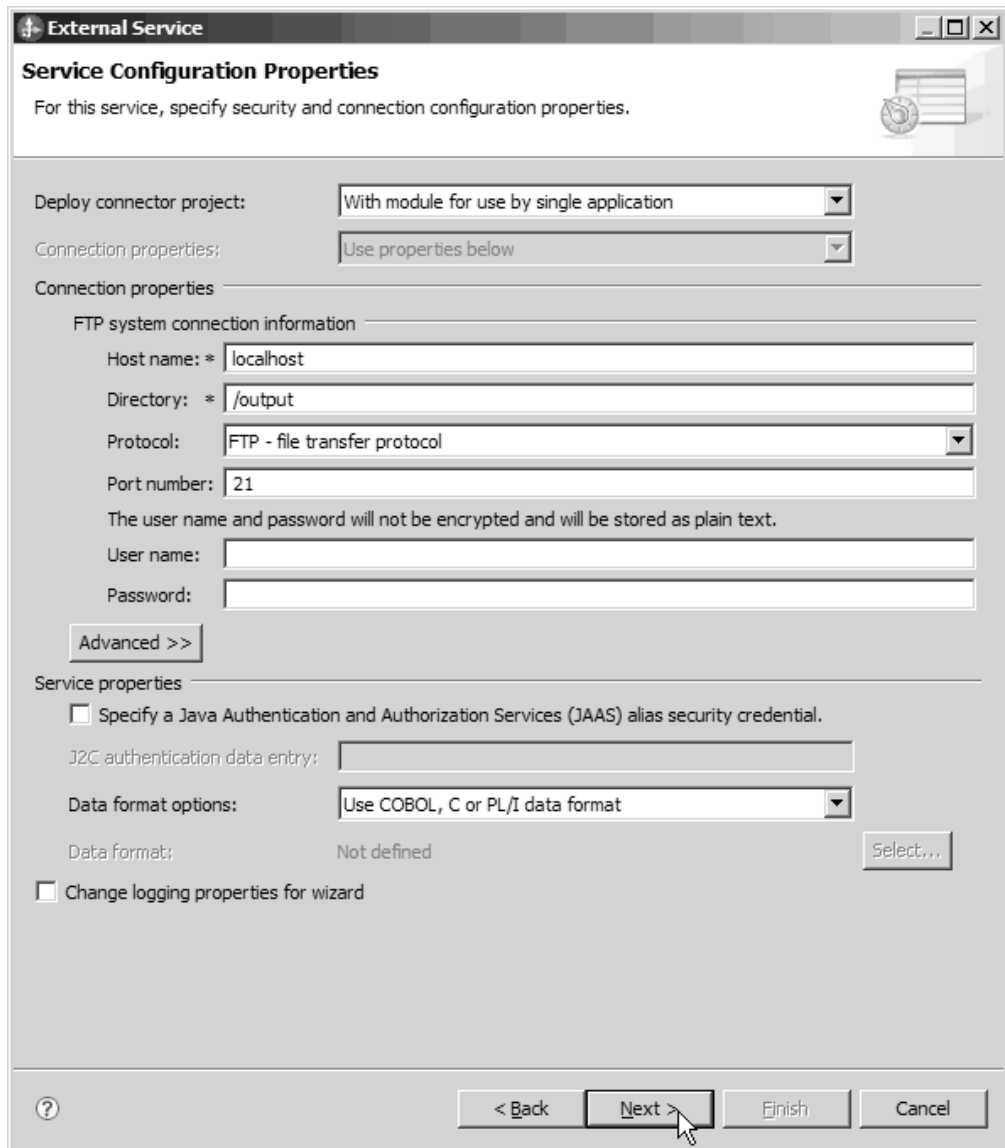


図 20. 「サービス構成プロパティ」ウィンドウ

- g. Outbound 処理に必要な他のプロパティを指定して、「次へ」をクリックします。
- h. 「操作」ウィンドウで、「追加」をクリックし、「Create」をクリックします。Retrieve 操作には、「Retrieve」を選択します。データ・タイプのリストから「ユーザー定義タイプ」を選択して、「次へ」をクリックします。
- i. 入力タイプ (DFHCOMMAREA、DFHCOMMAREAWrapper、または DFHCOMMAREAWrapperBG のいずれか) を参照し、「OK」をクリックします。「Retrieve」操作の場合、適切な出力タイプ (DFHCOMMAREA、DFHCOMMAREARetrieveWrapper、または DFHCOMMAREARetrieveWrapperBG のいずれか) を参照します。



図 21. 「データ・タイプの選択 (Data Type Selection)」 ウィンドウ

j. 「次へ」をクリックします。

COBOL コピーブック、WSDL ファイル、インポート・ファイル、およびその他の成果物によって使用されるデータ・バインディングが生成されます。生成されたデータ・バインディング・クラスは、「Project Explorer」を表示してください。

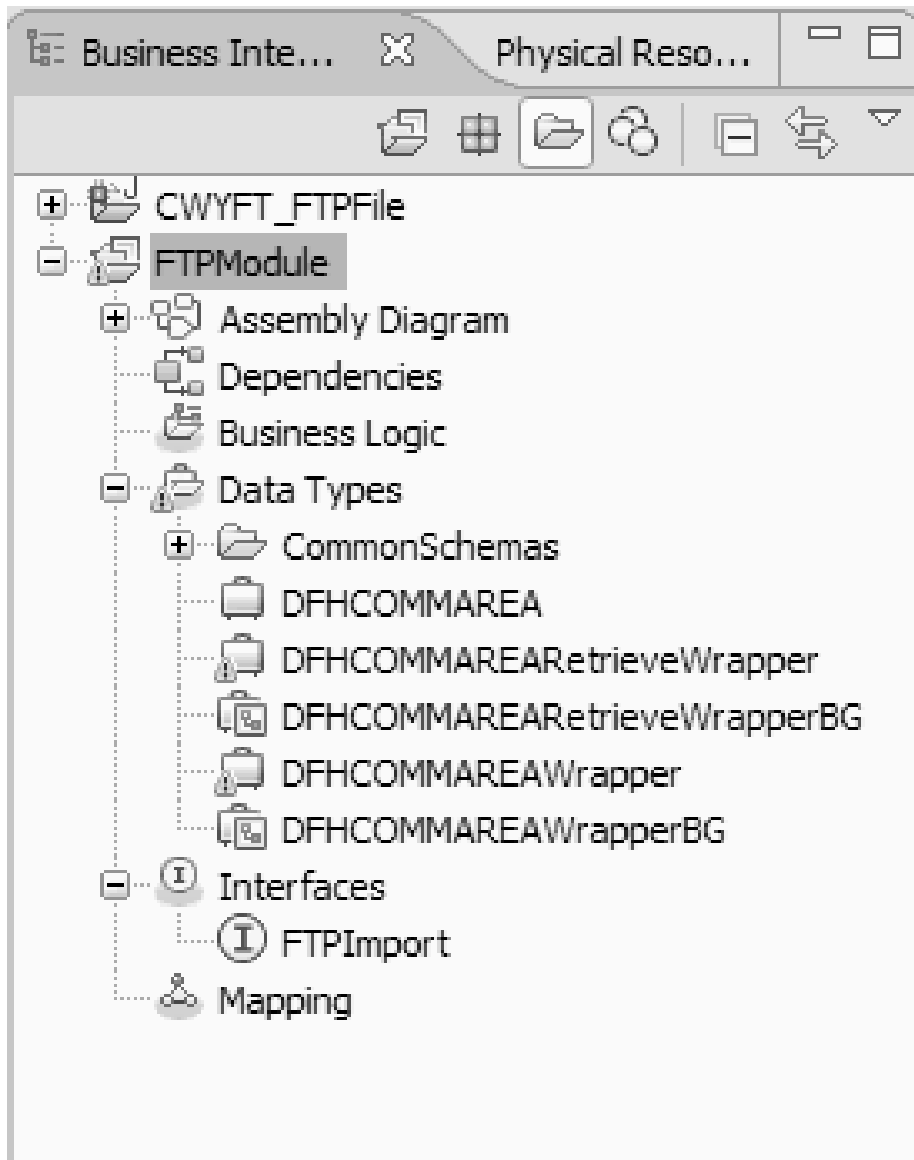


図 22. COBOL コピーブック、WSDL ファイル、インポート・ファイル、およびその他の成果物によって使用されるデータ・バインディング

結果

Outbound モジュールの COBOL プログラム・ソース・ファイル用の、ビジネス・オブジェクト、Wrapper ビジネス・オブジェクト、およびビジネス・グラフが作成されます。COBOL コピーブック・データ・バインディングを使用する Outbound Create 操作の成果物が生成されます。このモジュールは、WebSphere Process Server にデプロイして、Create 操作に対してテストできます。

注: 他のサポートされる操作 (Append と Overwrite) の成果物を生成するには、ステップ 3h から始まる手順に従います。

次のタスク

モジュールをデプロイします。

Inbound 処理での COBOL コピーブック・ファイルのビジネス・オブジェクトへの変換

WebSphere Integration Developer で外部データ・ウィザードを使用して、COBOL のプログラム・ソース・ファイルからビジネス・オブジェクト定義を生成します。これらのビジネス・オブジェクト定義は、Inbound 処理で使用されます。

始める前に

このタスクを実行する前に、次のことを確認してください。

1. WebSphere Integration Developer でモジュールを作成済みです。
2. COBOL プログラム・ソース・ファイル (.ccp ファイル) は、ワークステーションのローカル・ディレクトリにあります。
3. ローカル・イベント・ディレクトリを作成済みです。
4. Wrapper ビジネス・オブジェクト定義を生成する場合、アダプター RAR ファイルをワークスペースにイン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

外部データ・ウィザードを使用して、COBOL プログラム・ソース・ファイルのビジネス・オブジェクト定義を生成します。ビジネス・オブジェクト定義を生成した後、必要であれば外部データ・ウィザードを再実行して、生成したビジネス・オブジェクトから Wrapper ビジネス・オブジェクト定義を生成できます。

このタスクの手順

1. COBOL プログラム・ソース・ファイルのビジネス・オブジェクトを生成します。
 - a. ウィンドウの「ビジネス・インテグレーション」セクションで、モジュールを右クリックし、「新規」 → 「外部データからのビジネス・オブジェクト」を選択します。
 - b. 「外部データからのビジネス・オブジェクト」ウィンドウで、「言語」を展開して、「**COBOL**」を選択します。

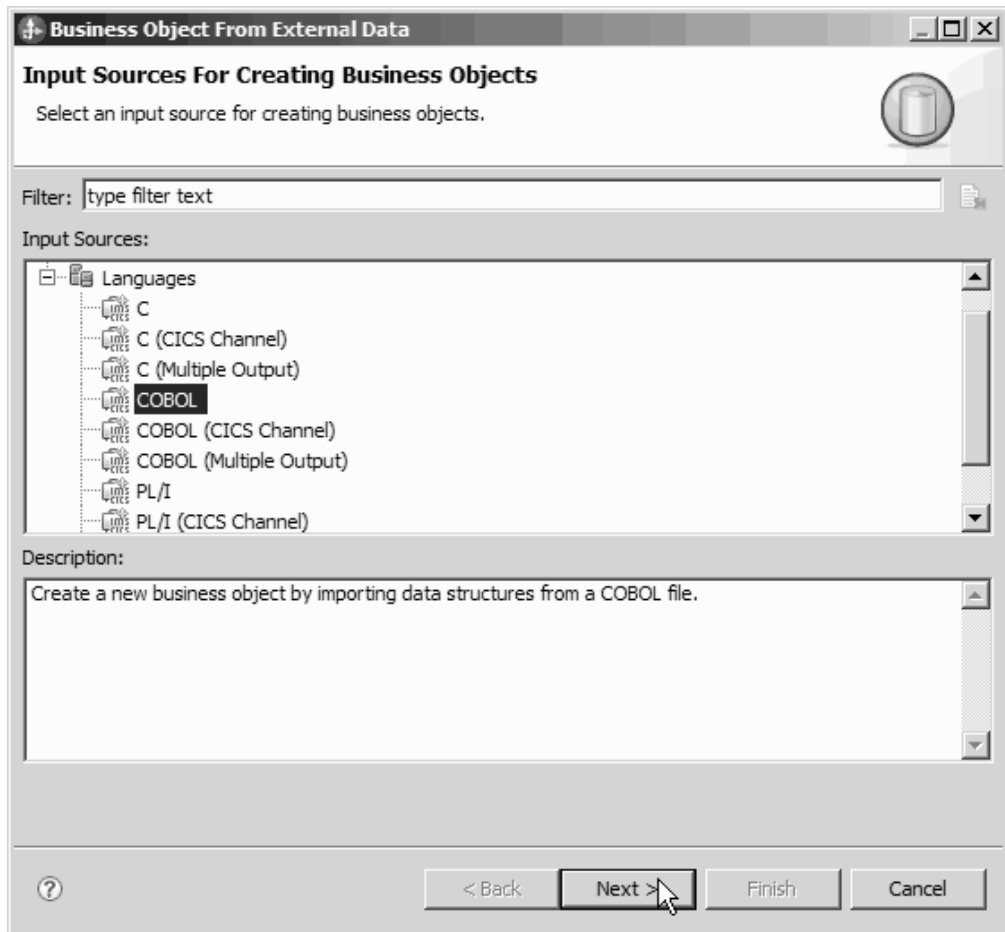


図 23. 「外部データからのビジネス・オブジェクト」ウィンドウ

- c. 「次へ」をクリックします。
- d. 「ビジネス・オブジェクトのマッピング詳細」ウィンドウで、「参照」をクリックして、.ccp ファイル (この例では、taderc99.ccp) を選択します。

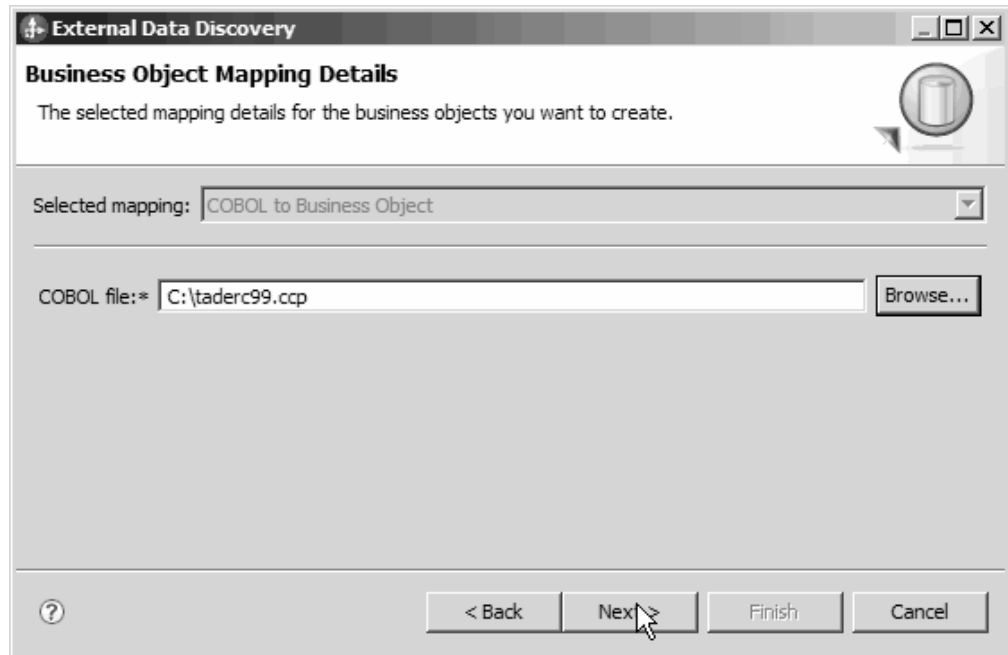


図 24. 「ビジネス・オブジェクトのマッピング詳細 (Business Object Mapping Details)」ウィンドウ

- e. 「次へ」をクリックします。
- f. 「データ構造の選択 (Select Data Structures)」ウィンドウで、「検索 (Find)」をクリックします。DFHCOMMAREA という新規ビジネス・オブジェクトが表示されます。

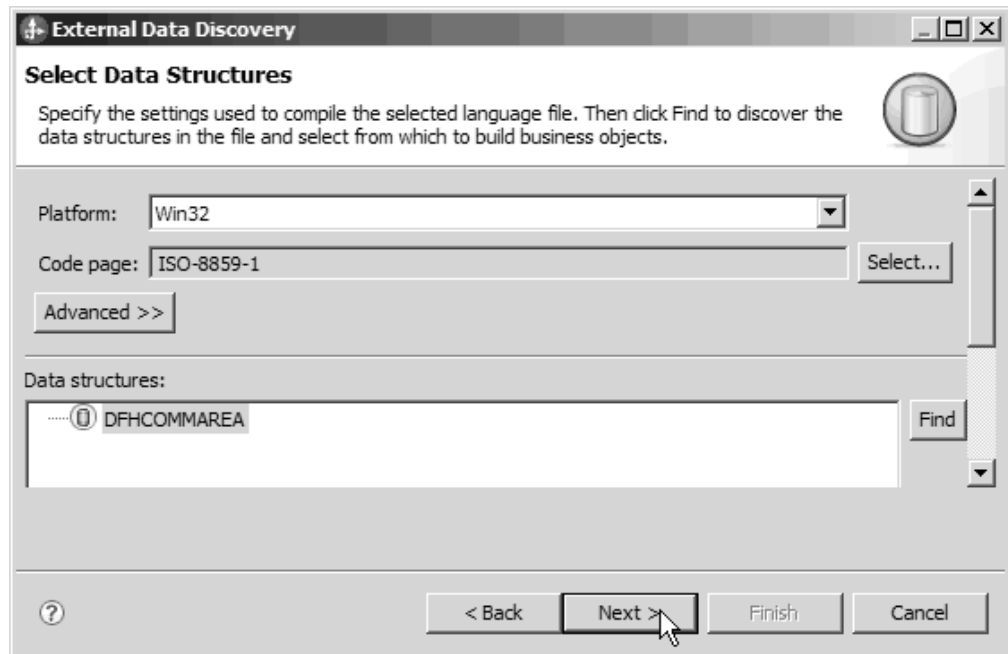


図 25. 「データ構造の選択 (Select Data Structures)」ウィンドウ

- g. 「DFHCOMMAREA」を選択し、「次へ」をクリックします。

h. 「終了」をクリックします。

DFHCOMMAREA という名前のビジネス・オブジェクトがモジュールの中に作成されます。

2. オプション: Wrapper ビジネス・オブジェクト定義を生成します。 Wrapper ビジネス・オブジェクト定義は、追加の機能で既存のビジネス・オブジェクト定義をラップします。 Wrapper ビジネス・オブジェクト定義の生成は、オプションです。 Wrapper ビジネス・オブジェクト定義を生成するオプションは、アダプター RAR ファイルがワークスペースにインポートされている場合に限り表示されません。

注: wrapper ビジネス・オブジェクト定義を生成したい場合は、外部サービス・ウィザードの実行前に生成する必要があります。

- a. ウィンドウの「ビジネス・インテグレーション」セクションで、モジュールを右クリックし、「新規」 → 「外部データからのビジネス・オブジェクト」を選択します。
- b. 「外部データからのビジネス・オブジェクト」ウィンドウで、「アダプター」を展開して、wrapper ビジネス・オブジェクトの生成対象となるアダプター・コネクター・プロジェクトを選択します。

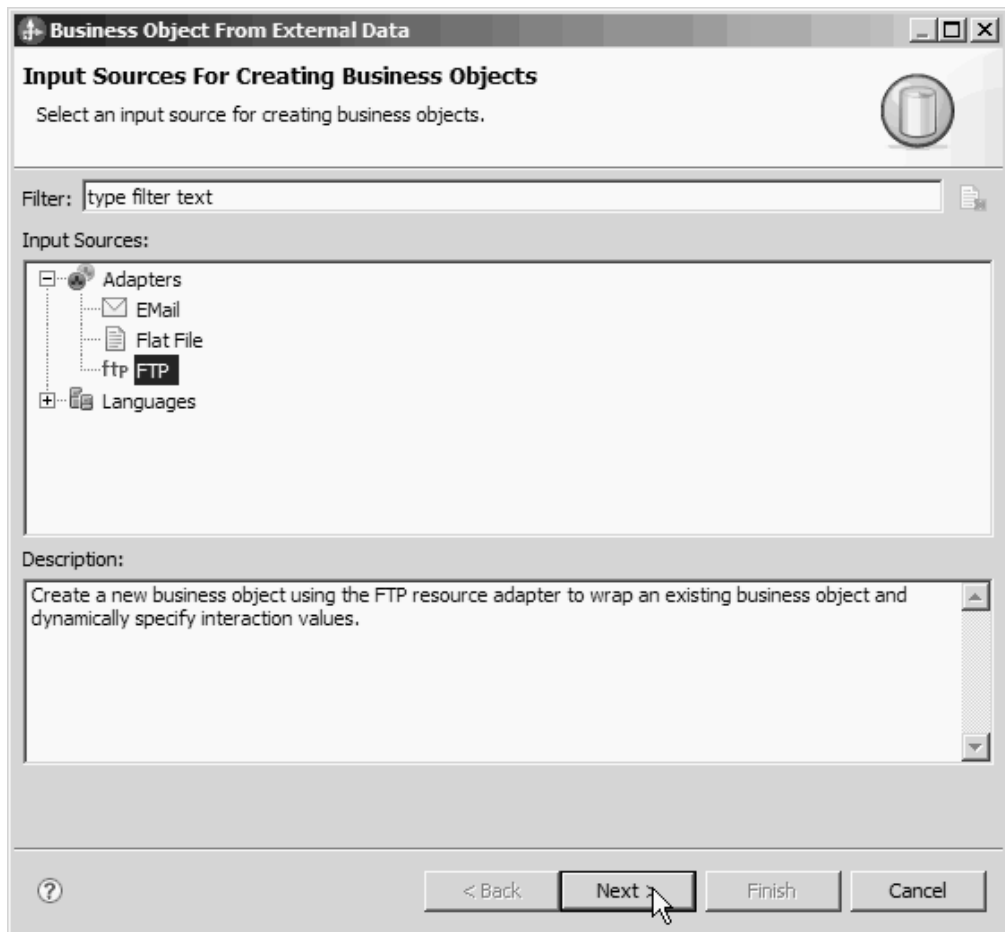


図 26. 「外部データからのビジネス・オブジェクト」ウィンドウ

c. 「次へ」をクリックします。

- d. 「ビジネス・オブジェクト・プロパティ」ウィンドウで、「参照」をクリックし、データ・タイプに DFHCOMMAREA を選択します。
- e. ビジネス・グラフを生成するには、「ビジネス・オブジェクトごとにビジネス・グラフを生成」チェック・ボックスを選択します。

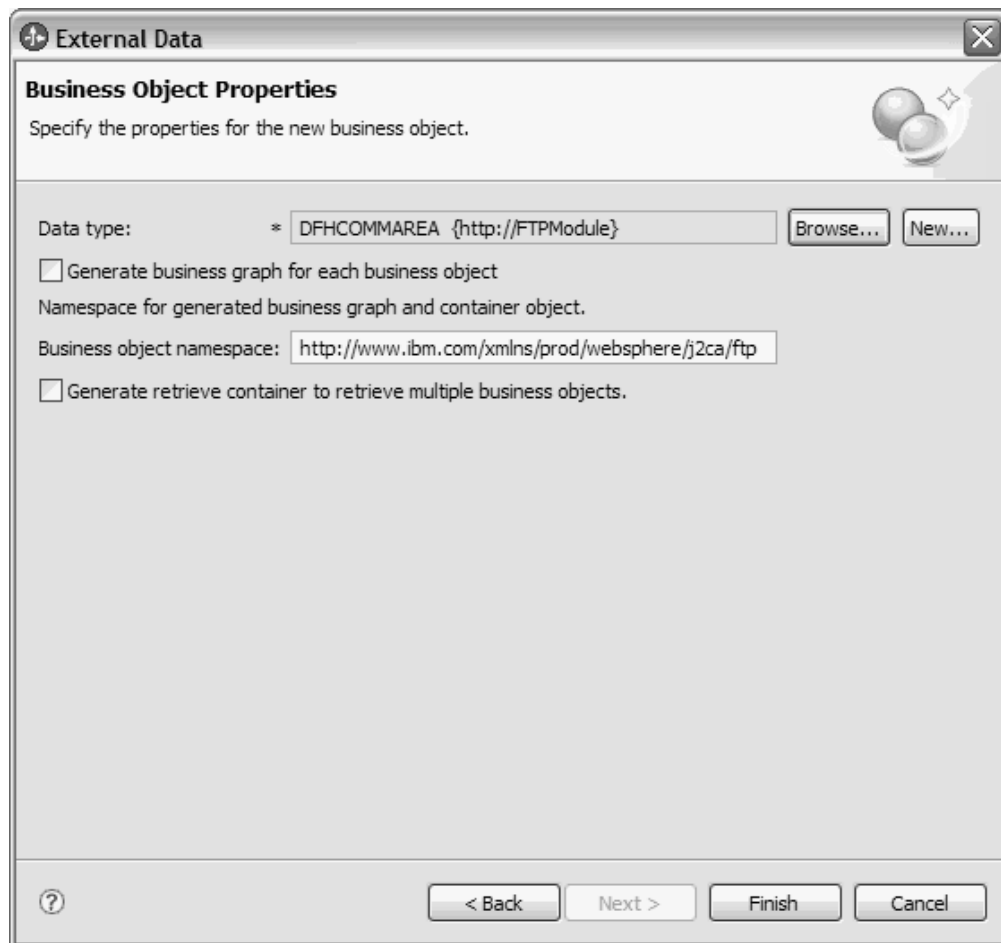


図 27. 「ビジネス・オブジェクト・プロパティ」ウィンドウ

注: 「複数のビジネス・オブジェクトを取得するための取得コンテナを生成」は、Outbound Retrieve 操作にのみ適用できます。

- f. 「終了」をクリックします。

現行モジュールの、DFHCOMMAREAWrapper という Wrapper ビジネス・オブジェクトと、DFHCOMMAREAWrapperBG というビジネス・グラフが、「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。

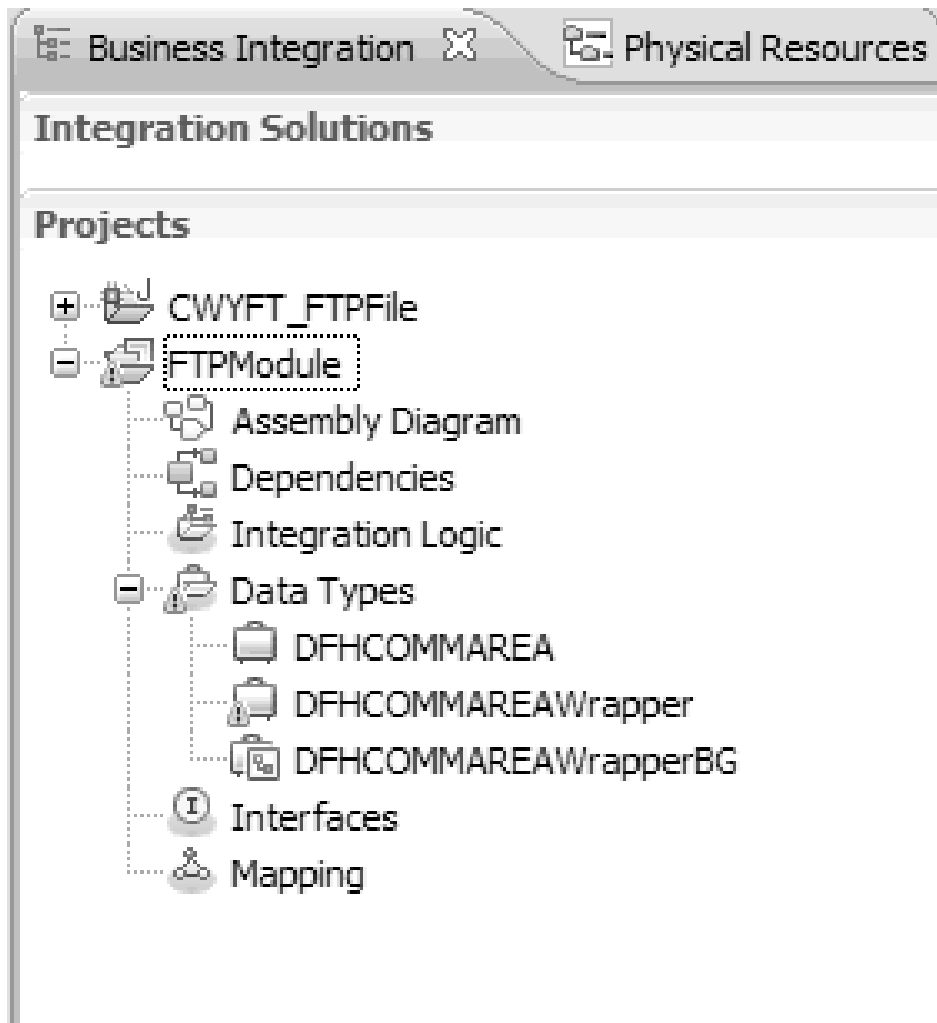


図 28. 「ビジネス・インテグレーション」ウィンドウにリストされた Wrapper ビジネス・オブジェクトとビジネス・グラフ

3. COBOL コピーブック Inbound モジュールの必要な成果物を生成します。
 - a. ウィンドウの「ビジネス・インテグレーション」セクションで、モジュールを右クリックし、「新規」 → 「外部サービス」を選択します。
 - b. 「アダプター」を選択して、「次へ」をクリックします。
 - c. 「アダプターの選択」ウィンドウで、FTP アダプター・コネクター・プロジェクトを選択して、「次へ」をクリックします。

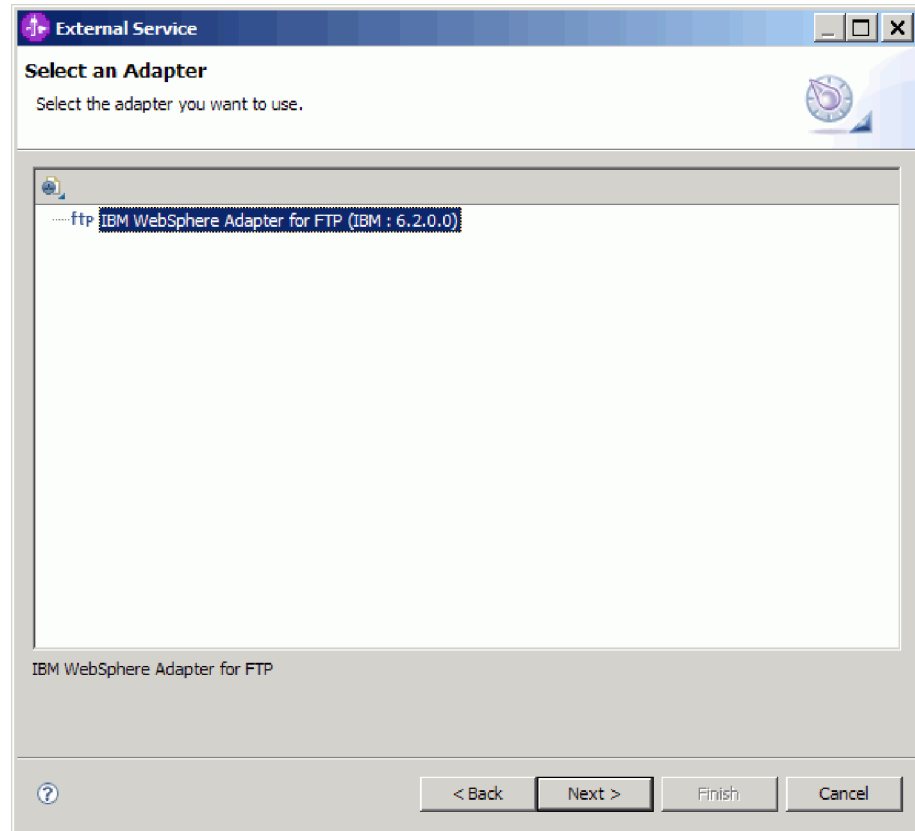


図 29. 「アダプターの選択」 ウィンドウ

- d. 「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。
- e. 「参照」をクリックして、イベント・ディレクトリーを選択します。
- f. 「関数セレクター」には、デフォルト値を選択します。
- g. 「データ・フォーマット・オプション」リストで、「**COBOL、C、または PL/I のデータ・バインディングを使用 (Use COBOL, C or PL/I data binding)**」オプションを選択します。

注: これはデータ・バインディングではなく、データ・バインディング生成プログラムです。ツールは、現行モジュールの中に適切なデータ・バインディング・コードを生成します。

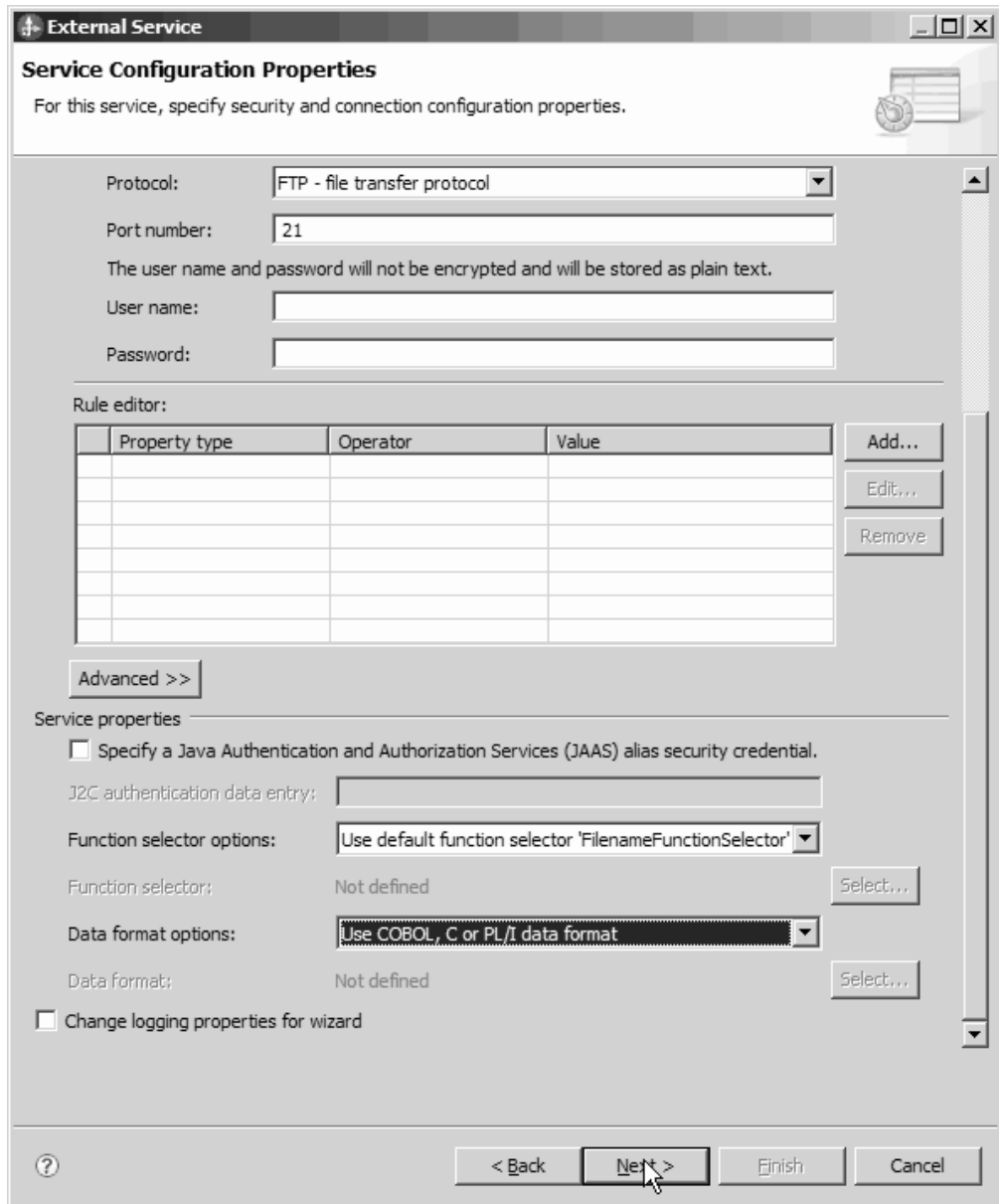


図 30. 「サービス構成プロパティ」ウィンドウ

- h. オプション: 入力ファイルに複数の COBOL プログラム・ソース・ファイルが含まれる場合、サイズまたは区切り文字によるファイル分割を使用可能にできます。ファイル分割を使用可能にするには、「**拡張**」をクリックし、「**拡張プロパティ**」をクリックします。サイズによるファイル分割を使用可能にするには、各 COBOL プログラム・ソース・ファイルの正しい長さを指定する必要があります。テキスト・エディターでビジネス・オブジェクトを開いて、最大長を加算するか、ファイルの先頭にある DFHCOMMAREA のコンテンツ・サイズを見つけます。232 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)』を参照してください。
- i. 「次へ」をクリックします。
- j. 「操作」ウィンドウで、「追加」をクリックします。

- k. 「操作」 ウィンドウで、データ・タイプに「ユーザー定義タイプ」を選択します。「次へ」をクリックします。
- l. 入力タイプの場合、「参照」をクリックして、生成されたビジネス・オブジェクト (DFHCOMMAREA) を選択します。「OK」をクリックします。

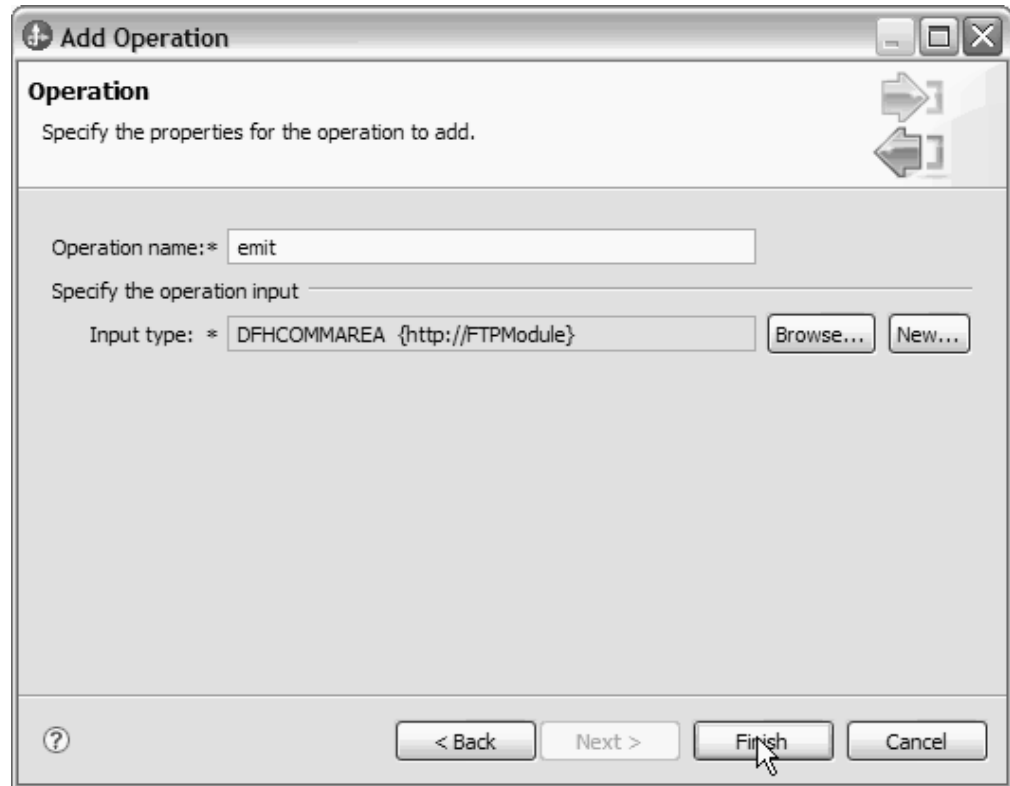


図 31. 「データ・タイプの選択 (Data Type Selection)」 ウィンドウでの入力タイプの選択

- m. 「次へ」をクリックし、「終了」をクリックします。

COBOL コピーブック、WSDL ファイル、エクスポート・ファイル、およびその他の成果物によって使用されるデータ・バインディングが生成されます。生成されたデータ・バインディング・クラスは、「Project Explorer」を表示してください。

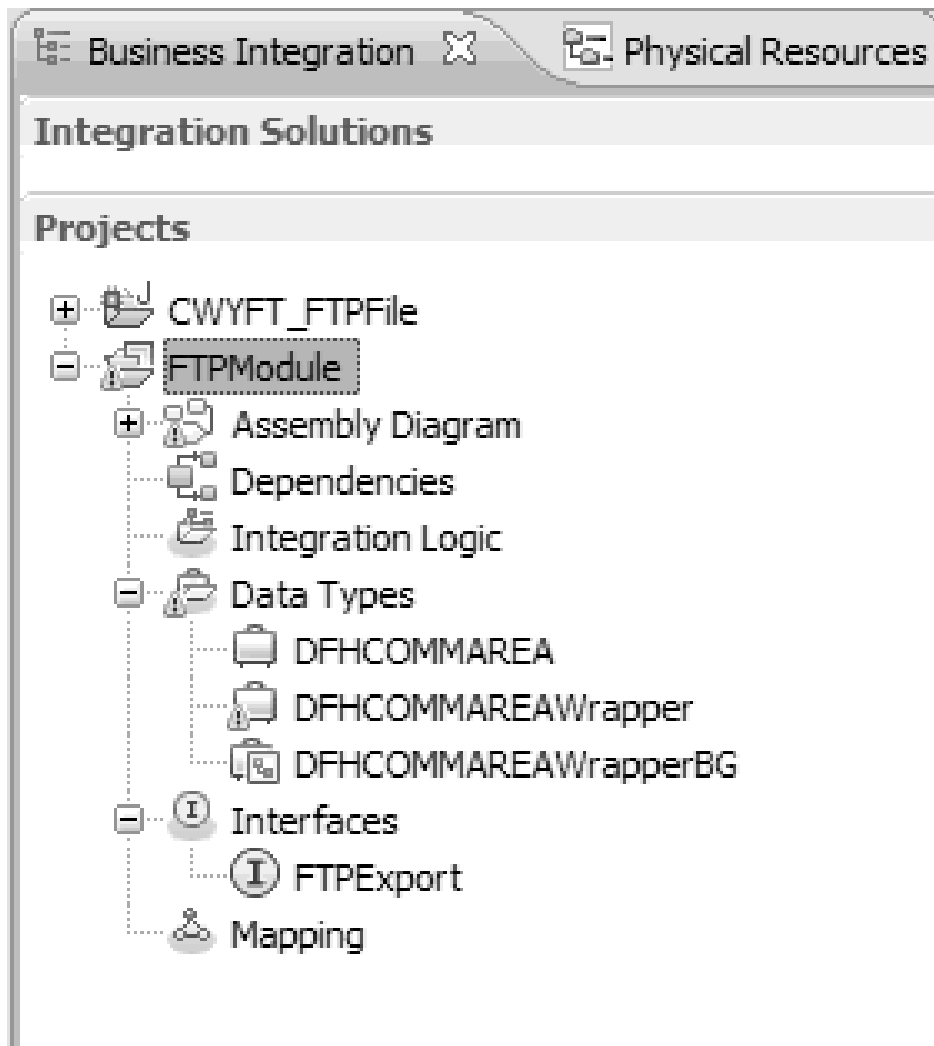


図 32. COBOL コピーブック、WSDL ファイル、エクスポート・ファイル、およびその他の成果物によって使用されるデータ・バインディング

結果

Inbound モジュールの COBOL プログラム・ソース・ファイル用の、ビジネス・オブジェクト、Wrapper ビジネス・オブジェクト、およびビジネス・グラフが作成されます。COBOL コピーブック・データ・バインディングを使用する Inbound 操作用の成果物が生成されます。このモジュールは、WebSphere Process Server にデプロイして、Inbound 操作に対してテストできます。

次のタスク

モジュールをデプロイします。

WebSphere Application Server 環境変数の定義

実行時環境の管理コンソールを使用して、WebSphere Application Server 環境変数を定義します。

始める前に

このタスクを実行する理由および時期

WebSphere Application Server 環境変数を定義するには、以下の手順を実行します。

このタスクの手順

1. サーバーの管理コンソールを開始します。
2. 左にあるメニューで、「環境」 → 「WebSphere 変数」を選択します。
3. 環境変数のスコープを選択します。スコープは、管理コンソール・パネルでリソース定義が表示されるレベルを指定します。使用可能な値は、サーバー、ノード、およびセルです。この例では、「Cell=widCell」を選択しています。

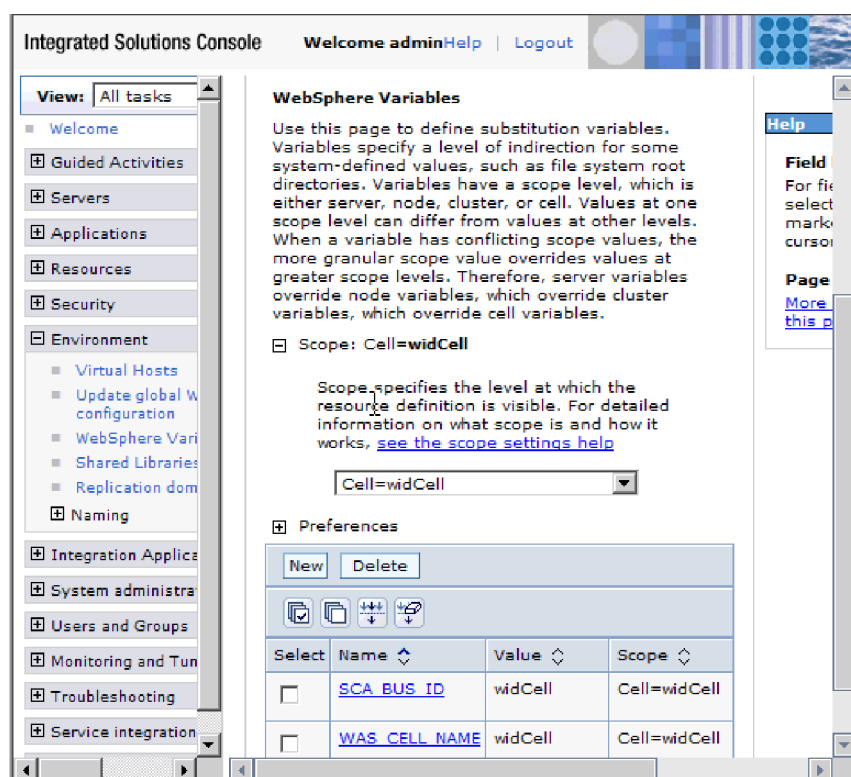


図 33. 環境変数のスコープの設定

4. 「新規」をクリックし、環境変数の名前と値を指定します。名前は、物理パスを表すシンボル名です。値は、変数が表す絶対パスです。この例では、名前は EVENT_DIRECTORY で、値は /home/user/event です。オプションの「説明」フィールドを使用して、変数の目的を記述できます。



図 34. 環境変数の名前と値の指定

5. 「OK」をクリックし、変更を保存します。

結果

EVENT_DIRECTORY という名前と、/home/user/event の値を持ち、スコープが Cell=widCell の環境変数が作成されます。これで、イベント・ディレクトリーを指定する必要があるときはいつでも、外部サービス・ウィザードの中でそれを使用できます。

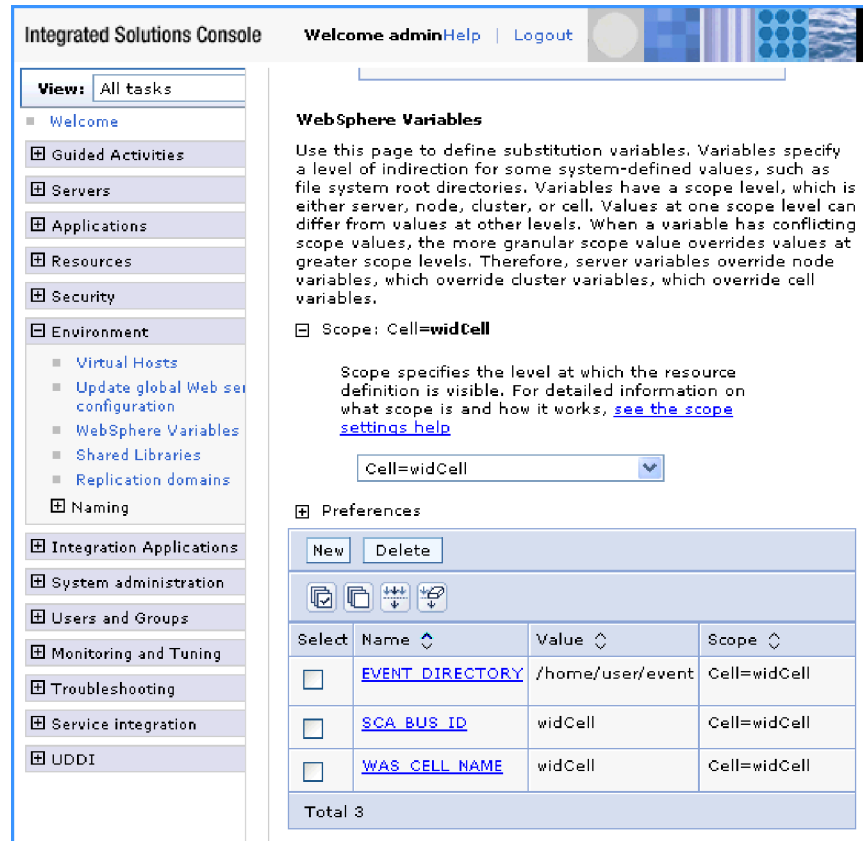


図 35. 「WebSphere 変数」ウィンドウに表示された新しい環境変数 `EVENT_DIRECTORY`

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・ウィザードを使用した単純サービスの作成

アダプター・パターンは、アダプターを使用して単純サービスを作成するための迅速で簡単な方法を提供します。

始める前に

RetrieveAFileModule というモジュールと、Customer というビジネス・オブジェクトが既に作成されています。ローカル・ファイルとディレクトリーの指定に

WebSphere Application Server 環境変数を使用していれば、それらの環境変数は WebSphere Process Server 管理コンソールを使用して定義されています。

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6.

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ー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。
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い場合は、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 Inbound 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。

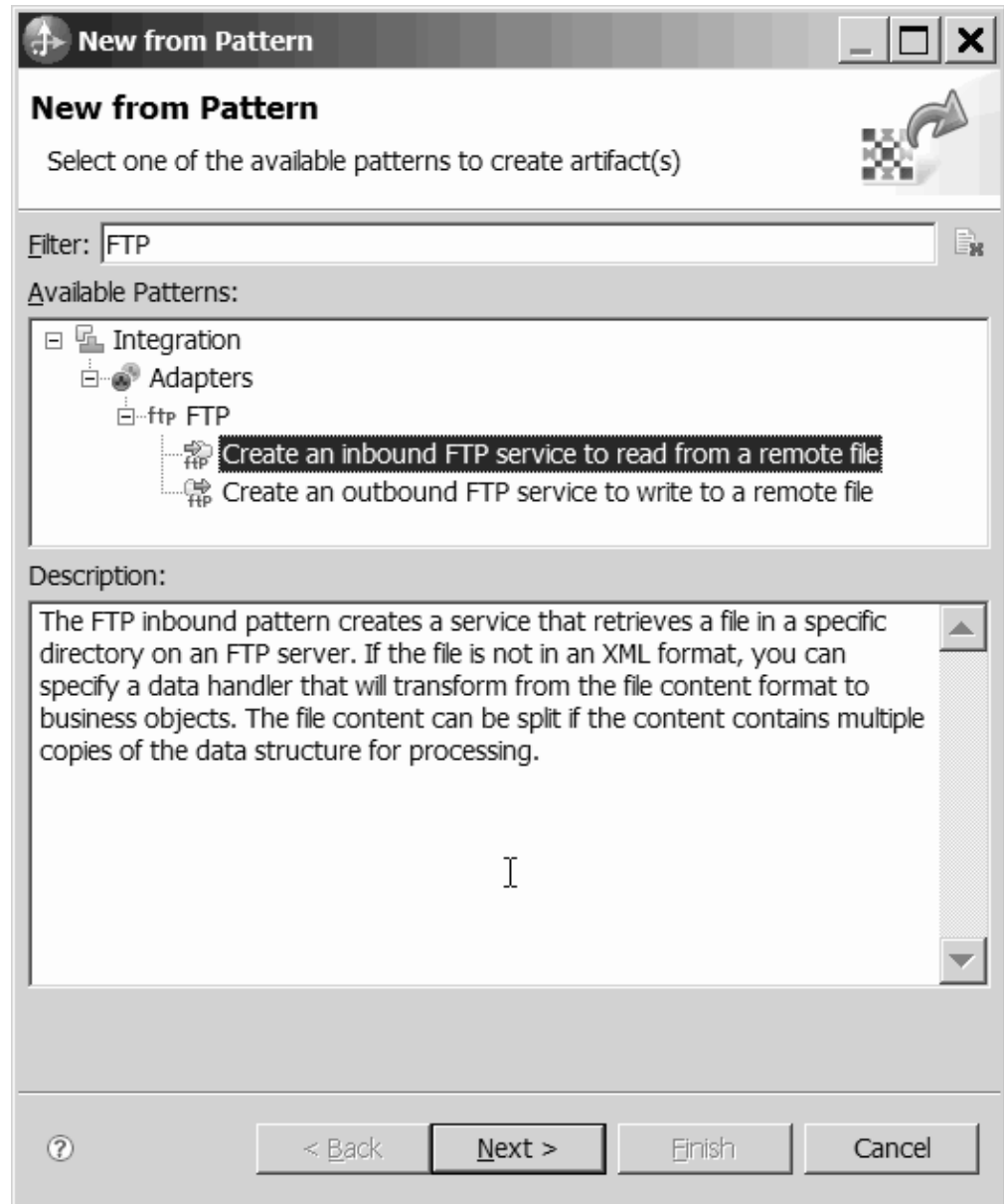


図 36. 「パターンから新規作成」 ウィンドウ

3. 「新規 Inbound 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。

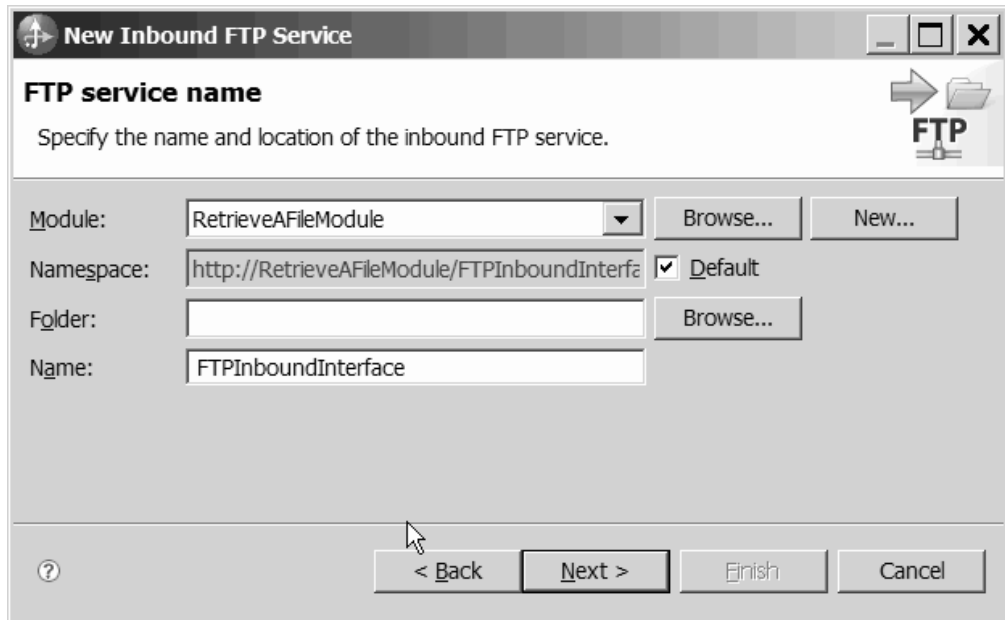


図 37. 「FTP サービス名」ウィンドウ

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. 入力ファイルを置くディレクトリー、この場合は「/home/user/event」ディレクトリーを指定して、「次へ」をクリックします。この値に、WebSphere Application Server 環境変数を使用するには、\$ 記号の後に、変数の名前を中括弧で囲んで指定します。例えば、\${FTPINBOUNDEVENTS} です。

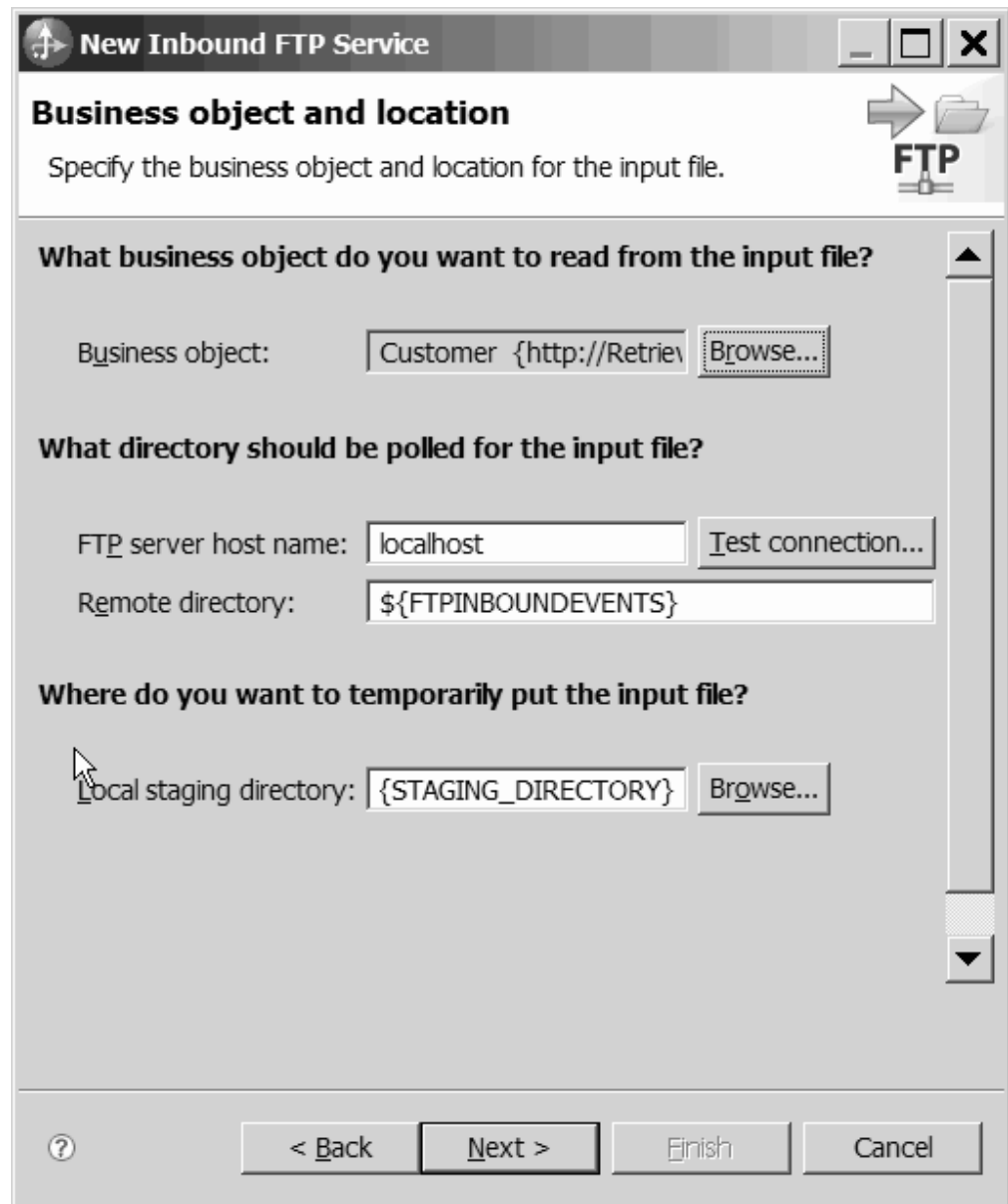


図 38. 「ビジネス・オブジェクトおよび場所 and location)」ウィンドウ

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。

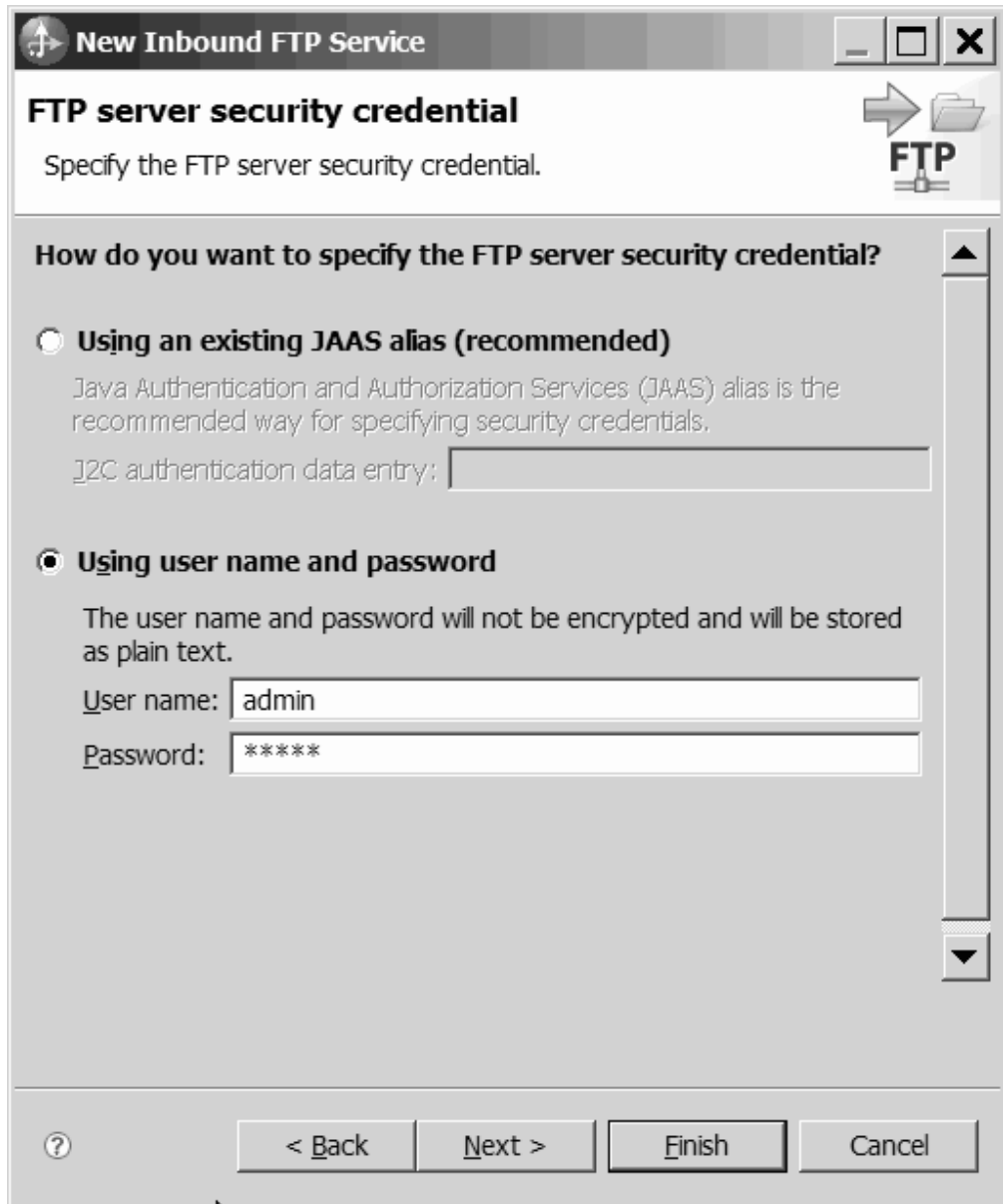


図 39. 「FTP サーバーのセキュリティー・クレデンシャル」 ウィンドウ

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。

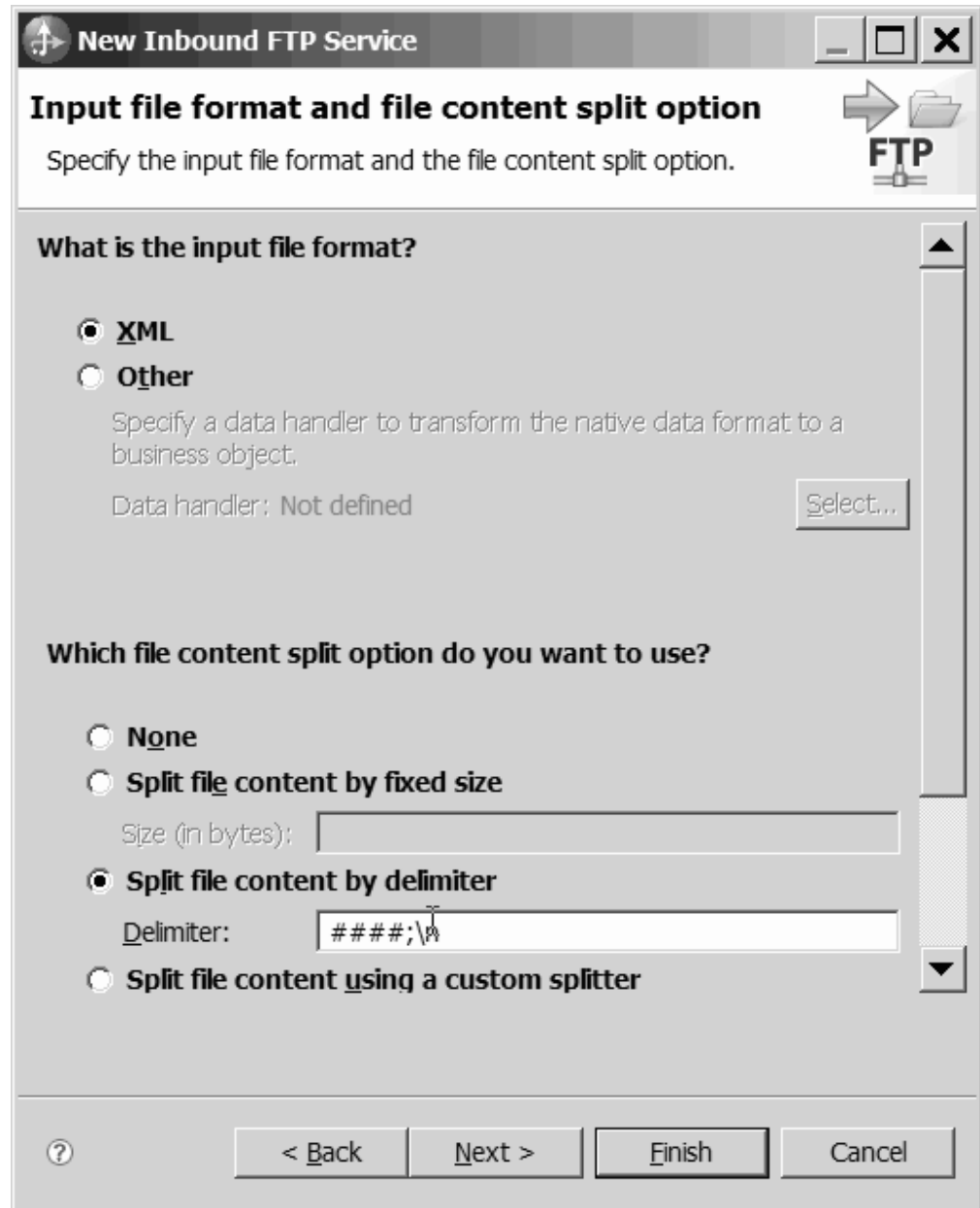


図 40. 「入力ファイル・フォーマットとファイル内容の分割オプション (Input file format and file content split option)」ウィンドウ

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。この値に、WebSphere Application Server 環境変数を使用するには、\$ 記号の後に、変数の名前を中括弧で囲んで指定します。例えば、\${FTPINBOUNDARCHIVE} 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。

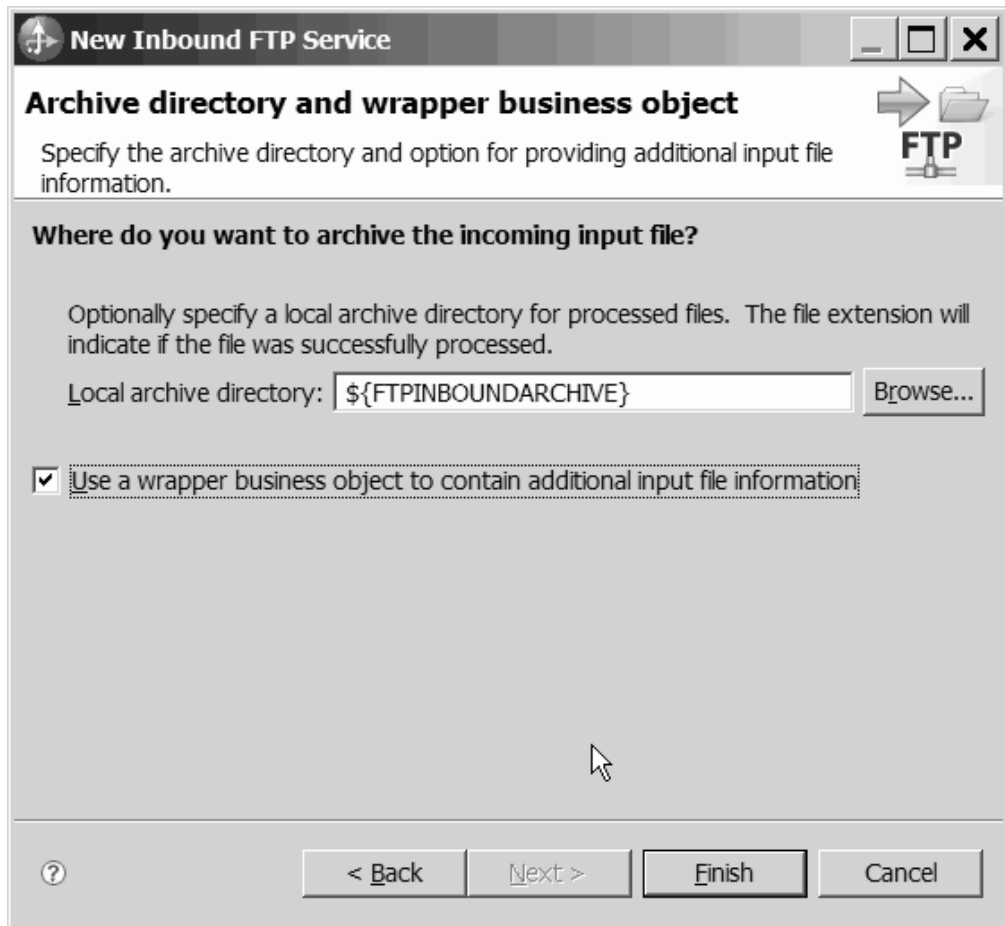


図 41. 「アーカイブ・ディレクトリーおよびラッパー・ビジネス・オブジェクト」ウィンドウ

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 7.

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。

表 7. (続き)

成果物	名前	説明
インターフェース	FTPIInboundInterface	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。

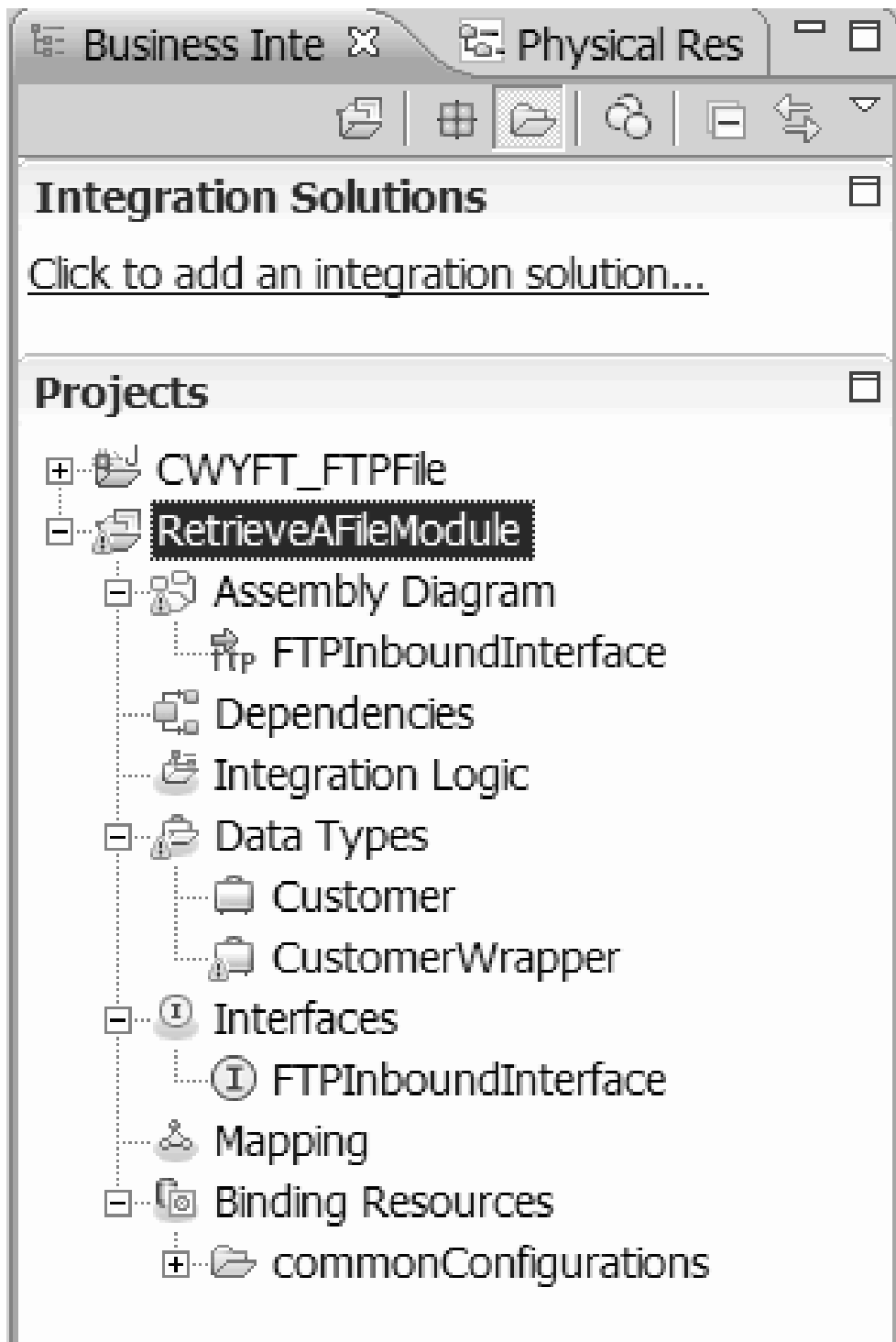


図42. 「WebSphere Integration Developer」 ウィンドウの新しい成果物が表示された「ビジネス・インテグレーション」セクション

外部サービス・ウィザードの開始

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

WebSphere Integration Developer で、外部サービス・ウィザードを開始し、アダプター用のプロジェクトを作成します。既存プロジェクトが存在する場合、ウィザードで新規に作成するのではなく、それを選択することができます。

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. 外部サービス・ウィザードを開始するには、WebSphere Integration Developer の Business Integration パースペクティブに進み、「ファイル」→「新規」→「外部サービス」の順にクリックします。
2. 「外部サービス」ウィンドウで、「アダプター」を展開して、作成するアダプターのタイプとして「FTP」を選択し、「次へ」をクリックします。
3. 「アダプターの選択」ウィンドウで、新規プロジェクトを作成する場合はアダプター名を選択し、既存のプロジェクトを再利用する場合はそのプロジェクトを選択します。
 - プロジェクトを作成するには、以下の手順を実行します。
 - a. 「IBM WebSphere Adapter for FTP (IBM : 6.2.0.0)」を選択し、「次へ」をクリックします。
 - b. 「アダプターのインポート」ウィンドウで、作成するプロジェクトに関する詳細を指定します。
 - 1) 「コネクター・プロジェクト」フィールドで、プロジェクトに別の名前を指定します (オプション)。
 - 2) 「ターゲット・ランタイム」フィールドで、サーバーを選択します (例えば、WebSphere Process Server v6.2)。
 - 3) 「次へ」をクリックします。
 - 既存のプロジェクトを選択するには、「IBM WebSphere Adapter for FTP (IBM : 6.2.0.0)」の下でプロジェクト・フォルダーを選択して、「次へ」をクリックします。

結果

新規プロジェクトの場合、プロジェクトが作成され、「ビジネス・インテグレーション」パースペクティブにリストされます。ウィザードは、指定されたプロジェクトにアダプター成果物を作成します。

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「**Outbound**」を選択し、「次へ」をクリックします。
2. 「コネクタ・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、中央の場所であ

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。詳しくは、Managed (J2C) Connection Factory プロパティの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 、セキュア FTP (FTPS)、または FTP over SSH (SFTP) のいずれかを指定。SFTP を選択すると、ポート番号はデフォルトの 22 に設定されます。
 - 「ユーザー名」 - 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を指定します。詳しくは、235 ページの『「ユーザー名」プロパティ (UserName)』を参照してください。
 - 「パスワード」 - 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を指定します。詳しくは、224 ページの『「パスワード」プロパティ (Password)』を参照してください。

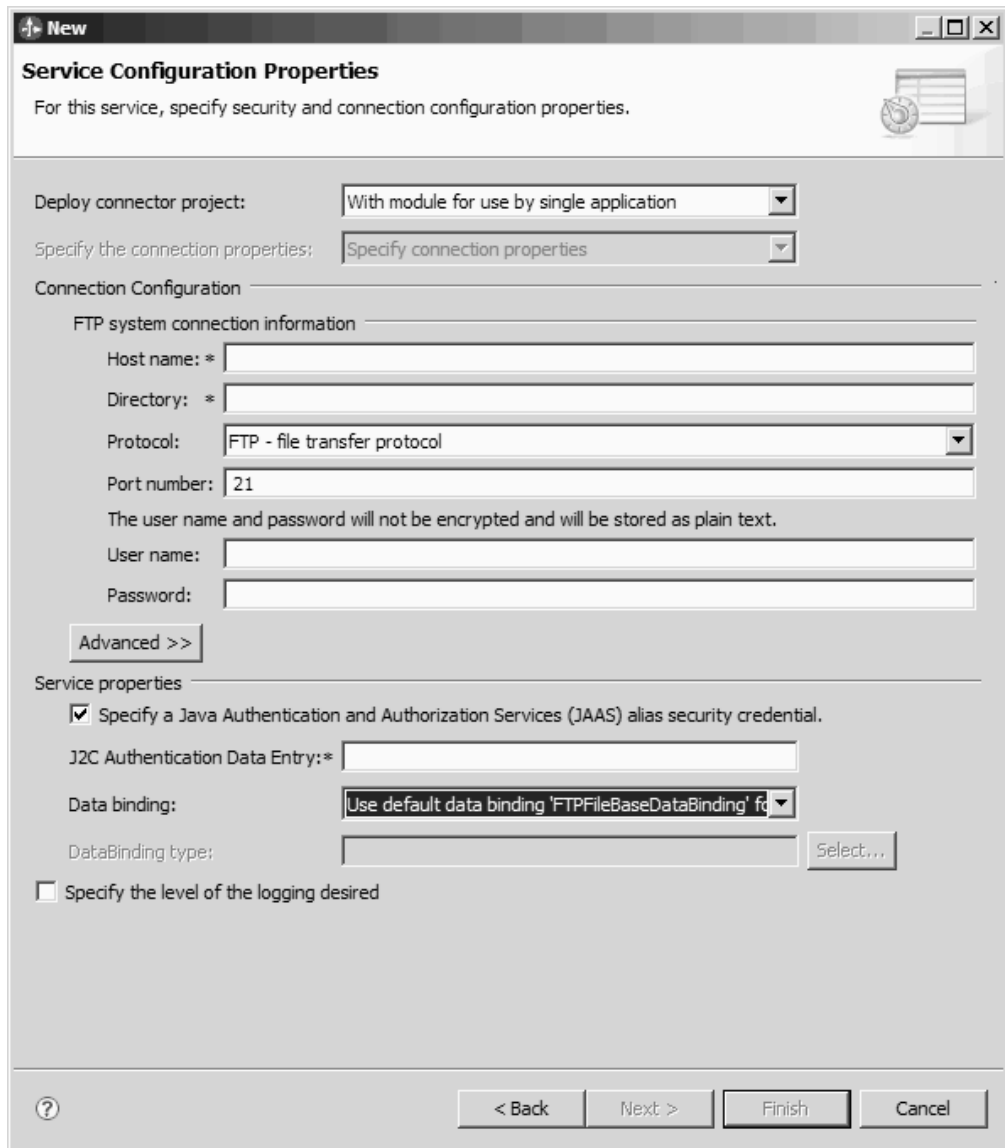


図 43. 「サービス構成プロパティ」 ウィンドウ

4. 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、ステージング・ディレクトリー、ロギングとトレース、セキュア接続、およびシーケンス・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。詳しくは、Managed (J2C) Connection Factory プロパティのトピックを参照してください。
 - a. アダプターのインスタンスが複数ある場合、「**ロギングおよびトレース**」を展開し、**アダプター ID** に、このインスタンスに固有の値を設定します。このプロパティについて詳しくは、173 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。
5. 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。

6. 「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。

7. オプション: ログ・ファイル出力の場所を指定したり、このモジュールのロギング・レベルを定義したりする場合は、「任意のロギング・レベルを指定 (Specify the level of logging desired)」チェック・ボックスを選択します。ロギング・レベルについての情報は、『トラブルシューティングおよびサポート』トピックの『ロギング・プロパティの構成』セクションを参照してください。

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、90 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 サーバーとの接続のために、アダプター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。

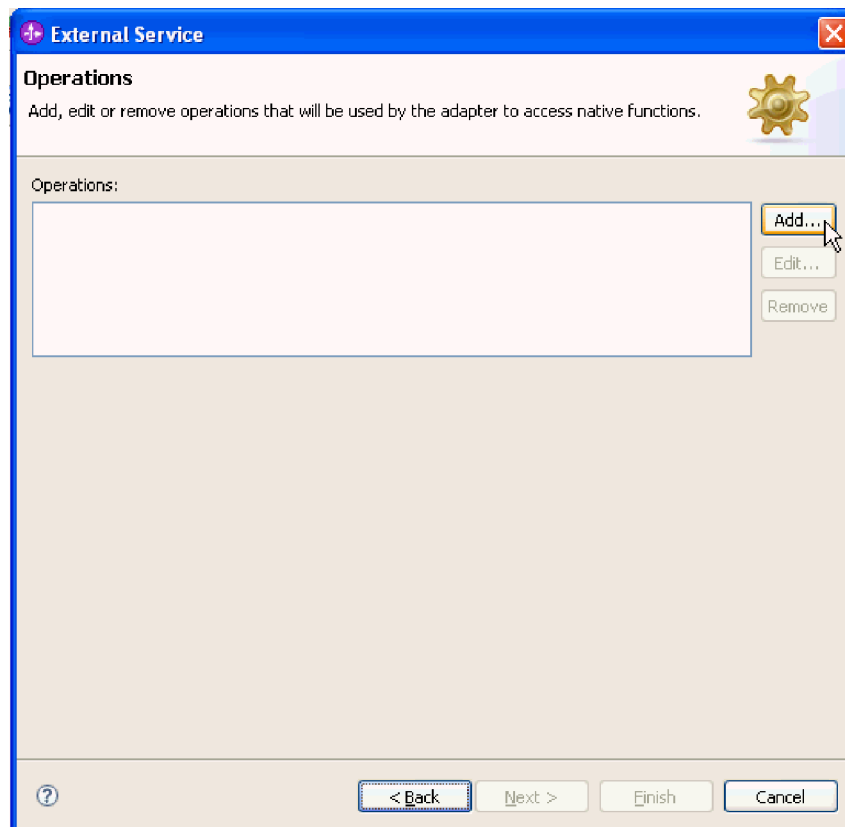


図 44. 操作の追加

2. 「操作の追加」ウィンドウで、「操作の種類」リストから操作を選択し、「操作入力のデータ型」リストからデータ型を選択して、「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」によって提供されるデータ・バインディングは、サポートされる操作の汎用入力タイプのみサポートします。

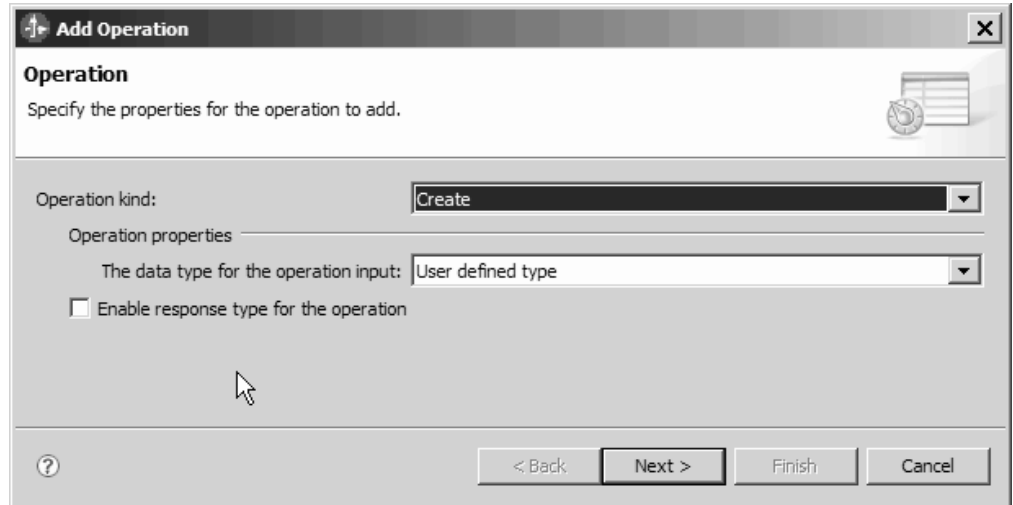


図 45. データ・タイプの選択

3. オプション: ファイル名が返されるか、あるいは Delete および ServerToServer 操作で True または False が返されるようにするには、「**操作の応答タイプを有効にする**」チェック・ボックスを選択します。Exists、List、および Retrieve 操作の場合、応答タイプは必須で、デフォルトで「**操作の応答タイプを有効にする**」チェック・ボックスが選択されています。「次へ」をクリックします。
4. 「操作」ウィンドウで、「**操作名**」フィールドに操作の名前を入力します。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、5 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。

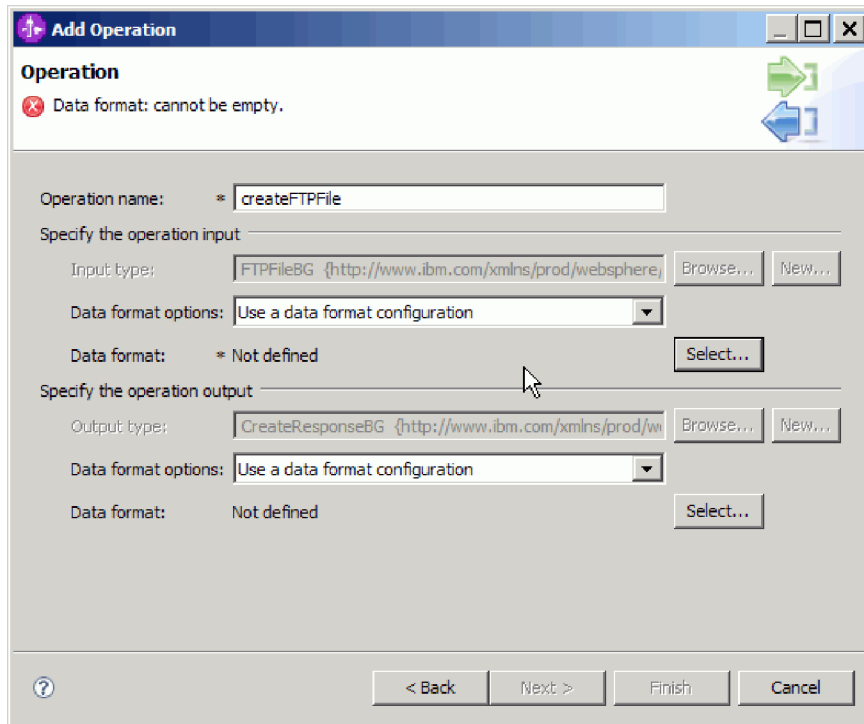


図 46. 操作への名前の指定

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」の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、97 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構成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. 「操作入力の指定」領域で、「データ・フォーマット・オプション」ドロップダウン・リストから「データ・フォーマット構成の使用」を選択します。
2. 「選択」をクリックします。

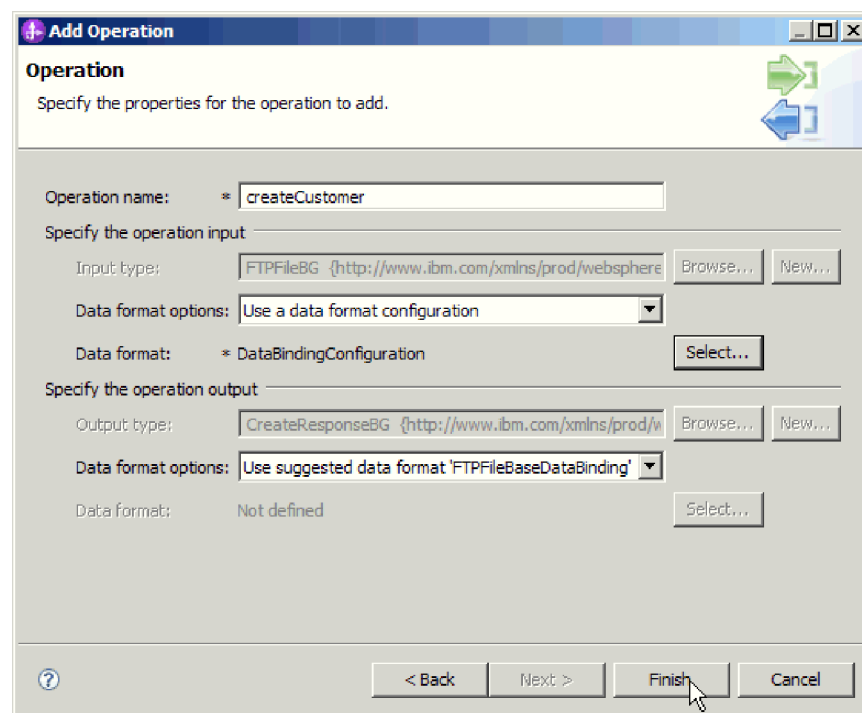


図 47. 「操作」ウィンドウ

3. 「操作出力の指定」領域で、「データ・フォーマット・オプション」ドロップダウン・リストから FTPFileBaseDataBinding を選択します。
4. 「次へ」をクリックします。
5. 「データ変換構成」プロパティの構成名を指定して、「終了」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」 → 「リソース構成のバインディング (Binding 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. 「新規データ変換構成」ウィンドウで、「新規」を選択します。
2. データ・ハンドラー構成の名前 (この例では DataBindingConfiguration を使用) を「名前」フィールドに入力します。この操作は、初めてデータ・ハンドラーを設定するときに実行します。あとで同じデータ・ハンドラーを使用するには、「参照」をクリックし、そのハンドラーを選択します。

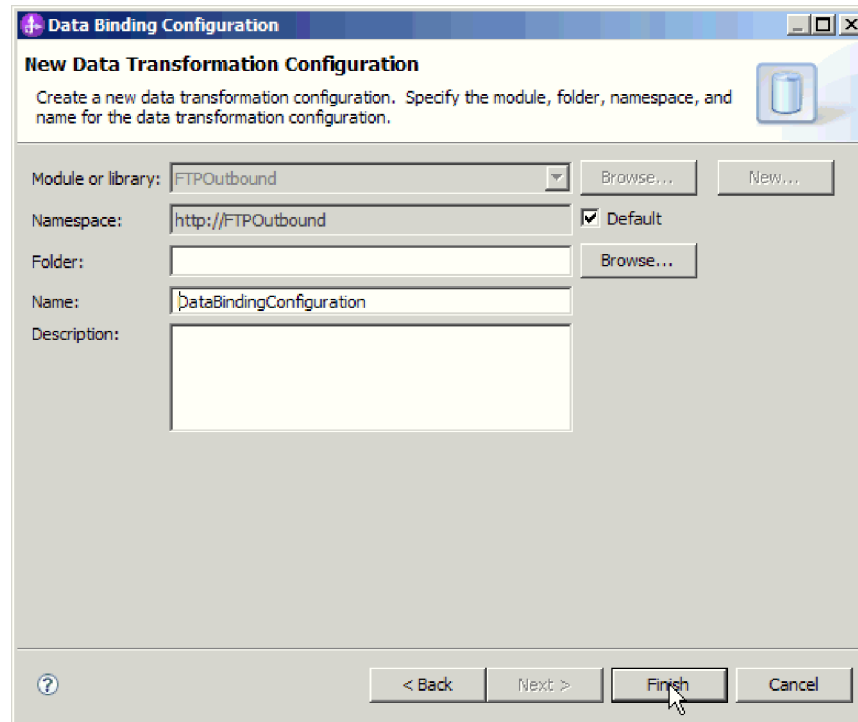


図 48. データ・ハンドラー構成の名前の指定

3. 「次へ」をクリックします。
4. 「データ変換プロパティ」ウィンドウで、「バインディング・タイプ」プロパティの横にあるドロップダウン・リストをクリックします。2つの選択項目「DataBinding」および「DataHandler」が表示されます。旧バージョンのアダプター用に作成されたデータ・バインディングを使用する場合は、「DataBinding」を選択します。新しいデータ・ハンドラーを構成する場合は、「DataHandler」を選択します。「選択」をクリックすると、新規データ・ハンドラー構成が作成されます。

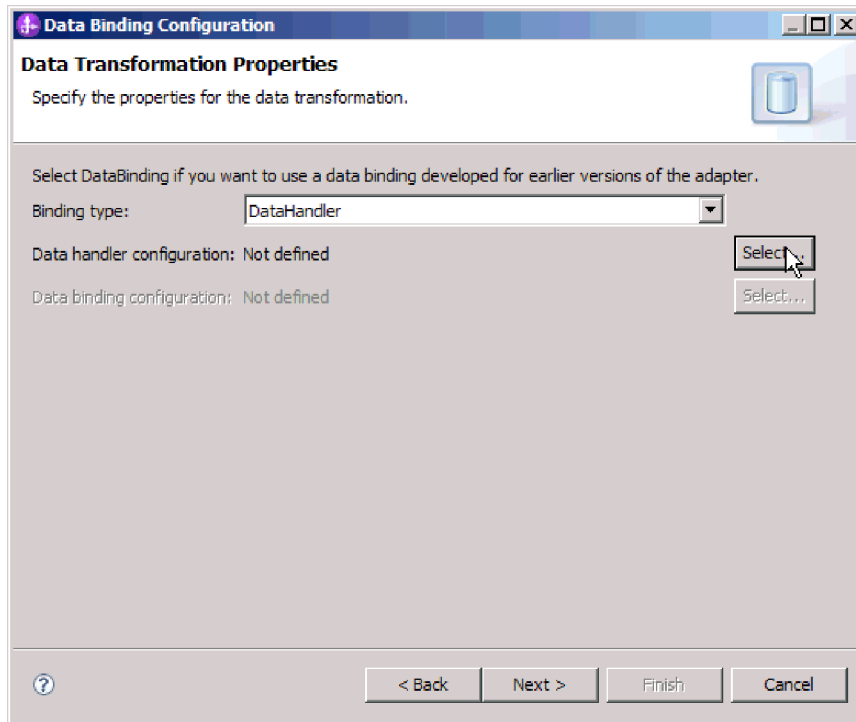


図 49. 「データ変換」ウィンドウでのバインディング・タイプの指定

5. 「次へ」をクリックします。
6. 「新規データ変換構成」ウィンドウで「新規」をクリックし、データ・ハンドラー構成のモジュール、名前空間、フォルダー、および名前を指定します。「次へ」をクリックします。

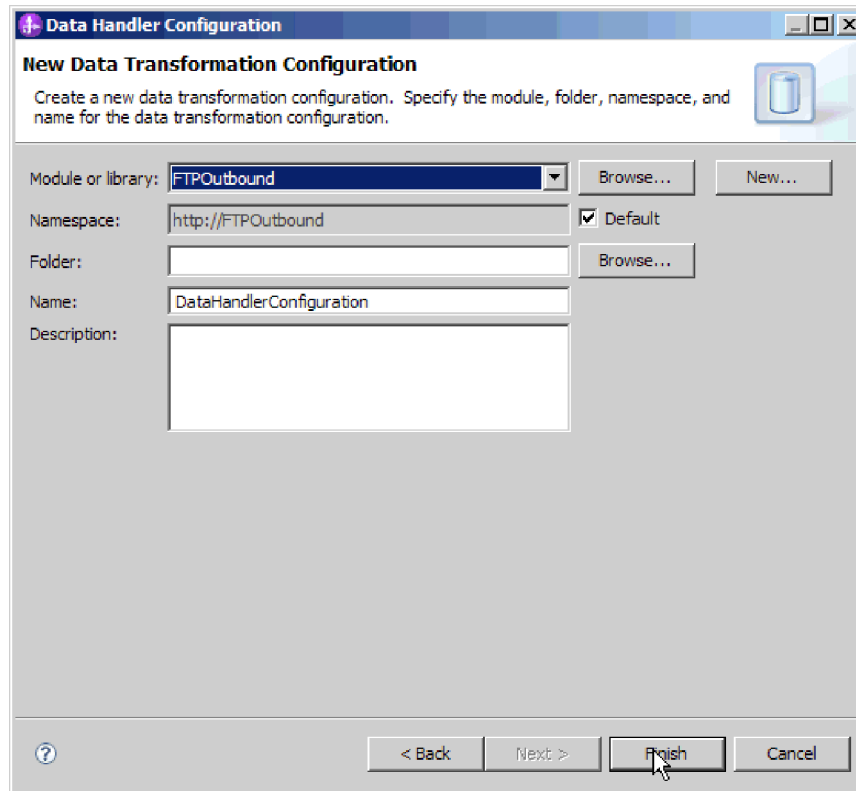


図 50. データ・ハンドラー構成の詳細の指定

7. データ・ハンドラーのクラス名を選択します。「データ・フォーマット変換の選択」ウィンドウで、リスト・オプションから「既存のデータ・フォーマット変換の使用」をクリックします。選択可能なデータ・ハンドラー・クラスのリストが表示されます。データ・ハンドラー・クラスを選択します (この例では XML データ・ハンドラーを使用)。「次へ」をクリックします。

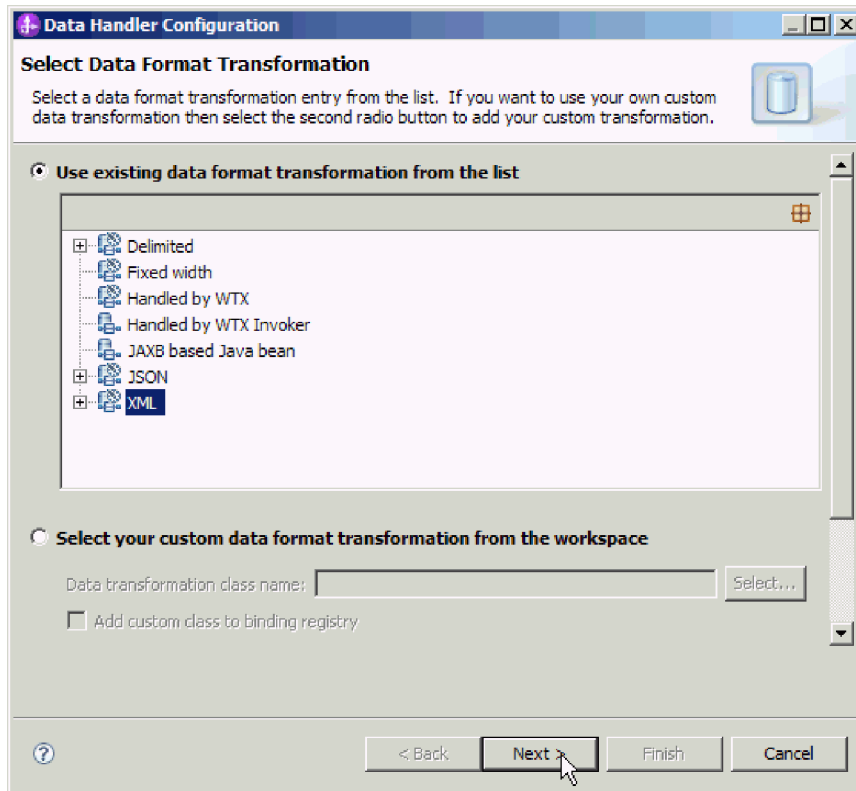


図 51. 新規データ・ハンドラー構成の作成

8. 「データ変換プロパティ」ウィンドウで、エンコード方式を指定します。デフォルトは UTF-8 です。

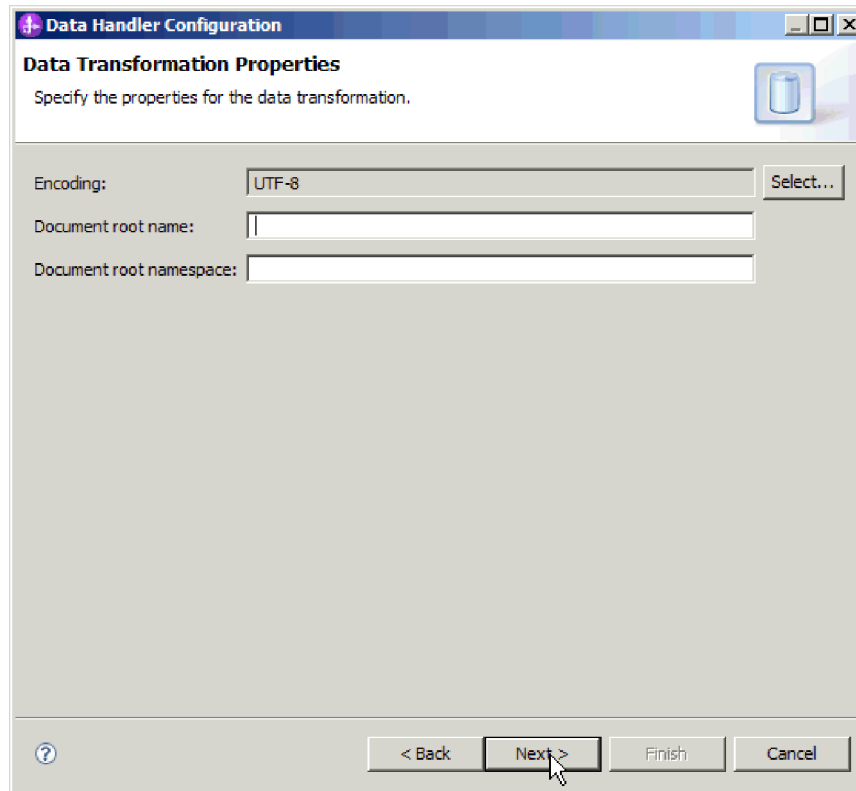


図 52. データ・ハンドラー構成のエンコード方式の指定

9. 「次へ」をクリックします。
10. Output 操作でのデータ・バインディング構成を選択します。「操作」ウィンドウの、「Output 操作のデータ・バインディング構成 (output Data binding configuration)」フィールドで、「参照」をクリックします。アダプターでは 1 つのデータ・バインディングのみを使用でき、そのデータ・バインディングは Input 操作の DataBinding タイプの設定時に構成済みであるため、Output 操作の DataBinding タイプでも同じデータ・バインディング・タイプ (DBConfig) を選択します。
11. 「終了」をクリックします。次の画面に、追加された Create 操作が対話仕様プロパティとともに表示されます。
12. 「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。

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。

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。

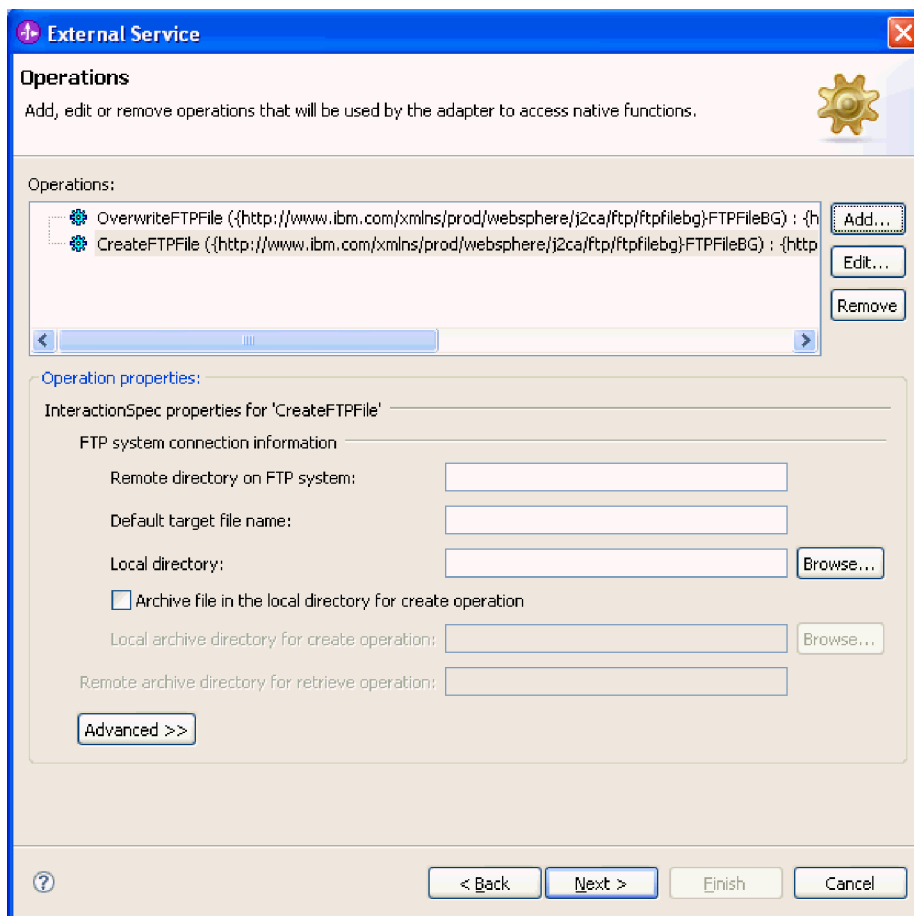


図 53. 対話仕様プロパティ

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。

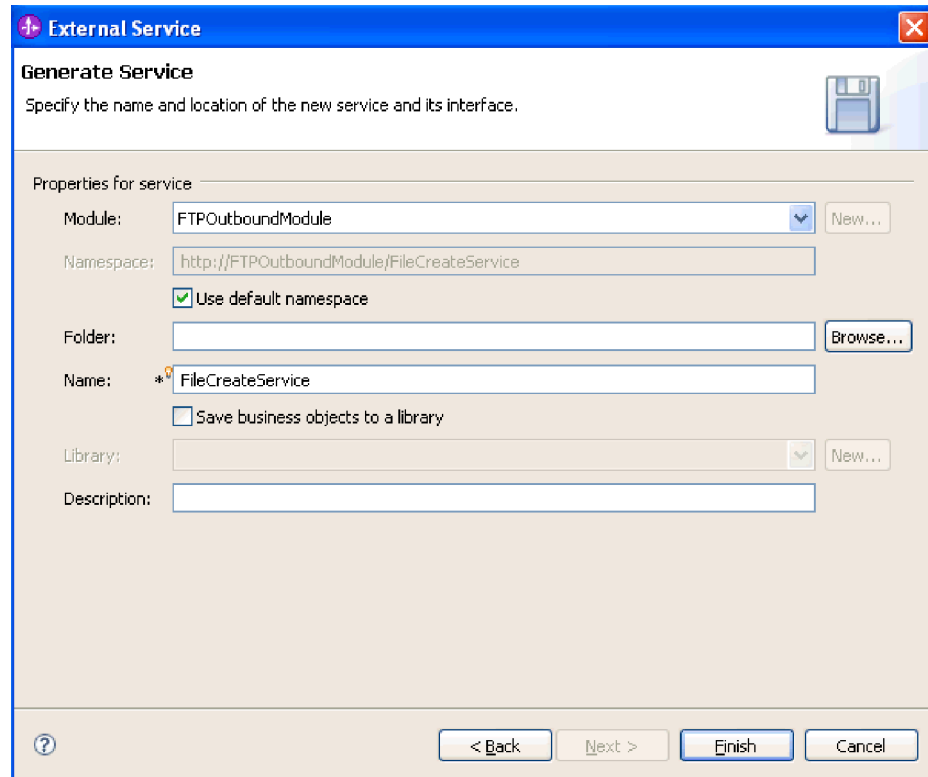


図 54. インターフェースへの命名

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。

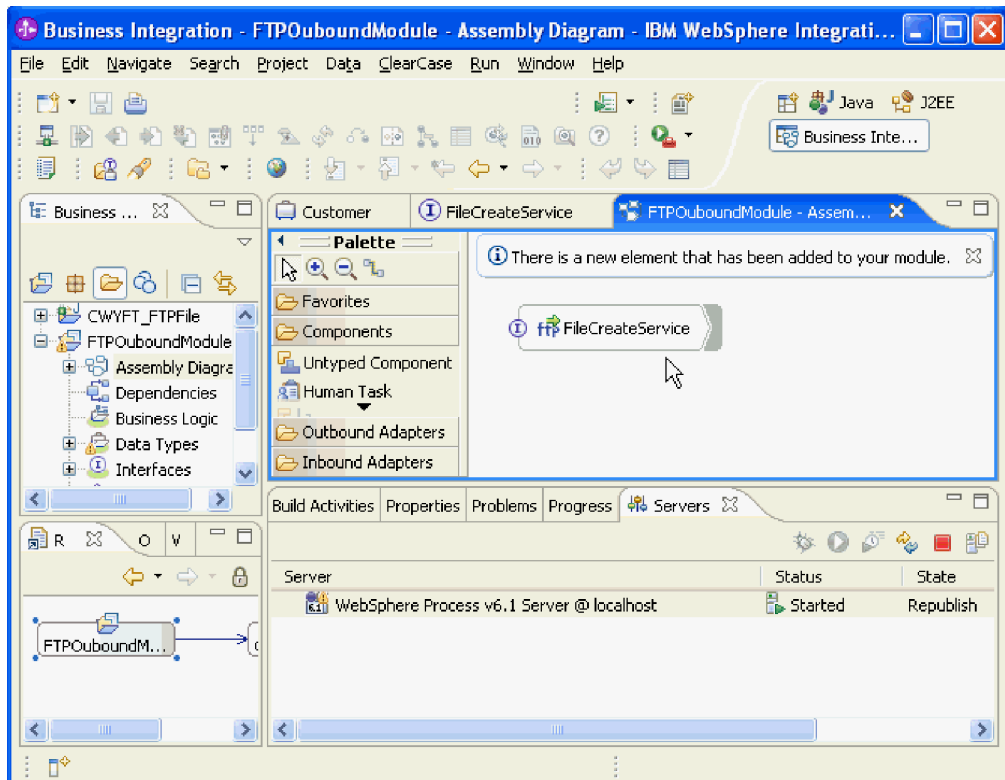


図 55. WebSphere Integration Developer のインターフェース

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 内の外部サービス・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 **Managed Connection Factory** 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、セキュア FTP (FTPS)、または FTP over SSH (SFTP) のいずれかを指定。SFTP を選択すると、ポート番号はデフォルトの 22 に設定されます。
- 「**ユーザー名**」 - 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を指定します。詳しくは、235 ページの『「ユーザー名」プロパティ (UserName)』を参照してください。
- 「**パスワード**」 - 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を指定します。詳しくは、224 ページの『「パスワード」プロパティ (Password)』を参照してください。

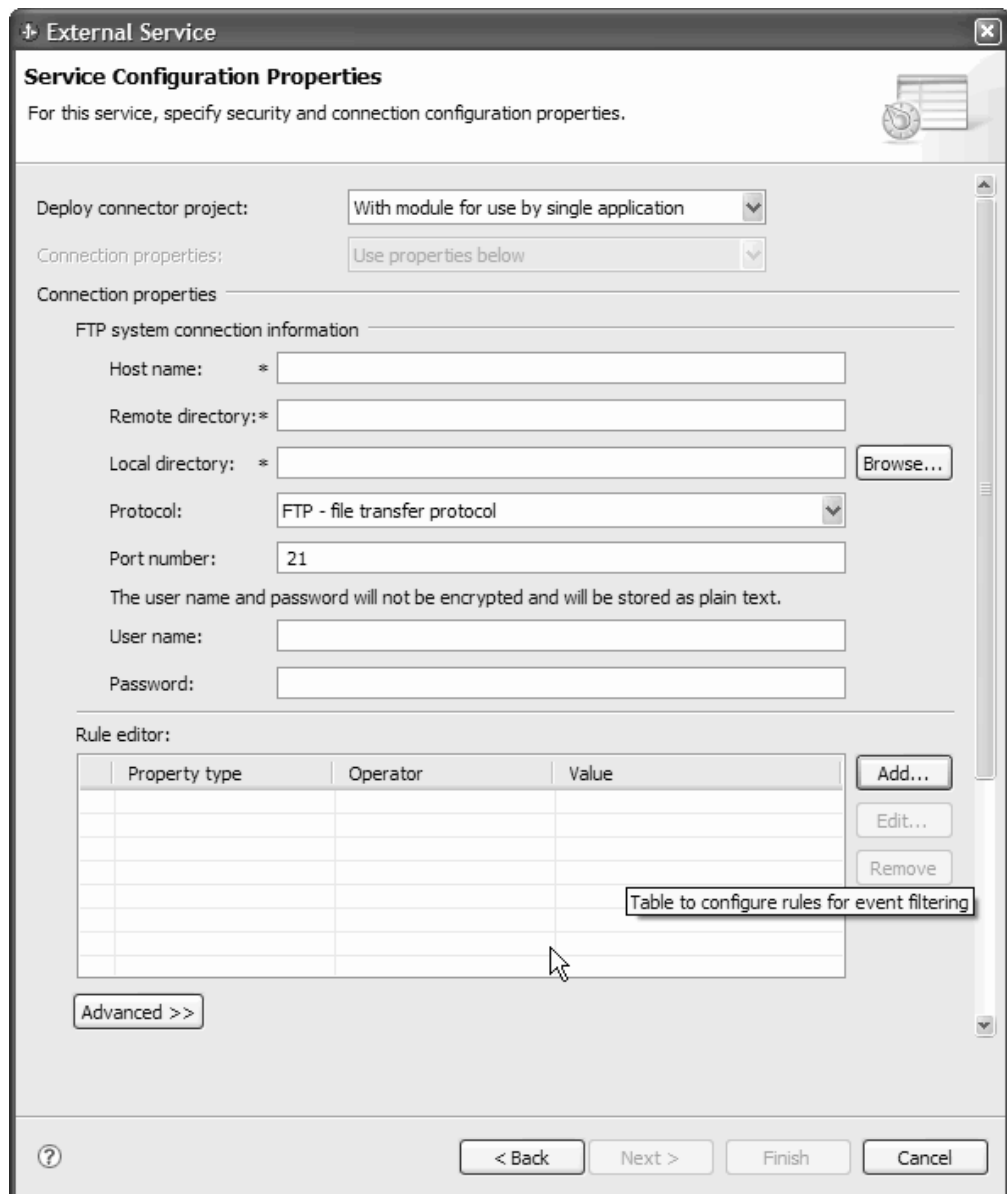


図 56. 「サービス構成プロパティ」ウィンドウ

4. ルールを構成して Inbound イベント・ファイルをフィルター処理するには、「ルール・エディター」テーブルで、「追加」または「編集」をクリックします。ルールは、プロパティ・タイプ、演算子、および値の 3 つのパラメータ

ーで構成されます。

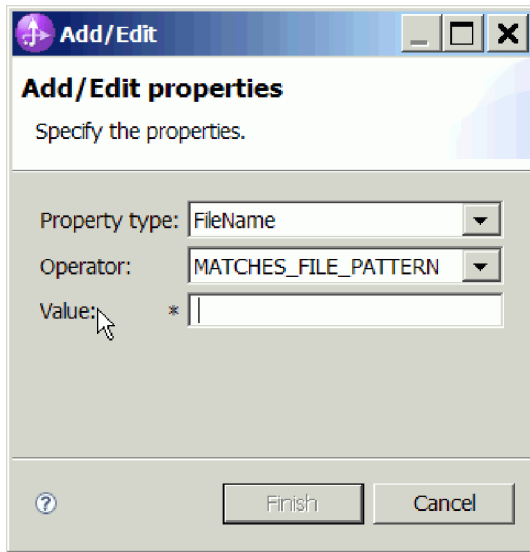


図 57. ルールの追加または編集

- a. 「プロパティ・タイプ」リストから、以下の任意のメタデータのフィルター・プロパティ・タイプを選択します。
 - FileName
 - FileSize
 - LastModified
- b. 「演算子」リストから、プロパティ・タイプに使用する演算子を選択します。各プロパティ・タイプ・メタデータごとに独自の演算子があります。FileName には、演算子として、MATCHES_FILE_PATTERN(matches pattern) および MATCHES_REGEXP(matches regular expression) が組み込まれます。FileSize には、バイト単位による GT (より大)、LT (より小)、GE (より大か等しい)、LE (より小か等しい)、EQ (等しい)、NE (等しくない) が演算子として組み込まれます。LastModified には、演算子として、曜日または時刻を示す GT (より大)、LT (より小)、GE (より大か等しい)、LE (より小か等しい)、EQ (等しい)、NE (等しくない) が組み込まれます。曜日値に使用できる演算子は、EQ のみです。
- c. イベント・ファイルをフィルター処理するための値を、「値」列に入力します。MATCHES_REGEXP 演算子には、有効な Java 正規表現を入力する必要があります。

複数のルールを構成する場合は、「プロパティ・タイプ」リストの各ルールに対して「**END-OF-RULE**」オプションを選択してください。

注: 各式は、プロパティ・フィールドで、「**END-OF-RULE**」が選択されている場合を除き、論理 **OR** 演算子でグループ化されます。1 つのルールを終了する場合、プロパティ・フィールドで、「**END-OF-RULE**」を選択する必要があるため、すべてのルールは、論理 **AND** 演算子でグループ化され、1 つのルール内のすべての式は、論理 **OR** 演算子でグループ化されます。

重要: 1つのルールのみを構成する場合、「**END-OF-RULE**」は選択しないでください。選択すると、ルールが実行時に無効になります。「**プロパティ・タイプ**」リストから「**END-OF-RULE**」を選択し、かつ、2番目のルールを選択しないと、アダプターはエラーをスローします。

詳しくは、235ページの『ルール・ベースのイベント・フィルター処理 (ruleTable)』を参照してください。

5. オプションで、「**拡張**」をクリックして拡張プロパティを指定できます。次のグループ・セクションをそれぞれ展開して、拡張プロパティを確認します。

- イベント・ポーリング構成
- イベント送達構成
- イベント・パーシスタンス構成
- 拡張接続構成
- FTP アーカイブ構成
- Socks プロキシ・サーバー接続情報
- セキュア構成
- BiDi プロパティ
- ロギングおよびトレースのプロパティ

次のセクションでは、これらの拡張プロパティ・グループで使用可能なオプションについて説明しています。

- **イベント・ポーリング構成**
 - a. 「**ポーリング期間の間隔**」に、アダプターがポーリング期間から次の期間まで待機する時間 (ミリ秒) を入力します。詳しくは、225ページの『**ポーリング期間の間隔 (ポーリング間隔)**』を参照してください。
 - b. 「**ポーリング期間内の最大イベント数**」に、各ポーリング期間で送達するイベント数を入力します。詳しくは、225ページの『**ポーリング期間内の最大イベント数 (ポーリング数量)**』を参照してください。
 - c. 「**接続が失敗した場合の再試行間隔**」に、ポーリング中に接続が失敗してから接続を再試行するまでの待機時間 (ミリ秒) を入力します。詳しくは、229ページの『**接続が失敗した場合の再試行間隔 (RetryInterval)**』を参照してください。
 - d. 「**システム接続を再試行する回数**」に、接続を再試行する回数を入力します。再試行がこの回数に達すると、ポーリング・エラーが報告されます。詳しくは、229ページの『**システム接続を再試行する回数 (RetryLimit)**』を参照してください。
 - e. ポーリング・エラーが発生したらアダプターを停止するようにしたい場合は、「**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**」を選択します。このオプションを選択しない場合、アダプターは例外をログに記録しますが、稼働し続けます。詳しくは、234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)**』を参照してください。

- f. 開始時に、アダプターが、失敗した接続を再試行するようにする場合は、「開始時に EIS 接続を再試行する」を選択します。詳しくは、228 ページの『開始時に EIS 接続を再試行する (RetryConnectionOnStartup)』を参照してください。

- イベント送達構成

- a. 「送達のタイプ」で、送達方法を選択します。この方式については、215 ページの『送達のタイプ (DeliveryType)』で説明します。
- b. イベントの送達は一回のみで、かつ 1 つのエクスポートにのみ送達されるようにする場合は、「送達は 1 回のみ」を選択します。このオプションはパフォーマンスを低下させる可能性があります、イベント送達が重複したり欠落したりすることはありません。詳しくは、213 ページの『イベント送達は 1 回のみ (AssuredOnceDelivery)』を参照してください。
- c. 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場合は、「将来のタイム・スタンプを持つイベントを処理しない」を選択します。詳しくは、219 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)』を参照してください。
- d. 「処理するイベント・タイプ」に、イベント送達の対象とするビジネス・オブジェクトのリストをコンマで区切って入力します。すべてのビジネス・オブジェクト・タイプのイベントを受信する場合は、このフィールドを空白のままにします。

例えば、エンタープライズ情報システム内で Customer および Order テーブルが変更されたときのみイベントを受信する (他のテーブルの変更時は受信しない) 場合は、このフィールドに Customer,Order を設定します。

詳しくは、217 ページの『処理するイベント・タイプ (EventTypeFilter)』を参照してください。

- e. 「失敗したイベントの再試行制限」に、アダプターがイベントに失敗としてマークを付けるまでにイベントの再送達の試行を行う回数を指定します。詳しくは、220 ページの『失敗したイベントの再試行制限 (FailedEventRetryLimit)』を参照してください。

- イベント・パーシスタンス構成

- a. アダプターがイベント・パーシスタンス・テーブルを作成するようにする場合は、「自動作成イベント・テーブル」を選択します。詳しくは、213 ページの『「自動作成イベント・テーブル」プロパティー (EP_CreateTable)』を参照してください。
- b. 「イベント・リカバリー・テーブル名」に、アダプターがイベント・パーシスタンスに使用するテーブルの名前を指定します。詳しくは、217 ページの『「イベント・リカバリー・テーブル名」プロパティー (EP_EventTableName)』を参照してください。
- c. 「イベント・リカバリー DataSource (JNDI) 名」に、イベント・パーシスタンスが JDBC データベース接続を取得するために使用するデータ・ソー

スの JNDI 名を指定します。詳しくは、216 ページの『「イベント・リカバリー DataSource (JNDI) 名」プロパティ (EP_DataSource_JNDIName)』を参照してください。

- d. 「**イベント DataSource に接続するのに使用されるユーザー名**」に、イベント・パーシスタンスがデータ・ソースからデータベース接続を取得するために使用するユーザー名を指定します。詳しくは、235 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 (EP_UserName)』を参照してください。
- e. 「**イベント DataSource に接続するのに使用されるパスワード**」に、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を指定します。詳しくは、224 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 (EP_Password)』を参照してください。
- f. 「**データベース・スキーマ名**」に、イベント・パーシスタンスで使用されるデータベースのスキーマ名を指定します。詳しくは、214 ページの『「データベース・スキーマ名」プロパティ (EP_SchemaName)』を参照してください。

• 拡張接続構成

- a. 「**このパターンを持つファイルの取得**」に、イベント・ファイルのフィルターを指定します。詳しくは、228 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 (EventFileMask)』を参照してください。
- b. 「**ソート・イベント・ファイル**」に、ポーリングされるイベント・ファイルのソート順を指定します。詳しくは、232 ページの『「ソート・イベント・ファイル」プロパティ (SortEventFiles)』を参照してください。
- c. 「**FTP サーバーで使用されるエンコード**」に、FTP サーバーのエンコードを指定します。詳しくは、216 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)』を参照してください。
- d. 「**ファイル内容のエンコード**」に、イベント・ファイルの読み取りに使用するエンコードを指定します。詳しくは、218 ページの『「ファイル内容のエンコード」プロパティ (FileContentEncoding)』を参照してください。
- e. 「**FTP サーバー接続モード**」に、ファイル転送の間に FTP サーバーが使用するデータ接続モードを指定します。詳しくは、215 ページの『「FTP サーバー接続モード」プロパティ (DataConnectionMode)』を参照してください。
- f. 「**ファイル転送タイプ**」に、Inbound 処理時に使用されるファイル転送タイプを指定します。詳しくは、219 ページの『「ファイル転送タイプ」プロパティ (FileTransferType)』を参照してください。
- g. 「**一度に取得するファイル数**」に、リモート FTP URL から取得するファイルの数を指定します。詳しくは、219 ページの『「一度に取得するファイル数」プロパティ (FTPGetQuantity)』を参照してください。
- h. 「**各ダウンロード間のポーリング期間の数**」に、アダプターが FTP サーバーをポーリングする頻度を指定します。詳しくは、220 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)』を参照してください。

- i. 「カスタム・パーサー・クラス名」に、ls 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。詳しくは、214 ページの『「カスタム・パーサー・クラス名」プロパティ (CustomParserClassName)』を参照してください。
 - j. 「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」を選択して、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します。詳しくは、218 ページの『「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)』を参照してください。
 - k. 後の処理用にビジネス・オブジェクトの内容と一緒に区切り文字を送信する場合は、「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」を選択します。詳しくは、222 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)』を参照してください。
 - l. ファイル内容の分割にサイズ (バイト単位) または区切り文字を使用する場合は、「サイズ (バイト単位) または区切り文字に基づくファイル内容の分割」を選択します。詳しくは、233 ページの『「分割関数クラス名」プロパティ』を参照してください。
 - m. 「ファイル内容を分割するための基準の指定」に、
SplittingFunctionClassName プロパティの値に基づいて、異なる値が使用されることを指定します。詳しくは、232 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)』を参照してください。
 - n. 「関数クラス名の分割」に、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します。詳しくは、233 ページの『「分割関数クラス名」プロパティ』を参照してください。
 - o. 「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」に、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。詳しくは、221 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 (FTPScriptFileExecutedBeforeInbound)』を参照してください。
 - p. 「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」に、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。詳しくは、221 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 (FTPScriptFileExecutedAfterInbound)』を参照してください。
- **FTP アーカイブ構成**
 - a. 「ローカル・アーカイブ・ディレクトリー」に、ローカル・アーカイブ・ディレクトリーの絶対パスを指定します。詳しくは、222 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 (LocalArchiveDirectory)』を参照してください。
 - b. 「ローカル・アーカイブ用のファイル拡張子」に、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を指定します。詳しくは、224 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 (OriginalArchiveExt)』を参照してください。

- c. 「ローカル・アーカイブ用の成功ファイル拡張子」に、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を指定します。詳しくは、234 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)』を参照してください。
 - d. 「ローカル・アーカイブ用の障害ファイル拡張子」に、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を指定します。詳しくは、217 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)』を参照してください。
 - e. 「リモート・アーカイブ・ディレクトリー」内。詳しくは、227 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)』を参照してください。
 - f. 「リモート・アーカイブ用のファイル拡張子」に、アダプターがリモート FTP ファイルの名前変更に使用するファイル拡張子または接尾部を指定します。詳しくは、218 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 (FTPRenameExt)』を参照してください。
- **Socks プロキシ・サーバー接続情報**
 - a. 「ホスト名」に、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を指定します。詳しくは、231 ページの『「ホスト名」プロパティ (SocksProxyHost)』を参照してください。
 - b. 「ポート番号」に、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を指定します。詳しくは、231 ページの『「ポート番号」プロパティ (SocksProxyPort)』を参照してください。
 - c. 「ユーザー名」に、プロキシ・サーバーの認証のためのユーザー名を指定します。詳しくは、231 ページの『「ユーザー名」プロパティ (SocksProxyUserName)』を参照してください。
 - d. 「パスワード」に、プロキシ・サーバーの認証のためのパスワードを指定します。詳しくは、231 ページの『「パスワード」プロパティ (SocksProxyPassword)』を参照してください。

- **セキュア構成**

SFTP サーバーのホスト鍵と、アダプターに既知のホスト鍵とを比較する場合は、次のようにします。

- a. 「SFTP プロトコルのリモート・サーバー認証を有効にする」チェック・ボックスを選択します。SFTP サーバーに対する最初の接続が試行される前に、ホスト鍵ファイルは、トラステッド・サーバーのホスト鍵に対して使用可能でなければなりません。詳しくは、「サーバー検査を使用可能にする」プロパティ (EnableServerVerification) を参照してください。
- b. 「ホスト鍵ファイル」フィールドに、ホスト鍵ファイルへの絶対ファイル・パスを指定します。ホスト鍵ファイルは、管理者によって作成され、すべてのトラステッド・サーバーのホスト鍵を格納します。ホスト鍵ファ

イル・プロパティは、アダプター・ワークステーション上のファイルを指します。詳しくは、ホスト鍵ファイル・プロパティ (HostKeyFile) を参照してください。

公開鍵認証を使用可能にする場合は、これらのプロパティを指定します。

- 1) 「**秘密鍵ファイル**」フィールドに、セキュア・シェル・サーバーに認証させるために使用される秘密鍵を指定します。詳しくは、226 ページの『**秘密鍵ファイル・プロパティ (PrivateKeyFilePath)**』を参照してください。
- 2) 「**パスフレーズ**」フィールドに、秘密鍵を暗号化して、拡張セキュリティのために使用される句を指定します。詳しくは、パスフレーズ・プロパティ (Passphrase) を参照してください。

- **BiDi プロパティ**

- **ロギングおよびトレース**

- a. アダプターのインスタンスが複数ある場合、展開して、アダプター ID に、このインスタンスに固有の値を設定します。このプロパティについて詳しくは、173 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。
 - b. ログ・ファイルとトレース・ファイルに機密のユーザー・データが書き込まれないようにする場合は、「**ログ・ファイルおよびトレース・ファイルで、ユーザー・データを「XXX」と記述する**」を選択します。詳しくは、175 ページの『**ログ・ファイルおよびトレース・ファイルで、ユーザー・データを「XXX」と記述する (HideConfidentialTrace) プロパティ**』を参照してください。
6. ウィンドウの「サービス・プロパティ」セクションで、「**関数セクター**」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. 「**関数セクター**」フィールドの横にある「**選択**」をクリックします。

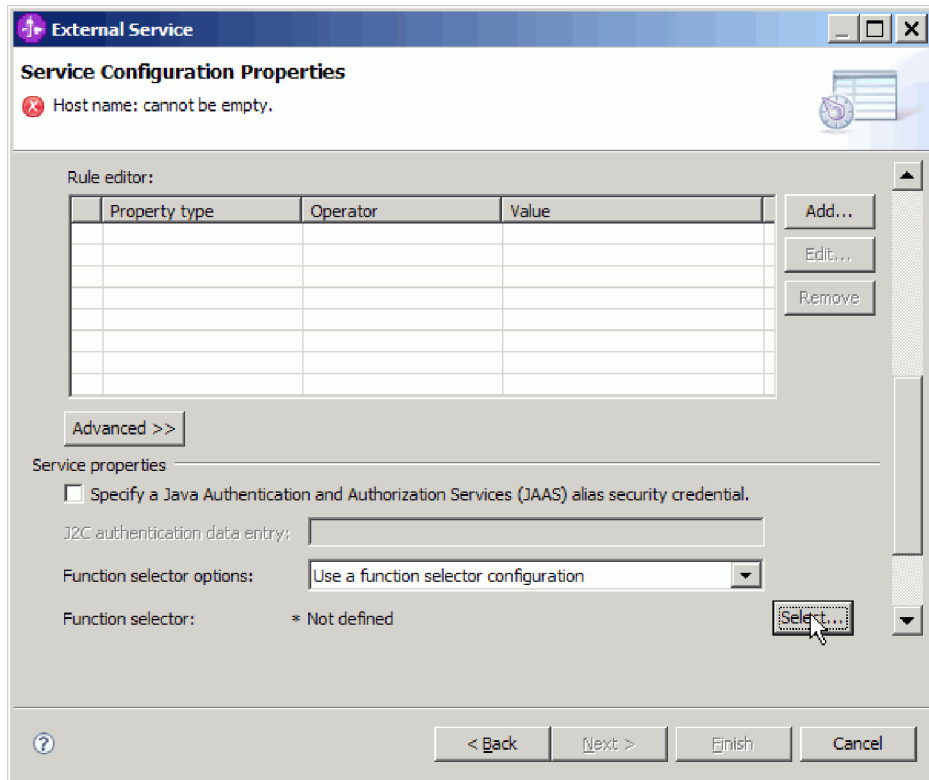


図 58. 「サービス構成プロパティ」 ウィンドウ

- b. 「関数セレクターの選択」 ウィンドウで、「**FilenameFunctionSelector** プロパティ」を選択します。「次へ」をクリックします。

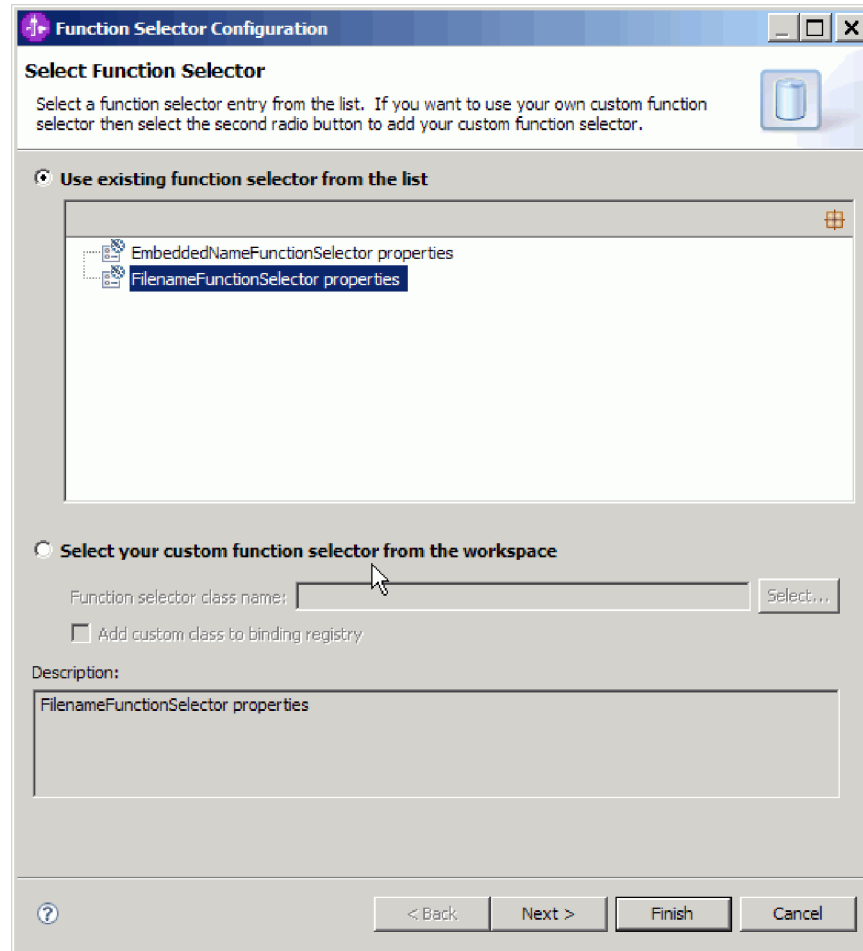


図 59. 「関数セクターの選択」 ウィンドウ

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。

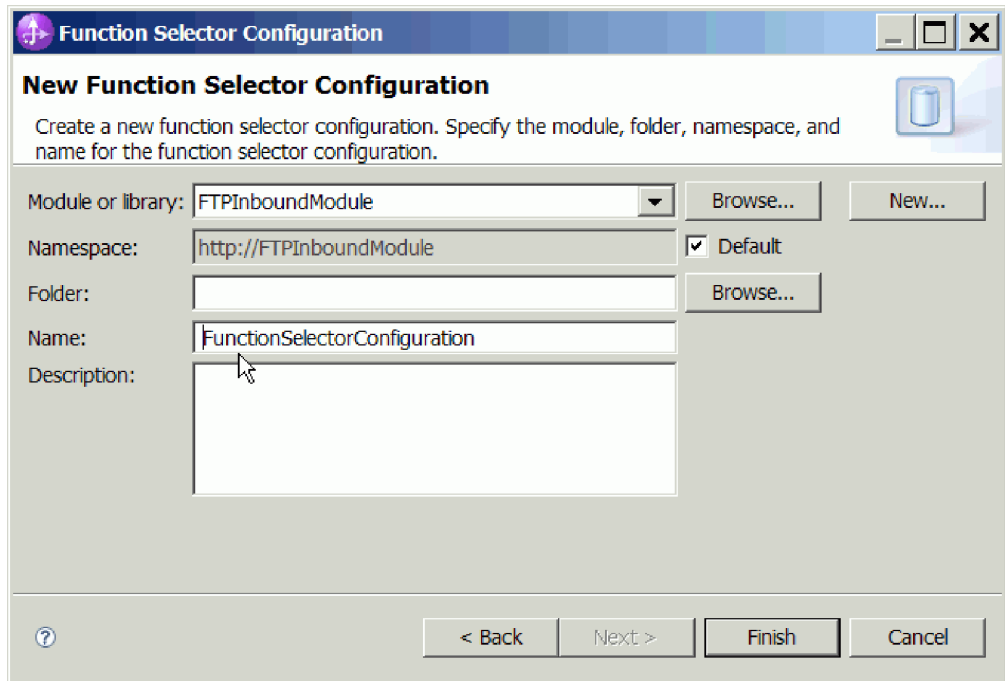


図 60. 「新規関数セクター構成」ウィンドウ

- 「関数セクター構成」ウィンドウで「終了」をクリックします。

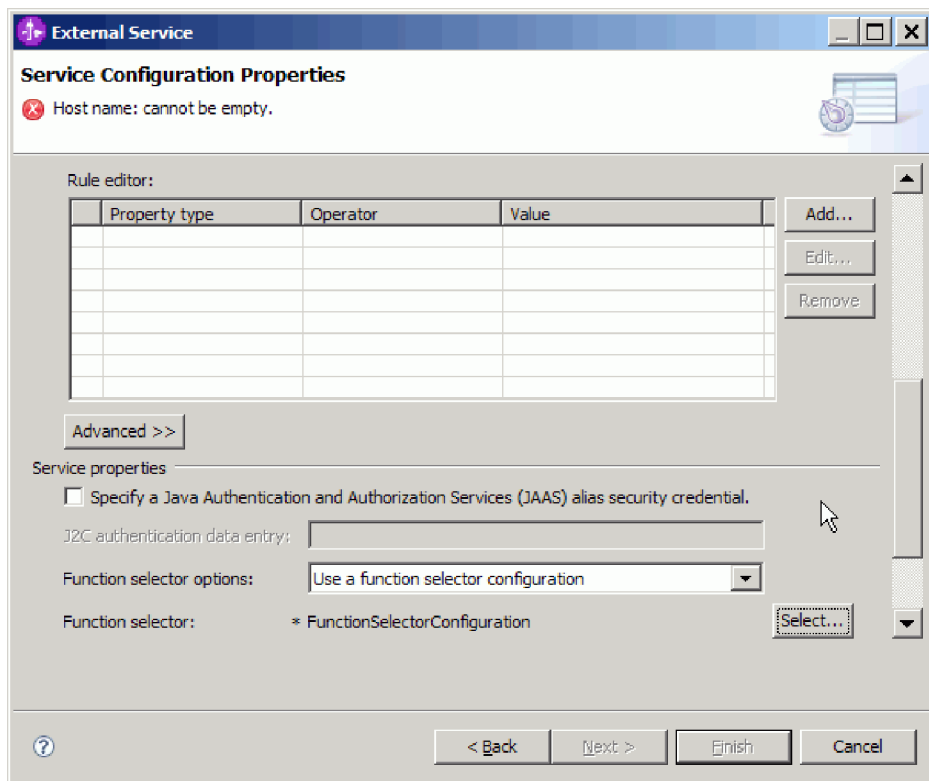


図 61. 「関数セクター選択」ウィンドウ

- 「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、90 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 62. 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。

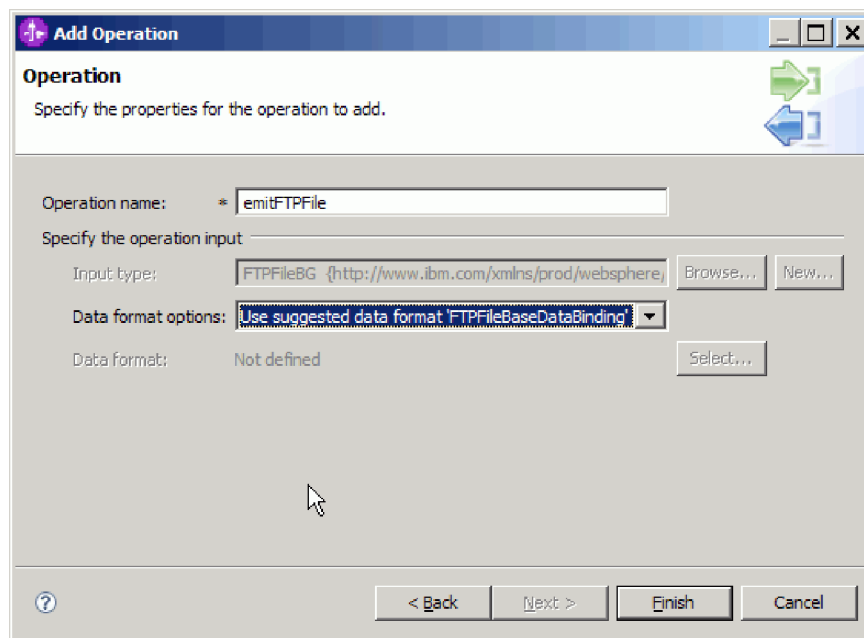


図 63. 操作への名前の指定

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、121 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構成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. 「操作」ウィンドウで、「新規」を選択して、「操作入力の指定」領域に「入力タイプ」を入力します。この操作は、初めてデータ・バインディングを設定するときに実行します。あとで同じデータ・バインディング構成を使用するには、「参照」をクリックし、その構成を選択します。

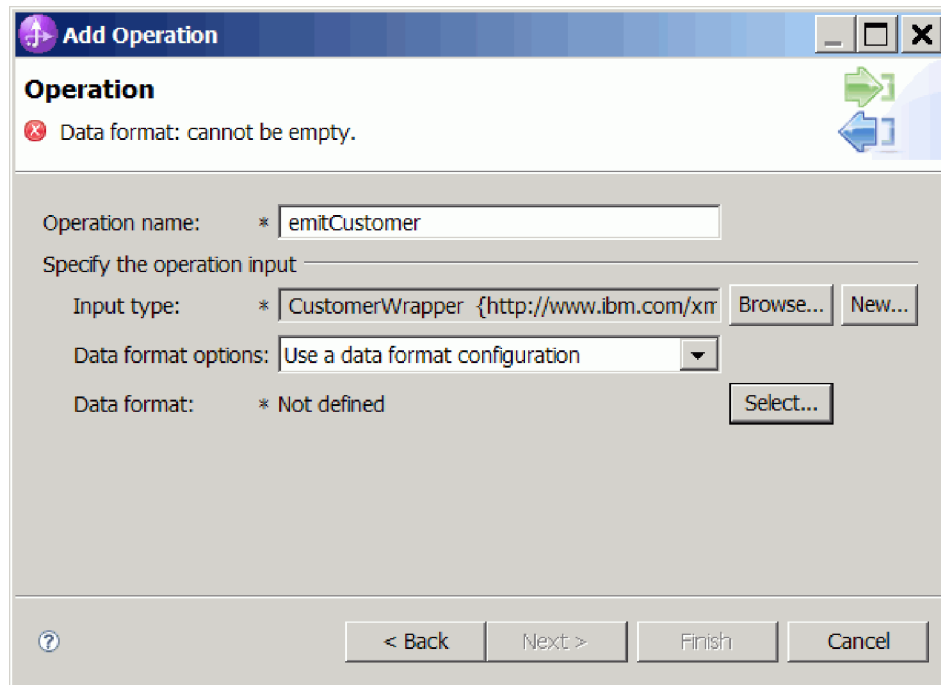


図 64. 「操作」ウィンドウ

2. オプション: 「新規データ・バインディング構成」ウィンドウには、「モジュール」にデフォルトでこのウィザードで既に入力したモジュール名が表示されています。このモジュールがデータ・バインディングを作成するモジュールでない場合は、「新規」を選択して新しいモジュールを作成します。
3. オプション: 成果物用に新しいフォルダーを選択する場合は、「参照」をクリックして、新しいフォルダーの格納場所を選択します。新しいフォルダーの場所を参照しなかった場合、成果物はモジュールのルート・ディレクトリーに作成されます。
4. 「データ・フォーマット・オプション」ドロップダウン・リストから「データ・フォーマット構成の使用」を選択します。
5. データ・バインディング構成の構成名を入力します (この例では DataBindingConfiguration を使用)。
6. 「次へ」をクリックします。 .

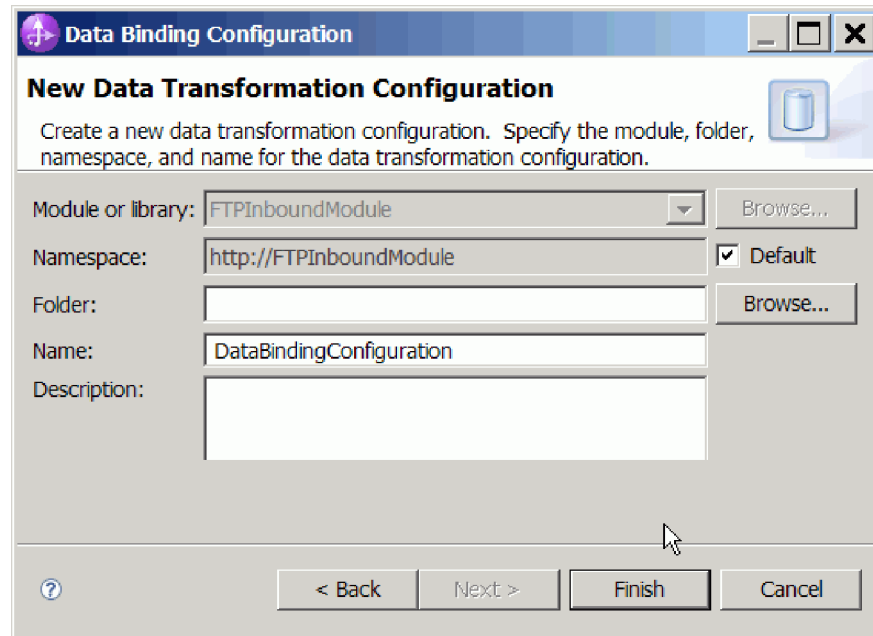


図 65. データ・バインディング構成の名前指定

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」→「リソース構成のバインディング (Binding 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. 「新規データ変換構成」ウィンドウで、「新規」を選択します。
2. データ・ハンドラー構成の名前 (この例では `DataBindingConfiguration` を使用) を「名前」フィールドに入力します。この操作は、初めてデータ・ハンドラーを設定するときに実行します。あとで同じデータ・ハンドラーを使用するには、「参照」をクリックし、そのハンドラーを選択します。
3. 「次へ」をクリックします。
4. 「データ変換プロパティ」ウィンドウで、「**バインディング・タイプ**」プロパティの横にあるドロップダウン・リストをクリックします。2つの選択項目「`DataBinding`」および「`DataHandler`」が表示されます。旧バージョンのアダプター用に作成されたデータ・バインディングを使用する場合は、「`DataBinding`」を選択します。新しいデータ・ハンドラーを構成する場合は、「`DataHandler`」を選択します。「**選択**」をクリックすると、新規データ・ハンドラー構成が作成されます。

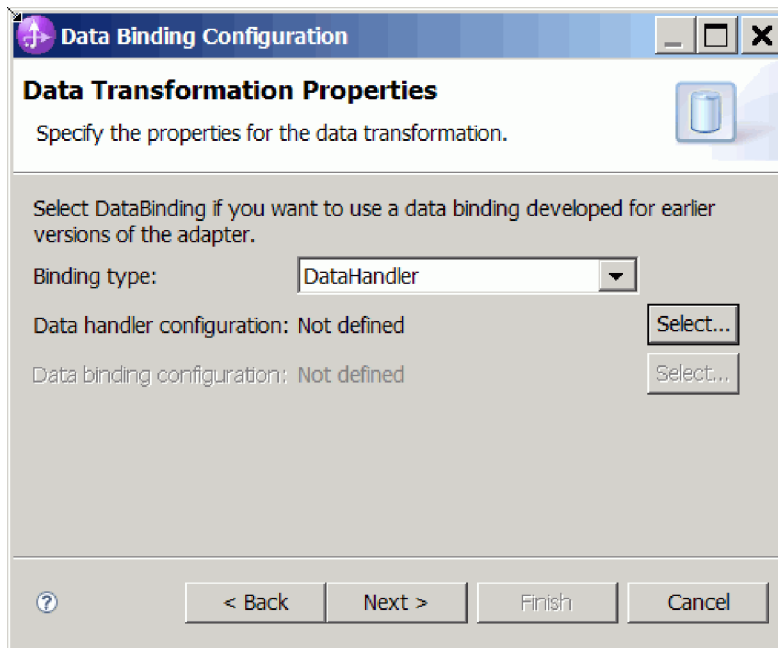


図 66. 「データ変換」ウィンドウでのバインディング・タイプの指定

5. 「次へ」をクリックします。
6. データ・ハンドラーのクラス名を選択します。「データ・フォーマット変換の選択」ウィンドウで、リスト・オプションから「**既存のデータ・フォーマット変換の使用**」をクリックします。選択可能なデータ・ハンドラー・クラスのリストが表示されます。データ・ハンドラー・クラスを選択します (この例では `XML` データ・ハンドラーを使用)。「次へ」をクリックします。

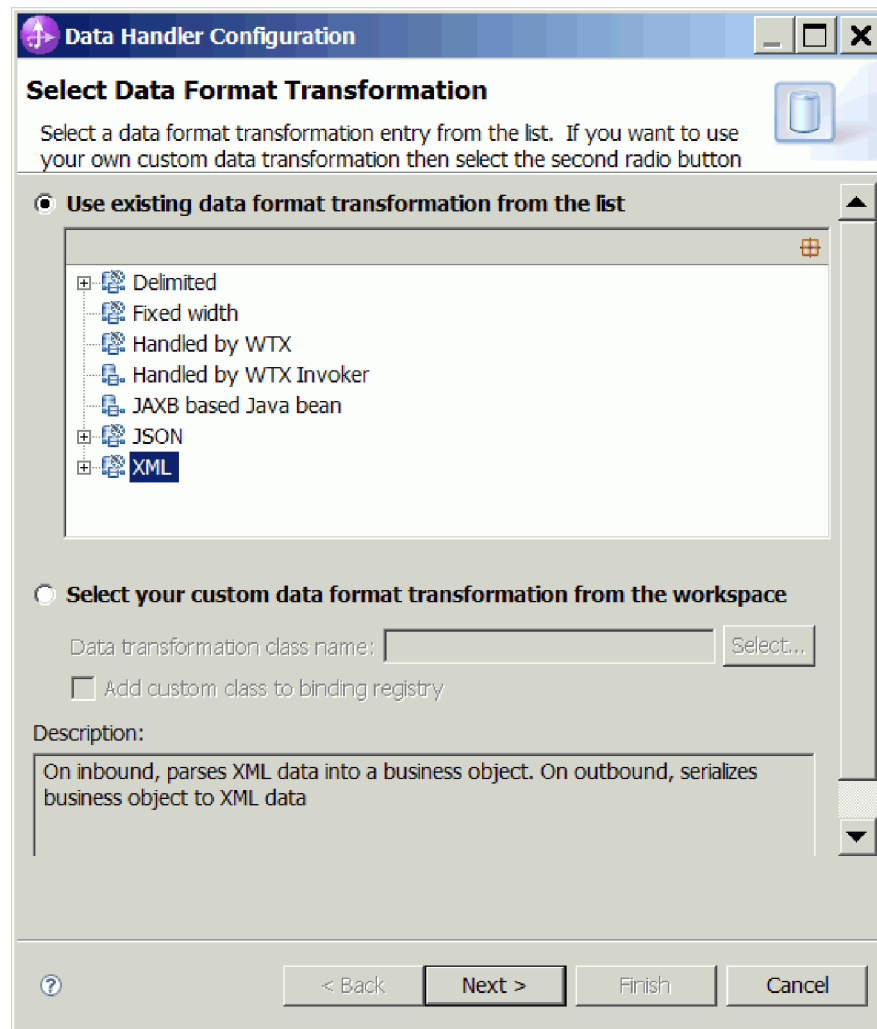


図 67. 新規データ・ハンドラー構成の作成

7. 「データ変換プロパティ」 ウィンドウで、エンコード方式を指定します。デフォルトは UTF-8 です。

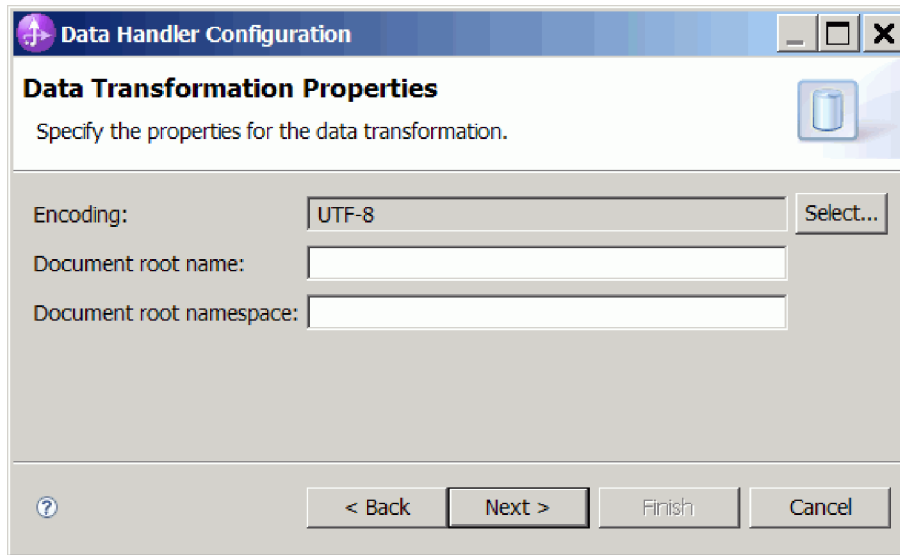


図 68. データ・ハンドラー構成のエンコード方式の指定

8. 「次へ」をクリックします。
9. 「新規データ変換構成」ウィンドウで、「新規」をクリックし、データ・ハンドラー構成の名前を入力します (この例では DataHandlerConfiguration を使用)。初めて「新規」をクリックするときには、データ・ハンドラーを設定します。あとでこのデータ・ハンドラーを使用するときには、「参照」をクリックします。

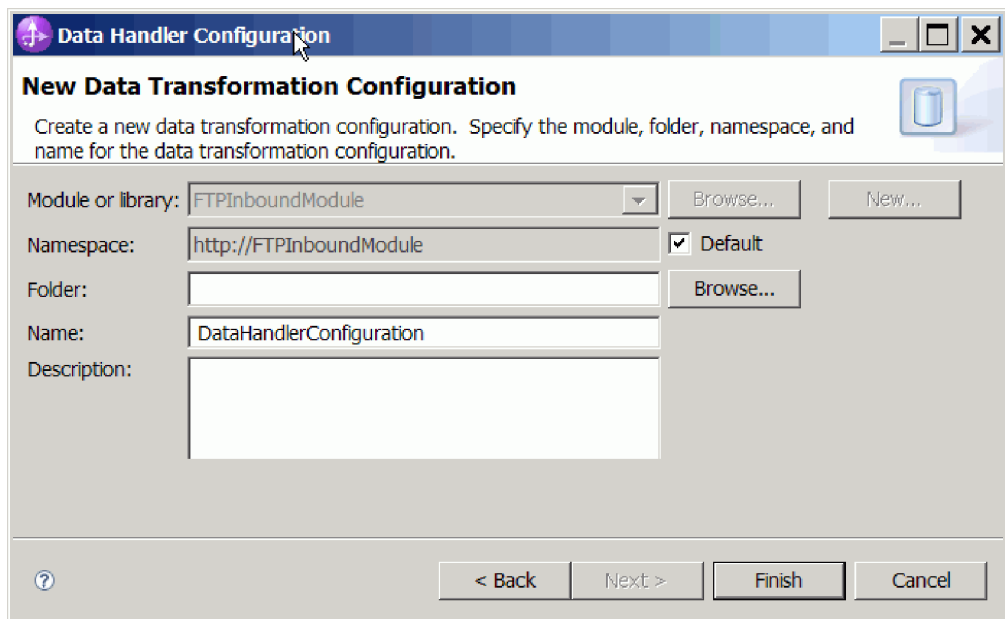


図 69. データ・ハンドラー構成の名前の指定

10. 「終了」をクリックします。
11. 「データ変換プロパティ」ウィンドウが表示されます。

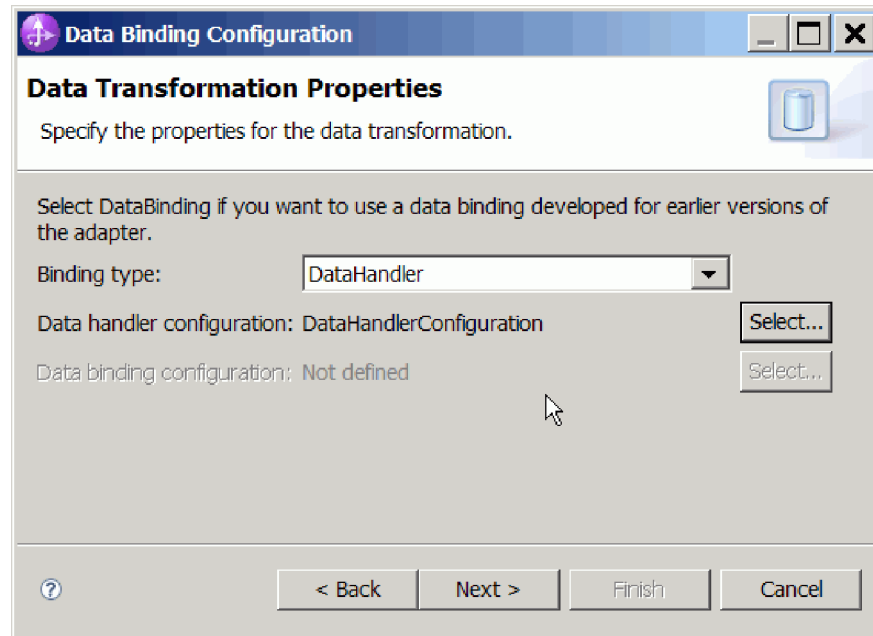


図 70. 「データ変換プロパティ」ウィンドウ

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 が成果物およびエクスポートを生成します。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・エディターによる対話仕様プロパティの変更

サービスの生成後にアダプター・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

アダプターに対してサービスを生成するには、あらかじめ外部サービス・ウィザードを使用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

アダプターのサービスを生成後に、対話仕様プロパティの変更が必要になる場合があります。対話仕様プロパティはオプションですが、特定のビジネス・オブジェクトの特定の操作に対して、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・ウィザードによって生成されるすべての親ビジネス・オブジェクトのデフォルトとして表示されます。これらのプロパティは、EAR ファイルをエクスポートする前に変更できます。アプリケーションをデプロイした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、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・エディターでインターフェースをクリックします。(追加のクリックをしない限り、モジュールのプロパティが表示されています。)
4. 「プロパティ」タブをクリックします。(ダイアグラム内でインターフェースを右クリックし、「プロパティを表示」をクリックすることもできます。)
5. 「バインディング」で、「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の組み合わせごとに 1 つずつ表示されます。
6. 変更する対話仕様プロパティを持つメソッドを選択します。
7. 「拡張」をクリックし、「汎用」タブでプロパティを変更します。 変更する対話仕様プロパティを持つ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ている対話仕様プロパティが変更されました。

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、モジュールおよびアダプターを構成するファイルを、実稼働またはテストのための動作環境に配置します。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のデプロイ先には、テスト環境と実稼働環境があります。

WebSphere Integration Developer では、モジュールをテスト環境内の 1 つ以上のサーバーにデプロイできます。通常は、これがビジネス・インテグレーション・モジュールの実行およびテストを行うための最も一般的な手法です。ただし、WebSphere Process Server または WebSphere Enterprise Service Bus 上で管理コンソールまたはコマンド行ツールを使用して、サーバーへのデプロイメント用のモジュールを EAR ファイルとしてエクスポートすることもできます。

テスト用のモジュールのデプロイ

WebSphere Integration Developer では、組み込みアダプターを内蔵するモジュールをテスト環境にデプロイし、サーバー構成の編集、サーバーの始動および停止、モジュール・コードのテストによるエラー有無の確認などの作業を実行できるサーバー・ツールと連携できます。テストは通常、コンポーネントのインターフェース操作について実行されますが、このテストを実行すると、コンポーネントが正しく実装され、参照先が正しく接続されているかどうかを判断できます。

このタスクを実行する理由および時期

Inbound 処理をテストするためのターゲット・コンポーネントの生成および接続

Inbound 処理用のアダプターが組み込まれているモジュールをテスト環境にデプロイする前に、まずターゲット・コンポーネントを生成して接続する必要があります。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・モジュールを生成してあるはずですが。

このタスクを実行する理由および時期

Inbound 処理のためにターゲット・コンポーネントを生成して接続する必要があるのは、テスト環境のみです。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 のアセンブリー・エディターを使用して、エクスポート・コンポーネントを (2 つのコンポーネントを接続している) ターゲット・コンポーネントに接続します。アダプターはこのワイヤーを使用して、(エクスポート・コンポーネントからターゲット・コンポーネントへ) 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 パースペクティブで、「アセンブリー・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、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 + **InboundInterface** になります。

インターフェースにより、呼び出すことができる操作と渡されるデータ (入力引数、戻り値、例外など) が指定されます。 **InboundInterface** コンポーネントには、Inbound 処理をサポートするためにアダプターが必要とする操作が格納されています。また、このコンポーネントは 外部サービス・ウィザード を実行すると作成されます。

- b. 「コンポーネント」を展開して「型なしコンポーネント」を選択し、そのコンポーネントをアセンブリー・ダイアグラムまでドラッグして、新規コンポーネントを作成します。

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、そのコンポーネントをアセンブリー・ダイアグラムに表示します。
2. コンポーネントを接続します。
 - 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にドラッグします。これにより、次の図に示すように、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線を引くことができます。
 - b. アセンブリー・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」を選択します。

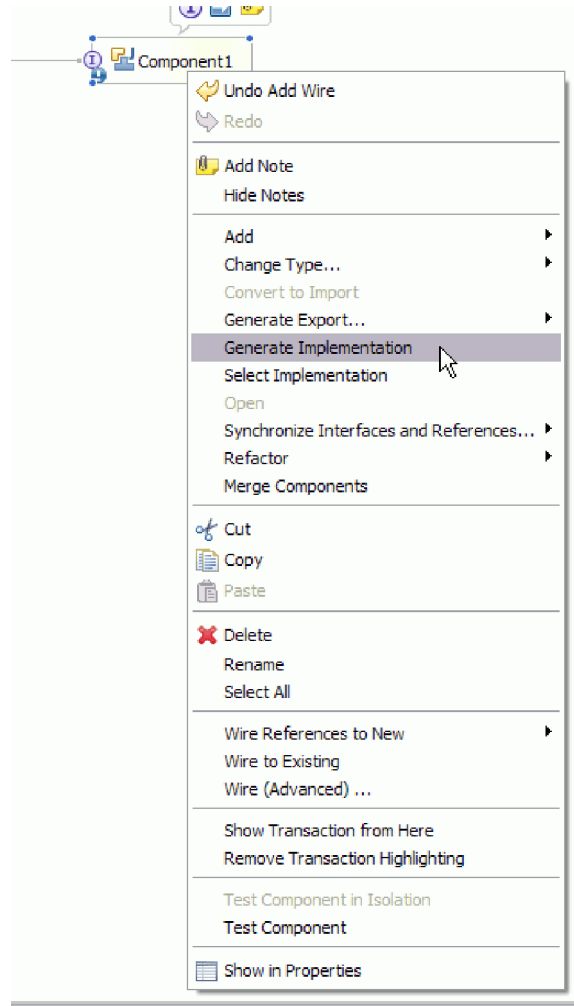


図 71. Java 実装環境の生成

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

別のタブに Java 実装環境が表示されます。

- c. **オプション:** print ステートメントを追加して、エンドポイント・メソッドのそれぞれのエンドポイントで受信したデータ・オブジェクトを出力します。
- d. 「ファイル」 → 「保管」をクリックして、変更内容を保存します。

次のタスク

テストを行うモジュールのデプロイを続行します。

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

テストしている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は、アダプターによるイベント送信先となるターゲット・コンポーネント を生成して、そこに接続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

ご使用のモジュールと、モジュールによるアダプターの使用をテストするため、その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: 「サーバー・ビュー」にサーバーがない場合は、以下の手順を実行し、新規サーバーを追加して定義します。
 - a. カーソルを「サーバー・ビュー」の内側に置き、右クリックして、「新規」→「サーバー」を選択します。
 - b. 「新規サーバーの定義」ウィンドウで、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」をクリックして、サーバーを公開します。
2. サーバーにモジュールを追加します。
 - 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 画面の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、「開始」を選択します。
3. サーバーの状況が「開始済み」である場合は、サーバーを右クリックし、「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、対象のプロジェクトを選択して「追加」をクリックします。 プロジェクトは、「使用可能プロジェクト」のリストから「構成プロジェクト」の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

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 Outbound 処理用モジュールのテスト

Outbound 処理用のアセンブル済みモジュールおよびアダプターを、WebSphere Integration Developer の統合テスト・クライアントを使用してテストします。

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

モジュールのテストは、通常、コンポーネントのインターフェース操作について実行されますが、このテストを実行すると、コンポーネントが正しく実装され、参照先が正しく接続されているかどうかを判断できます。

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」 → 「テスト・モジュール」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『モジュールおよびコンポーネントのテスト』のトピック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納得したら、モジュールおよびアダプターを実稼働環境にデプロイできます。

実稼働のための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したモジュールを、実稼働環境で WebSphere Process Server または WebSphere Enterprise Service Bus にデプロイする処理は、2 段階構成になっています。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 または WebSphere Enterprise Service Bus 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。

このタスクを実行する理由および時期

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ないが、サーバー・インスタンス内にデプロイされているすべてのアプリケーションに対してアダプターを使用可能にする場合は、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ことが必要になります。RAR ファイルとは、Java 2 Connector (J2C) アーキテクチャーに合わせてリソース・アダプターを圧縮するときに使用する Java アーカイブ (JAR) ファイルのことです。

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、「コネクタ・プロジェクトのデプロイ」を「複数アダプターが使用するサーバー上 (On server for use by multiple adapters)」に設定してあるはずです。

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、そのアダプターは、サーバー・ランタイムで実行されているすべての J2EE アプリケーション・コンポーネントで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」 をクリックします。
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。

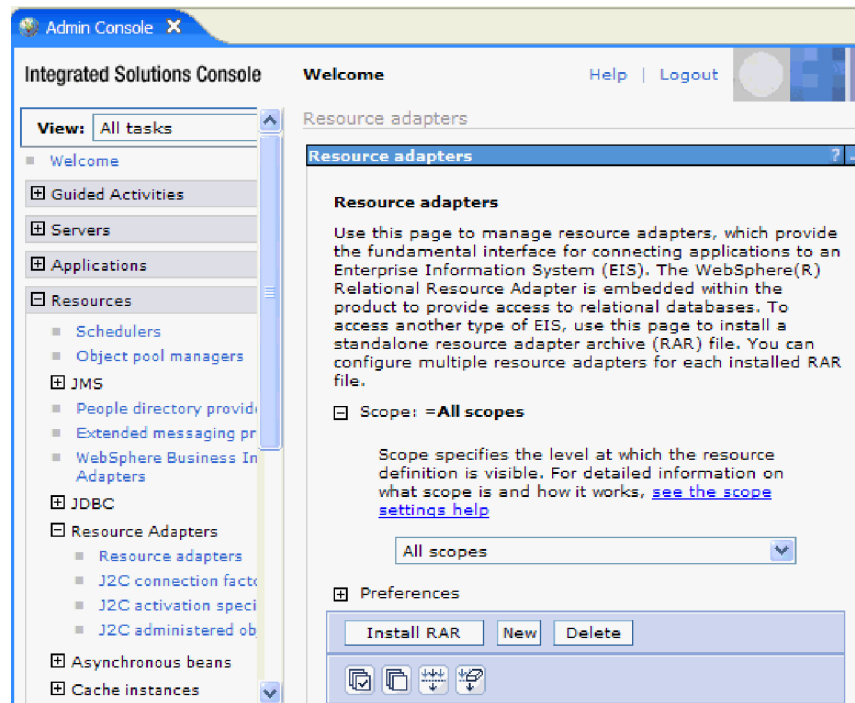


図 72. 「リソース・アダプター」 ページの「RAR のインストール」 ボタン

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、ご使用のアダプターの RAR ファイルへ移動します。

RAR ファイルは、通常、`WID_installation_directory/ResourceAdapters/adapter_name/deploy/adapter.rar` のパスにインストールされます。

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

次の手順は、サーバーにデプロイできる EAR ファイルとしてモジュールをエクスポートすることです。

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、「Java EE」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。
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モジュールと同じ名前が付けられますが、名前の末尾に「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. 必要に応じて、ソース・ファイルをエクスポートする場合は、「ソース・ファイルのエクスポート」を選択します。このオプションは、EAR ファイルのほかにソース・ファイルをエクスポートする場合に表示されます。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ているファイルがあります。
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。

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 または WebSphere Enterprise Service Bus 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・プロセスの最終手順です。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、インストール済みアプリケーションの一部として稼働します。

始める前に

モジュールを WebSphere Process Server または WebSphere Enterprise Service Bus にインストールするには、その前にモジュールを 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、次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 または WebSphere Enterprise Service Bus 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」 → 「新規アプリケーションのインストール」をクリックします。



図 73. 「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

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構成プロパティーを変更するには、実行時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に使用)、Managed Connection Factory プロパティー (Outbound 処理に使用)、および活動化仕様プロパティー (Inbound 処理に使用) を更新できます。

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、こ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、その値を変更または設定します。

始める前に

アダプター・モジュールを WebSphere Process Server または WebSphere Enterprise Service Bus 上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・プロパティーとは、すべての WebSphere アダプターが共用するデフォルト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、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。
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、プロパティーを変更するアダプター・モジュールの名前をクリックします。「構成」ページが表示されます。

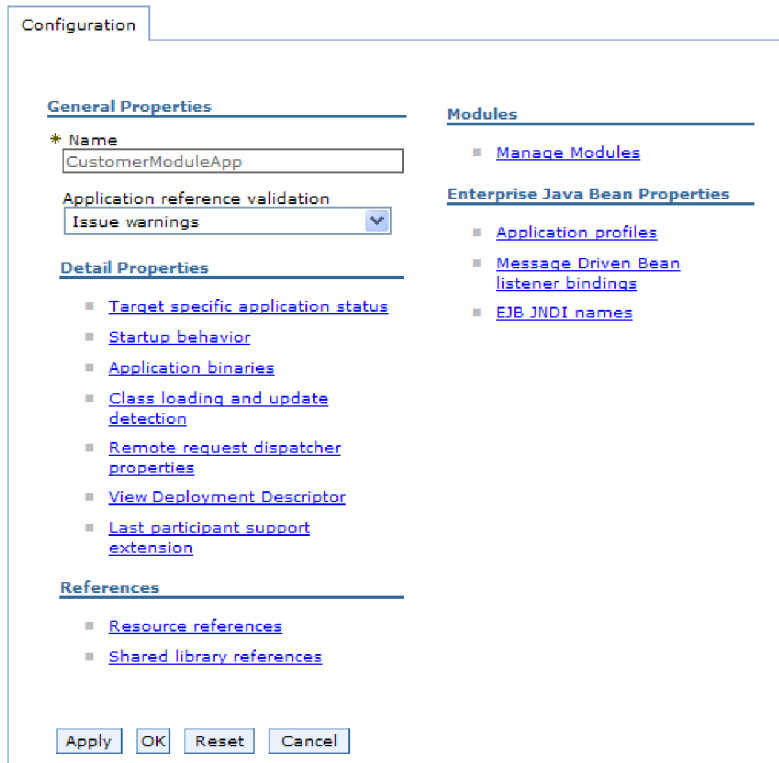


図 74. 「構成」タブでの「モジュールの管理」の選択

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。
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. 次のページで、「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、173 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。選択したプロパティの「構成」ページが表示されます。例えば、「logNumberOfFiles」プロパティをクリックすると、下図のような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 75. logNumberOfFiles プロパティの「構成」タブ

- b. 「値」フィールドの値の内容を変更するか、フィールドが空の場合は値を入力します。
 - 「値」フィールドの数値を変更して、プロパティの説明を追加できます。
 - 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」ボックス内の「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ているリソース・アダプター・プロパティが変更されました。

組み込みアダプターの Managed (J2C) Connection Factory 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、このアダプターの Managed Connection Factory 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、その値を変更または設定します。

始める前に

アダプター・モジュールを WebSphere Process Server または WebSphere Enterprise Service Bus 上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

Managed Connection Factory 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、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。
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」リストで、プロパティを変更するアダプター・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。

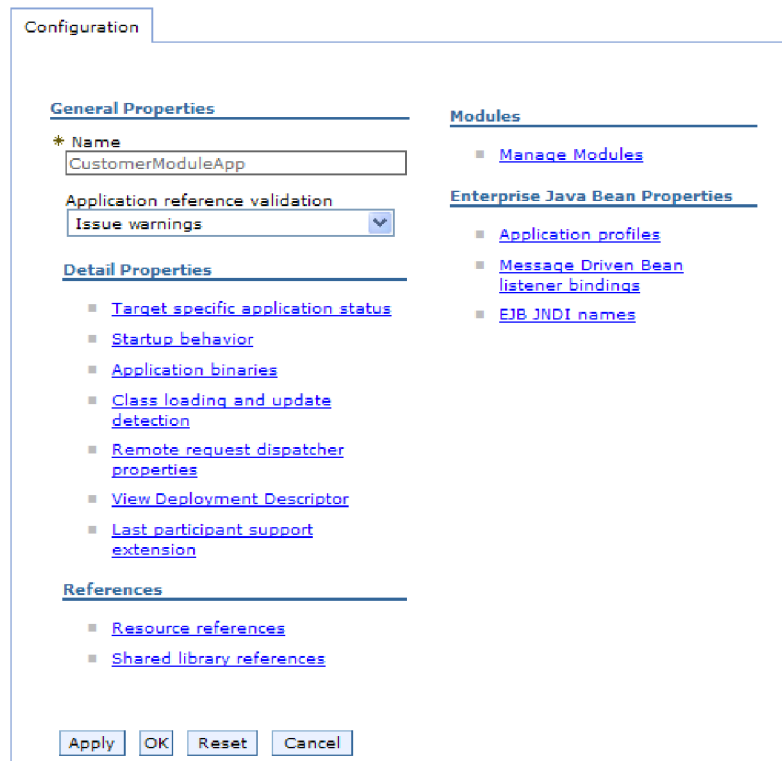


図 76. 「構成」タブでの「モジュールの管理」の選択

5. **IBM WebSphere Adapter for FTP**をクリックします。
6. 「追加プロパティ」リストで、「リソース・アダプター」をクリックします。
7. 次のページで、「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」リストで、「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 に特有の J2C 接続ファクトリー・プロパティです。接続プールおよび拡張接続ファクトリー・プロパティは、ユーザーが独自にアダプターを作成する場合に構成するプロパティです。

10. 変更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、178 ページの『Managed (J2C) Connection Factory 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」フィールドの値の内容を変更するか、フィールドが空の場合は値を入力します。
 - 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」ボックス内の「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 Managed Connection Factory プロパティが変更されます。

組み込みアダプターの活動化仕様プロパティの設定

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、そのアダプターの活動化仕様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、その値を変更または設定します。

始める前に

アダプター・モジュールを WebSphere Process Server または WebSphere Enterprise Service Bus 上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

活動化仕様プロパティは、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、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。
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、プロパティを変更するアダプター・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。

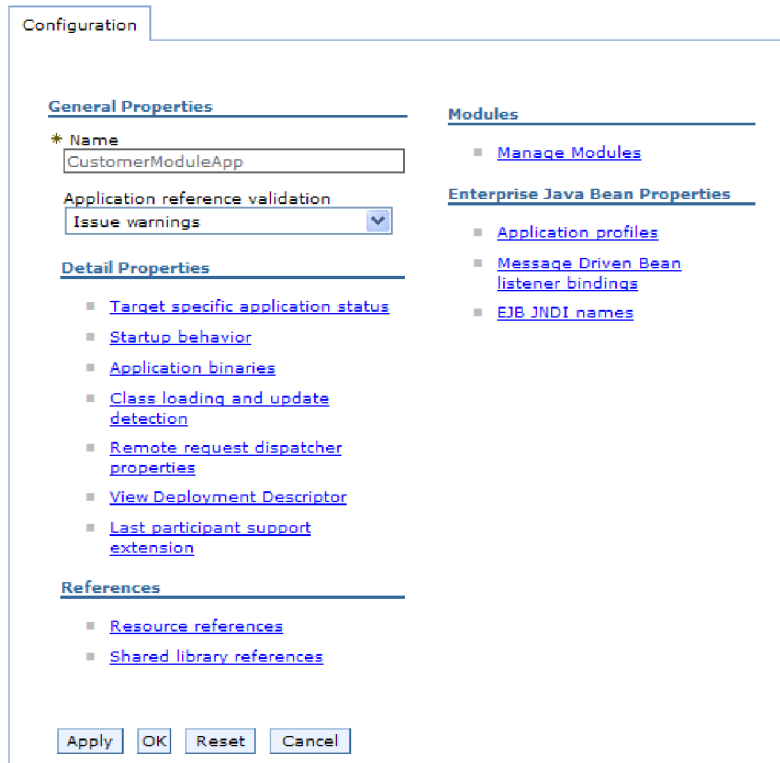


図 77. 「構成」タブでの「モジュールの管理」の選択

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. 次のページで、「追加プロパティ」リストから、「**J2C 活動化仕様**」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ている活動化仕様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 活動化仕様のカスタム・プロパティ**」をクリックします。
10. 変更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、209 ページの『活動化仕様プロパティ』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」フィールドの値の内容を変更するか、フィールドが空の場合は値を入力します。
 - 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」ボックス内の「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ている活動化仕様プロパティーが変更されました。

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・アダプターのインストール後に構成プロパティーを設定するには、実行時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的な情報を入力して、(汎用のアダプター操作に使用される) リソース・アダプター・プロパティーを設定します。アダプターを **Outbound** 操作に使用する場合は、接続ファクトリーを作成して、それに対してプロパティーを設定します。アダプターを **Inbound** 操作に使用する場合は、活動化仕様を作成して、それに対して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、その値を変更または設定します。

始める前に

アダプターを WebSphere Process Server または WebSphere Enterprise Service Bus に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・プロパティーとは、すべての WebSphere アダプターが共用するデフォルト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、以下の手順を実行します。

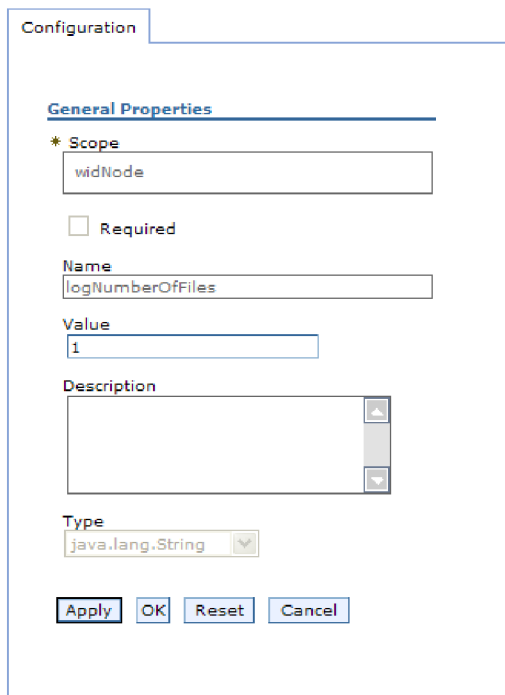
このタスクの手順

1. 管理コンソールを開始します。
2. 「リソース」 → 「リソース・アダプター」 → 「リソース・アダプター」 をクリックします。
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」 リストで、「**カスタム・プロパティー**」 をクリックします。
5. 変更する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、173 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**」 フィールドの値の内容を変更するか、フィールドが空の場合は値を入力します。

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



The screenshot shows a configuration window titled 'Configuration' with a 'General Properties' section. The 'Scope' field is set to 'widNode'. There is an unchecked 'Required' checkbox. The 'Name' field contains 'logNumberOfFiles'. The 'Value' field contains '1'. The 'Description' field is an empty text area. The 'Type' dropdown menu is set to 'java.lang.String'. At the bottom, there are four buttons: 'Apply', 'OK', 'Reset', and 'Cancel'.

図 78. logNumberOfFiles プロパティの「構成」タブ

「値」フィールドの数値を変更して、プロパティの説明を追加できます。

- 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ているリソース・アダプター・プロパティが変更されました。

スタンドアロン・アダプターの Managed (J2C) Connection Factory 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、その値を変更または設定します。

始める前に

アダプターを WebSphere Process Server または WebSphere Enterprise Service Bus に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

Managed Connection Factory 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、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。
2. 「リソース」 → 「リソース・アダプター」 → 「リソース・アダプター」をクリックします。
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」リストで、「**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、既存の接続ファクトリーのリストから選択に進んでください。

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ときに「事前定義された接続プロパティを使用する」を選択していた場合は、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」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 と入力できます。
- c. 「JNDI 名」に値を入力します。例えば、com/eis/AdapterCF と入力できます。
- 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。

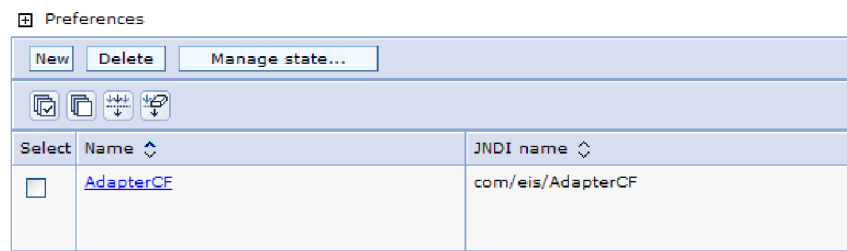


図 79. リソース・アダプターで使用するユーザー定義接続ファクトリー

6. 接続ファクトリーのリストで、使用するものをクリックします。

7. 「追加プロパティ」リストで、「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 に特有の J2C 接続ファクトリー・プロパティです。接続プールおよび拡張接続ファクトリー・プロパティは、ユーザーが独自にアダプターを作成する場合に構成するプロパティです。

8. 変更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、178 ページの『Managed (J2C) Connection Factory 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」フィールドの値の内容を変更するか、フィールドが空の場合は値を入力します。
 - c. 「OK」をクリックします。
9. プロパティの設定が終了したら、「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」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ている Managed Connection Factory プロパティが設定されます。

スタンドアロン・アダプターの活動化仕様プロパティの設定

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、その値を変更または設定します。

始める前に

アダプターを WebSphere Process Server または WebSphere Enterprise Service Bus に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

活動化仕様プロパティは、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、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。
2. 「リソース」 → 「リソース・アダプター」 → 「リソース・アダプター」をクリックします。

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」リストで、「**J2C 活動化仕様**」をクリックします。
5. 既存の活動化仕様を使用する場合は、既存の活動化仕様のリストから選択に進んでください。

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ときに「事前定義された接続プロパティを使用する」を選択していた場合は、活動化仕様を作成する必要はありません。

活動化仕様を作成する場合は、以下の手順を実行します。

- a. 「**新規**」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、活動化仕様の名前を入力します。例えば、AdapterAS と入力できます。
- c. 「**JNDI 名**」に値を入力します。例えば、com/eis/AdapterAS と入力できません。
- 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。

新規に作成された活動化仕様が表示されます。

6. 活動化仕様のリストで、使用するものをクリックします。
7. 「追加プロパティ」リストで、「**J2C 活動化仕様のカスタム・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、次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、209 ページの『活動化仕様プロパティ』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**」フィールドの値の内容を変更するか、フィールドが空の場合は値を入力します。
 - c. 「**OK**」をクリックします。
9. プロパティの設定が終了したら、「**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活動化仕様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、サーバーが始動すると、アプリケーションは自動的に開始します。

このタスクを実行する理由および時期

アプリケーションが使用するのが組み込みアダプターの場合でもスタンドアロン・アダプターの場合でも、アプリケーションを開始するには、以下の手順に従います。組み込みアダプターを使用するアプリケーションの場合、アダプターはアプリケーションの開始時に開始されます。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の場合、アダプターはアプリケーション・サーバーの始動時に開始されます。

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」をクリックします。

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。アプリケーション名は、インストールした EAR ファイルの名前からファイル拡張子 .EAR を除いたものです。
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され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、サーバーが停止すると、アプリケーションは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

アプリケーションが使用するのが組み込みアダプターの場合でもスタンドアロン・アダプターの場合でも、アプリケーションを停止するには、以下の手順に従います。アプリケーションと組み込みアダプターの組み合わせの場合、アダプターはアプリケーションの停止時に停止します。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の場合、アダプターはアプリケーション・サーバーの停止時に停止します。

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」をクリックします。

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。アプリケーション名は、インストールした EAR ファイルの名前からファイル拡張子 .EAR を除いたものです。
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」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) を通して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。使用するアダプター用に PMI を構成した後、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、150 ページの『Common Event Infrastructure (CEI) を使用した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。

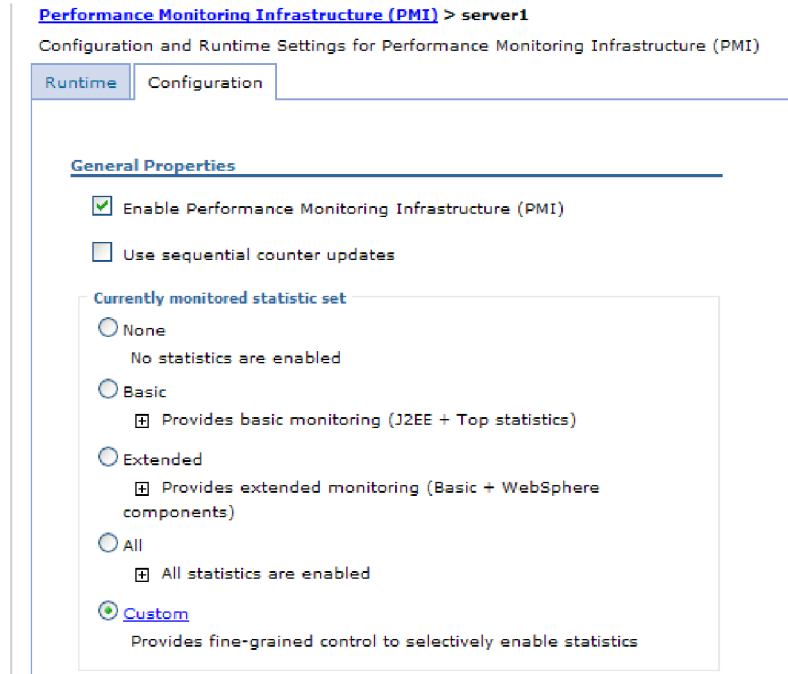


図 80. Performance Monitoring Infrastructure の使用可能化

- 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。

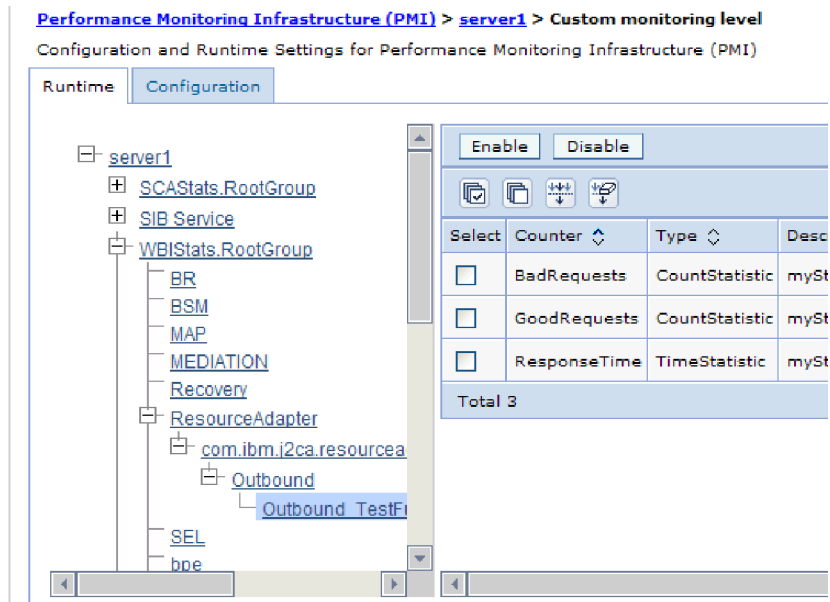


図 81. PMI の構成に使用される「ランタイム」タブ

- 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することができるようになりました。

Common Event Infrastructure (CEI) を使用したトレースの使用可能化

アダプターは、サーバー内に組み込まれたコンポーネントである Common Event Infrastructure を使用して、ポーリング周期の開始または停止などの重要なビジネス・イベントに関するデータを通知できます。 イベント・データの書き込み先は、構成設定に応じてデータベースまたはトレース・ログ・ファイルになります。

このタスクを実行する理由および時期

このタスクの手順

1. 管理コンソールにて、「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」を選択します。
3. サーバーのリストにて、ご使用のサーバーの名前をクリックします。
4. 「ログ詳細レベルの変更」ボックスで、アダプターによるイベント・データの書き込み先にする CEI データベースの名前 (例えば、WBIEventMonitor.CEI.ResourceAdapter.*) またはトレース・ログ・ファイルの名前 (例えば、WBIEventMonitor.LOG.ResourceAdapter.*) をクリックします。
5. アダプターを使用し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ビジネス・イベントの詳細レベルを選択し、(必要に応じて) メッセージおよびトレースに関連付けられている詳細レベルの細分度を調整します。
 - **ロギングなし。** イベント・ロギングをオフにします。
 - **メッセージのみ。** アダプターはイベントを通知します。
 - **すべてのメッセージおよびトレース。** アダプターは、イベントの詳細を通知します。
 - **メッセージとトレースのレベル。** イベントに関連付けられているビジネス・オブジェクト・ペイロードについてアダプターが通知する詳細度を制御するための設定です。詳細度を調整する場合は、以下のいずれかを選択してください。
 - 詳細 - 中。** アダプターはイベントを通知しますが、ビジネス・オブジェクト・ペイロードについては通知しません。
 - 詳細 - 高。** アダプターは、イベントおよびビジネス・オブジェクト・ペイロードの説明を通知します。
 - 詳細 - 最高。** アダプターは、イベントおよびビジネス・オブジェクト・ペイロード全体を通知します。
6. 「OK」をクリックします。

結果

イベント・ロギングが使用可能になります。CEI 項目は、トレース・ログ・ファイル内で参照できます。または、管理コンソール内で Common Base Event Browser を使用して表示することもできます。

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・データは、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. サーバーのリストにて、ご使用のサーバーの名前をクリックします。
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. 「WBIStatsRootGroup」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。

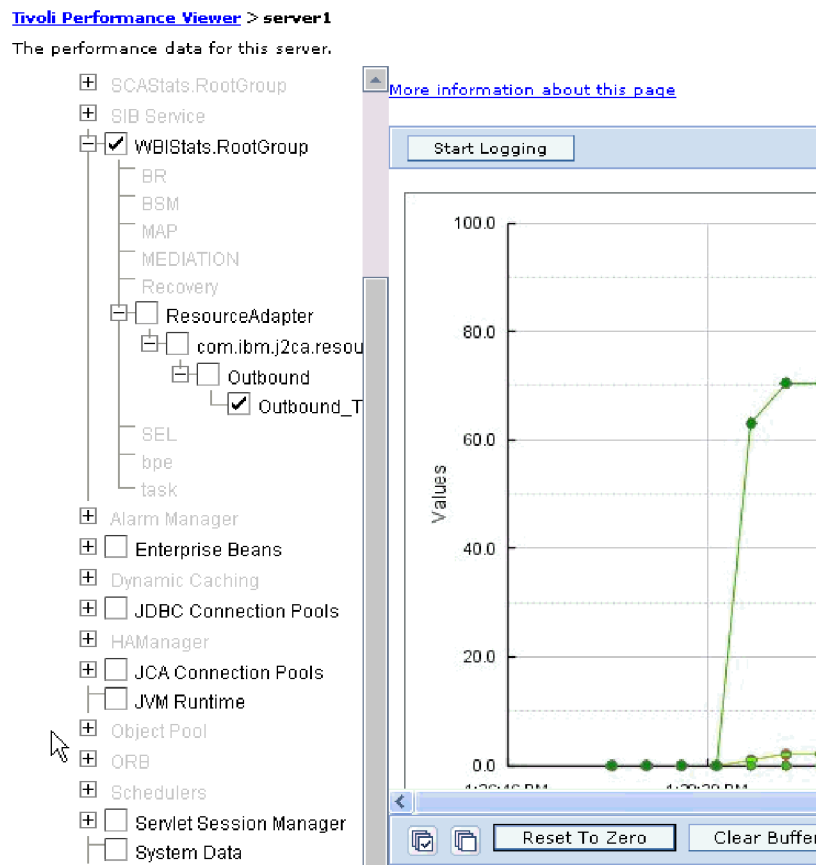


図 82. グラフ表示によるアダプターのパフォーマンス統計

第 8 章 トラブルシューティングおよびサポート

一般的なトラブルシューティング技術やセルフ・ヘルプ情報は、問題の確認および早期解決に役立ちます。

Log and Trace Analyzer のサポート

アダプターは、Log and Trace Analyzer で表示できるログ・ファイルとトレース・ファイルを作成します。

Log and Trace Analyzer は、ログ・ファイルとトレース・ファイルをフィルタリングして、アダプターのメッセージとトレース情報を分離することができます。また、ログ・ビューアーの中で、アダプターのメッセージとトレース情報を強調表示することもできます。

フィルタリングと強調表示の対象とするアダプターのコンポーネント ID は、文字 FTPRA とアダプター ID プロパティの値から構成されるストリングです。例えば、アダプター ID プロパティが、001 に設定されている場合、コンポーネント ID は、FTPRA001 となります。

同じアダプターを複数インスタンス実行する場合、アダプター ID プロパティの最初の 8 文字は、必ずインスタンスごとに固有にし、ログおよびトレース情報を特定のアダプター・インスタンスに相互に関連付けられるようにしてください。アダプター ID プロパティの最初の 7 文字を固有にすることにより、そのアダプターの複数インスタンスのコンポーネント ID も固有になり、アダプターの特定インスタンスにログおよびトレース情報を相互に関連付けることができます。アダプター ID プロパティの長さが、ログおよびトレース・ファイルのフィルター処理にどのように関係するかを示すため、2 つのインスタンスの WebSphere Adapter for FTP のアダプター ID プロパティを 001 と 002 に設定するとします。これらのインスタンスのコンポーネント ID、FTPRA001 および FTPRA002 は、短いので固有性を保つことができ、別のアダプター・インスタンスとして区別することができます。しかし、もっと長いアダプター ID プロパティのインスタンスの場合、互いを区別できなくなります。例えば、2 つのインスタンスのアダプター ID プロパティを、Instance01 および Instance02 に設定するとします。この場合、各アダプター・インスタンスのログおよびトレース情報を個別に調べることはできなくなります。これは、両方のインスタンスのコンポーネント ID が FTPRAInstance に切り詰められるためです。

Outbound 処理については、アダプター ID プロパティは、リソース・アダプターおよび Managed Connection Factory プロパティ・グループの両方にあります。外部サービス・ウィザードを使用して、Outbound 処理用にアダプターを構成後、アダプター ID プロパティを更新する場合、リソース・アダプター・プロパティと Managed Connection Factory プロパティを矛盾がないように必ず設定し、ログおよびトレース・エントリーのマーキングが不整合にならないようにしてください。Inbound 処理については、アダプター ID プロパティは、リソース・アダプター・プロパティのみに設定されますので、このような配慮は不要です。

アダプター ID プロパティについて詳しくは、次を参照してください。174 ページの『アダプター ID (AdapterID)』。Log and Trace Analyzer について詳しくは、<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/org.eclipse.hyades.log.ui.doc.user/concepts/cltviews.htm>を参照してください。

うるう年例外の取り扱い

Commons Net API v1.4.1 が提供するデフォルトのパースャーは、作成日が 2 月 29 日のファイル进行处理できません。com.ibm.j2ca.ftp.util.FTPLongListEntryParser クラスを使用すると、2 月 29 日に作成されたファイル进行处理できます。

ディレクトリーに 2 月 29 日に作成されたファイルが含まれる場合、com.ibm.j2ca.ftp.util.FTPLongListEntryParser クラスを使用します。このクラスは、ファイル名を取得し、エラーなしで処理を続行する基本的なインプリメンテーションです。長いリスト出力の構文解析を実行するために、インターフェースをインプリメントできます。

注: このクラスを使用してファイルを取り出すと、FTP サーバー上のファイルのタイム・スタンプは失われます。

ServerToServerFileTransfer

指定されたファイルは、1 つの FTP サーバー・ディレクトリーから別の FTP サーバー・ディレクトリーに転送される。

「ホスト名」プロパティの値が localhost に設定されている場合、1 次サーバーは、アダプター・ワークステーションと同じシステムにあります。Adapter for FTP は、「421 エラー - データ接続を開けません (421 error-Can't open data connection)」を生成します。この問題に対処するには、ホスト・ファイル (Windows プラットフォームの場合、<WindowsHome>/system32/drivers/etc/hosts にあります) を編集し、外部 IP アドレス (9.186.116.151 localhost など) を持つ新規エントリーを追加します。

ホスト名値または外部 IP アドレスが使用される場合 (例えばホスト名形式として FTPTEST が使用される場合や IP 形式として 9.186.116.151 が使用される場合) にも、アダプターは動作します。

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、ほかのログ・ファイルおよびトレース・ファイルと区別します。

このタスクを実行する理由および時期

ロギング・プロパティの構成

管理コンソールを使用して、ロギングを使用可能にして、ログの出力プロパティ (ログの場所、詳細レベル、および出力フォーマットなど) を設定します。

このタスクを実行する理由および時期

アダプターでモニター対象イベントをログに記録できるようにするには、モニターしたいサービス・コンポーネントのイベント・ポイント、イベントごとに必要となる詳細レベル、およびイベントをログにパブリッシュするのに使用する出力のフォーマットを指定する必要があります。管理コンソールを使用して、次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設定する
- ログの詳細レベルを指定する
- ログ・ファイルの保管場所および保有数を指定する
- ログ出力のフォーマットを指定する

ログ・アナライザー出力のフォーマットを設定する場合、ログ・アナライザー・ツールを使用してトレース出力を開くことができます。ログ・アナライザー・ツールは、プロセス・サーバーに組み込まれたアプリケーションです。このツールは、ログ・アナライザーのマージ機能を使用できるため、異なる 2 つのサーバー・プロセスのトレースを相関させる場合に便利です。

サービス・コンポーネントやイベント・ポイントなどのプロセス・サーバー 上でのモニターについて詳しくは、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。アプリケーション・サーバーを開始または再始動すると、静的構成が有効になります。動的構成変更 (またはランタイム構成変更) は、即時適用されます。

ログが作成されると、そのログの詳細レベルが構成データから設定されます。特定のログ名に対して、構成データが使用可能でない場合、そのログのレベルは、ログの親から取得されます。親ログに構成データが存在しない場合、そのログの親が確認される、という具合に、ヌル以外のレベル値があるログが見つかるまでツリーを上昇します。ログのレベルを変更すると、その変更はログの子に伝搬されます。また、必要に応じて、ログの子からその子へと変更が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、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、「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. 作業したい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」で「ログおよびトレース」をクリックします。
4. 「ログ詳細レベルの変更」をクリックします。
5. 変更を有効にするには、以下を行います。
 - 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、「ランタイム」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 用のパッケージ名は、**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とすべてのデプロイ済みアダプターの場合は、**com.ibm.j2ca.*** 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. ログイン・レベルを選択します。

ログイン・レベル (Logging Level)	説明
致命的	タスクを続行できない。または、コンポーネントが機能しない。
重大	タスクを続行できないが、コンポーネントは機能する。このログイン・レベルには、差し迫った致命的エラーを示す (すなわち、リソースが枯渇寸前であることを強く示唆する) 状況も含まれる。
警告	潜在的なエラーが発生したか、重大エラーが差し迫っている。このログイン・レベルには、例えばリソース・リークの可能性など、進行性のフォールトを示す状況も含まれる。
監査	サーバーの状態やリソースに影響を与える重大なイベントが発生した。
情報	タスクが稼働中である。このログイン・レベルには、タスクの全体的な進行を概説する一般情報が含まれる。
構成	構成の状況が報告されるか、構成変更が発生した。
詳細	サブタスクが稼働中である。このログイン・レベルには、サブタスクの進行を詳細に説明した一般情報が含まれる。

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・サーバーを開始または再始動すると、静的変更が有効になります。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、以下の手順を使用します。

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) 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実行され、実行時に発生するイベントやエラーを収集します。この機能はさまざまな障害を相互に関連付ける手段を提供するため、この機能を利用すると、ソフトウェアは、ある 1 つの障害の影響をその原因に結びつけ、その結果、障害の根本原因を素早く突き止めることが容易になります。取り込まれたデータは、アダプターの実行時に発生した例外処理を識別するときに使用できます。

問題が発生すると、例外メッセージおよびコンテキスト・データがアダプターによってログ・ファイルに書き込まれます。このログ・ファイルは `install_root/profiles/profile/logs/ffdc` ディレクトリーに置かれます。

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

アダプターは、予想される例外で **Outbound** サービス記述で宣言されている例外であるビジネス・フォールトか、インポートをサポートします。ビジネス・フォールトは、ビジネス・ルールの違反または制約違反が原因で、ビジネス・プロセスの予測可能なポイントに発生します。

WebSphere Process Server と WebSphere Enterprise Service Bus は、他のタイプのフォールトをサポートしますが、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、これらを単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、**Outbound** 操作に応答タイプが構成されている場合にのみ使用されます。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、アダプターで必要なデータが含まれていない **Outbound** 処理のビジネス・オブジェクトを受け取るか、あるいはアダプターで **Outbound** 処理中に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。

注: 特定の操作のフォールトは、その操作に応答が構成されている場合にのみ有効になります。

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、アダプターが生成できる各フォールト用のビジネス・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`、`primarySetKey` の各属性など) が含まれています。159 ページの図 83 を参照してください。

WBIFault	
message	string
errorCode	string
primaryKeySet	PrimaryKeyPairType []

図 83. WBIFault ビジネス・オブジェクトの構造

WebSphere Adapter for FTP はフォールトを使用可能にします。フォールトを手動で構成する必要はありません。アダプターは、ウィザードが作成する次のフォールト・ビジネス・オブジェクトを提供します。

- 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。このフォールトは、ServerToServerFileTransfer 操作時、および、2 次サーバー・ディレクトリーにファイルが既に存在する場合にスローされます。

- RecordNotFoundFault

Create、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。このフォールトは、ディレクトリー・パスが存在しないとき、および、Create 操作時にシーケンス・ファイルが存在しないときに発生します。

- 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。Splittingfunctionclassname が SplitByDelimiter で、LocalDirectoryPath が NULL の場合に、splitCriteria が NULL か無効であると、LocalDirectoryPath が欠落しているというメッセージとともに、MissingData フォールトがスローされます。splitCriteria が NULL または無効で、SplitBySize が構成されている場合、アダプターは例外をスローしません。Retrieve 操作中に、区切り文字が NULL で SplitByDelimiter が構成されているために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。

XAResourceNotAvailableException

プロセス・サーバー・ログに

com.ibm.ws.Transaction.XAResourceNotAvailableException 例外のレポートが繰り返し出力されている場合、トランザクション・ログを除去して問題を訂正してください。

症状:

アダプターを始動すると、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。アダプターを始動すると、アダプターはトランザクションをリカバリーしようとはしますが、リソースが除去されているためできません。

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. トランザクションを含むトランザクション・ログ・ファイルを除去します。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、サーバーはこれらのトランザクションのリカバリーを試行しなくなります。

注: 通常、テスト環境または開発環境では、すべてのトランザクション・ログを削除できます。WebSphere Integration Developer で、トランザクション・ログ・ディレクトリー `server_install_directory\profiles\profile_name\tranlog` のファイルおよびサブディレクトリーを削除します。

実稼働環境では、処理が不要なイベントを表すトランザクションのみを削除します。この処理を行う方法の 1 つとしては、アダプターを再インストールし、使用されている元のイベント・データベースにアダプターを位置指定して、不要なトランザクションのみを削除する方法があります。別の方法として、以下のディレクトリーにある `log1` または `log2` ファイルのどちらかから、トランザクションを削除します。

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

3. プロセス・サーバー を開始します。

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は、最新のサポート情報やテクニカル文書を手入したり、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ために使用することができます。また、セルフ・ヘルプ・リソースは、アダプターに関連する問題を診断するのに役立ち、IBM ソフトウェア・サポートへの連絡方法についての情報を提供します。

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[®]、およびホワイト・ペーパーなどの技術情報
- 研修関連
- テクニカル・ノート

推奨フィックス

適用することが望ましい推奨フィックスのリストは、<http://www.ibm.com/support/docview.wss?fdoc=aimadp&rs=695&uid=swg27010397> にあります。

テクニカル・ノート

テクニカル・ノートは、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 のテクニカル・ノートのリストについては、以下のアドレスにアクセスしてください。

[http://www.ibm.com/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 のプラグインを提供します。これは、無料の保守容易性ローカル・ソフトウェア・ワークベンチです。プラグインは、動的トレース・フィーチャーをサポートします。IBM Support Assistant のインストールおよび使用については、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 9 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、ユーザー定義タイプ。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。

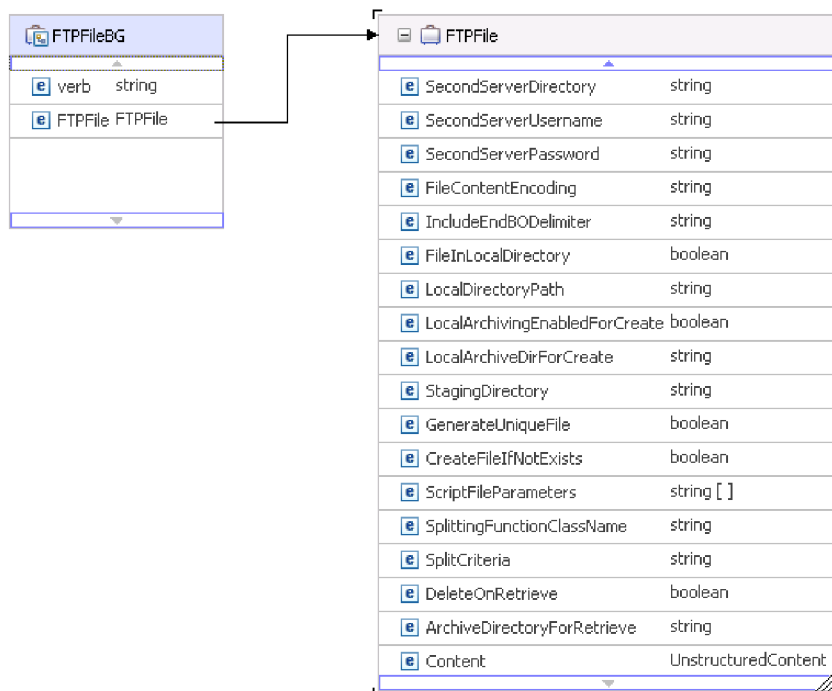


図 84. FTPFileBG ビジネス・オブジェクト

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。

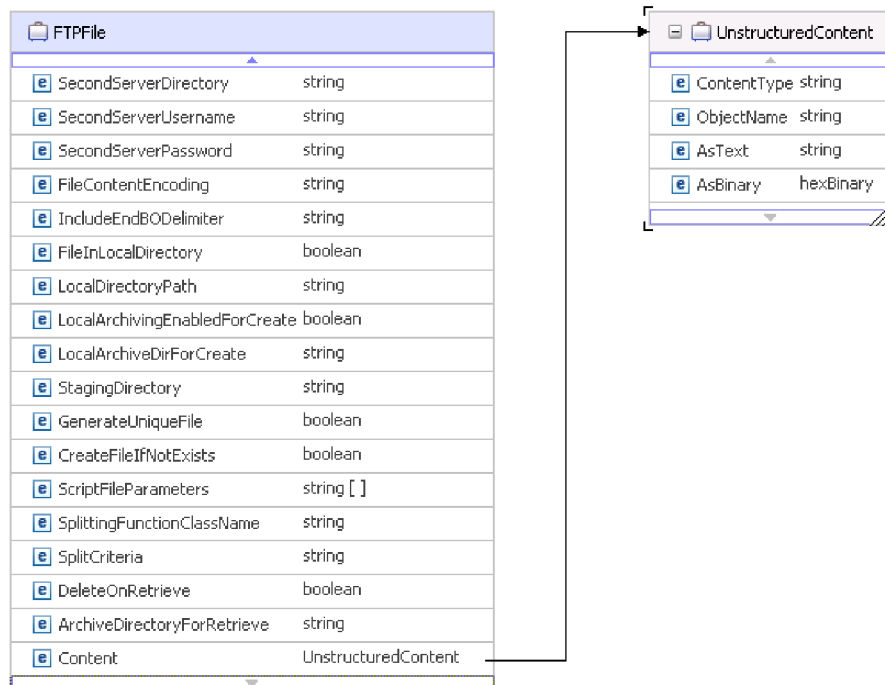


図 85. FTPFile ビジネス・オブジェクト

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。

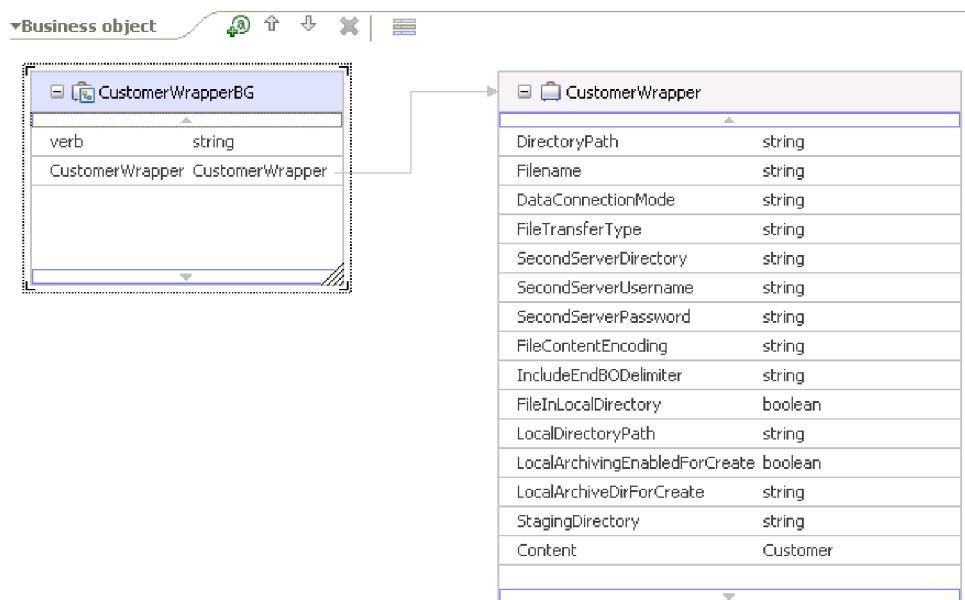


図 86. CustomerWrapperBG ビジネス・オブジェクト

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。

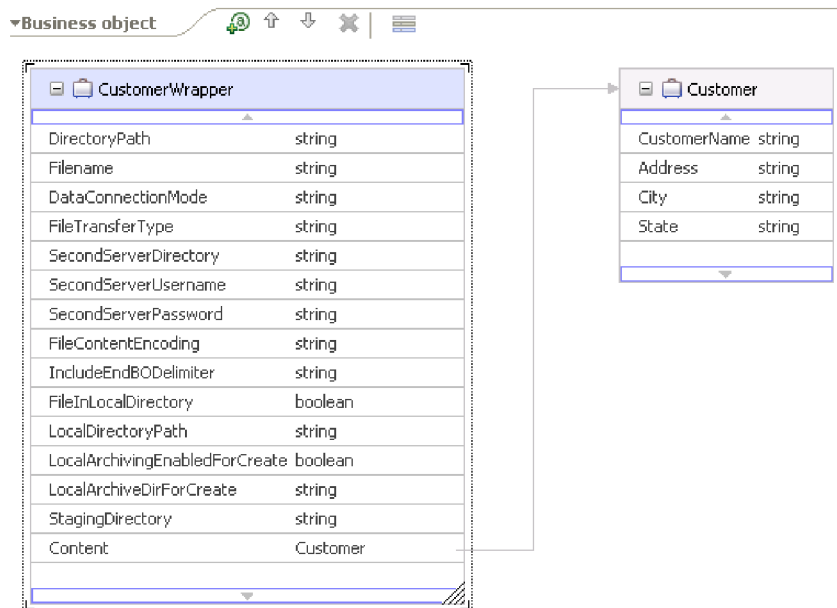


図 87. CustomerWrapper ビジネス・オブジェクト

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

エ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

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と、アダプターがプロパティを解釈する方法を示します。

表 8. ビジネス・オブジェクト属性プロパティ

プロパティ	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 9. 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。
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。

表9. 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
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詳しくは、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、オブジェクトやサービスを生成したり作成したりするとき、外部サービス・ウィザードを使用して設定する、いくつかの種類の Outbound 接続構成プロパティーがあります。WebSphere Integration Developer または 管理コンソールを使ってモジュールを WebSphere Process Server または WebSphere Enterprise Service Bus 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。

プロパティーの詳細についてのガイド

WebSphere Adapter for FTP を構成するときに使用されるプロパティーは、リソース・アダプター・プロパティーや Managed Connection Factory プロパティーなど、それぞれの構成プロパティーのトピックに記載されている表で詳細に説明されています。これらの表を使用しやすくするため、参照する各行の情報を以下に説明します。

次の表では、構成プロパティの表に表示される場合がある各行の意味を説明します。

行	説明
必須	<p>アダプターが動作するためには、必須フィールド (プロパティ) に値が必要です。必須プロパティに対しては、外部サービス・ウィザードがデフォルト値を提供する場合があります。</p> <p>外部サービス・ウィザードの必須フィールドからデフォルト値を除去しても、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入っていないと、外部サービス・ウィザードは、その割り当て済みのデフォルト値を使用してフィールドを処理し、そ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>可能な値は「はい」および「いいえ」です。</p> <p>プロパティは、他のプロパティが特定の値の場合のみ必須となることがあります。その場合は、表にこの依存関係が記載されます。以下に例を示します。</p> <ul style="list-style-type: none"> • EventQueryType プロパティが Dynamic に設定された場合は「はい」 • Oracle データベースの場合は「はい」
使用可能な値	<p>プロパティで選択可能な値をリストして説明します。</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る事前定義値。プロパティが必須の場合は、デフォルト値を受け入れるか、ユーザーが値を指定する必要があります。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>None という語は、受け入れ可能なデフォルト値です。デフォルト値がないという意味ではありません。</p>
計測単位	<p>プロパティの計測単位を指定します (例: キロバイト、秒)。</p>
プロパティ・タイプ	<p>プロパティ・タイプを示します。有効なプロパティ・タイプとしては、以下のものがあります。</p> <ul style="list-style-type: none"> • Boolean • String • Integer
使用法	<p>プロパティに適用される場合がある使用の条件または制限について記述します。制限の記載例を以下に示します。</p> <p>Rational® Application Developer for WebSphere Software バージョン 6.40 またはそれ以前では、パスワードに以下の制限があります。</p> <ul style="list-style-type: none"> • 大文字である必要があります • 長さが 8 文字である必要があります <p>Rational Application Developer for WebSphere Software バージョン 6.40 よりも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"> • 大文字小文字を区別しません • 長さが 40 文字まで可能です <p>このセクションでは、このプロパティに影響を及ぼす他のプロパティ、またはこのプロパティによって影響を受ける他のプロパティをリストし、その条件付き関係の内容を説明します。</p>

行	説明
例	プロパティ値のサンプルを示します。例: 「言語が JA (日本語) に設定された場合、コード・ページ番号は 8000 に設定され ます。」
グローバル化	グローバル化されたプロパティには各国語サポートが備わっているので、値を各国 語で設定できます。 有効な値は「はい」および「いいえ」です。
BIDI 対応	プロパティが双方向 (bidi) 処理でサポートされるかどうかを示します。双方向処理 とは、1 つのファイルに左から右 (ヘブライ語やアラビア語など) と右から左 (URL やファイル・パスなど) の両方の意味内容を含むデータを処理するタスクを指しま 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

アダプター・タイプ・プロパティは、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・プロパティのい
ずれかを設定する場合、そのプロパティ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・プロパティおよびその目的を以下の表に示します。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。後続セクションのプ
ロパティ詳細表の見方について詳しくは、169 ページの『プロパティの詳細に
ついてのガイド』を参照してください。

表 10. アダプター・タイプ・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	
172 ページの 『「説明」プロパ ティ (Description)』	Description	アダプターの説明。
172 ページの 『「表示名」プロ パティ (DisplayName)』	DisplayName	アダプター表示名。
172 ページの 『「ID」プロパテ ィー (ID)』	ID	アダプター・タイプの ID。
172 ページの 『「ベンダー」プ ロパティ (Vendor)』	Vendor	アダプターを提供するベンダーの名前。

表 10. 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 11. 「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 12. 「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 13. 「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 14. 「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 15. 「バージョン」プロパティ特性

必須	はい
----	----

表 15. 「バージョン」プロパティ特性 (続き)

デフォルト	6.2
プロパティ・タイプ	String

リソース・アダプター・プロパティ

リソース・アダプター・プロパティは、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は、バージョン 6.1.x 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。

- LogFileSize
- LogFileName
- LogNumberOfFiles
- TraceFileSize
- TraceFileName
- TraceNumberOfFiles

以下の表は、リソース・アダプター・プロパティおよびその目的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 16. Adapter for FTP 用のリソース・アダプター・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	
アダプター ID	AdapterID	PMI イベントのアダプター・インスタンス、ロギングおよびトレースのアダプター・インスタンスを識別する場合に使用します。
175 ページの『EISEncoding (EISEncoding)』	EISEncoding	FTP サーバーのエンコード。
(なし)	enableHASupport	このプロパティは変更しないでください。
175 ページの『ログ・ファイルおよびトレース・ファイルで、ユーザー・データを「XXX」と記述する (HideConfidentialTrace) プロパティ』	HideConfidentialTrace	ログおよびトレース・ファイルへの書き込み時に、ユーザー・データを「X」のストリングで記述し、潜在的な機密データを隠すようにするかどうかを指定します。
(なし)	ログ・ファイル最大サイズ	非推奨
(なし)	LogFilename	非推奨
(なし)	ログ・ファイル数	非推奨

表 16. Adapter for FTP 用のリソース・アダプター・プロパティー (続き)

プロパティー名		説明
ウィザード内	管理コンソール内	
(なし)	トレース・ファイル最大サイズ	非推奨
(なし)	トレース・ファイル名	非推奨
(なし)	トレース・ファイル数	非推奨

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) を識別します。

表 17. 「アダプター ID」の詳細

必須	はい
デフォルト	001
プロパティー・タイプ	String
使用法	<p>このプロパティーは、ログおよびトレース・ファイル内のアダプター・インスタンスを識別し、また、アダプターのモニター時にアダプター・インスタンスを識別する場合に役立ちます。アダプター ID は、アダプター固有の ID、FTPRA と共に使用され、Log and Trace Analyzer ツールによって使用されるコンポーネント名を構成します。例えば、アダプター ID プロパティーが、001 に設定されている場合、コンポーネント ID は、FTPRA001 となります。</p> <p>同じアダプターを複数インスタンス実行する場合、アダプター ID プロパティーの最初の 8 文字は、必ずインスタンスごとに固有にし、ログおよびトレース情報を特定のアダプター・インスタンスに相互に関連付けられるようにしてください。アダプター ID プロパティーの最初の 7 文字を固有にすることにより、そのアダプターの複数インスタンスのコンポーネント ID も固有になり、アダプターの特定インスタンスにログおよびトレース情報を相互に関連付けることができます。</p> <p>アダプター ID プロパティーの長さが、ログおよびトレース・ファイルのフィルター処理にどのように関係するかを示すため、2 つのインスタンスの WebSphere Adapter for FTP のアダプター ID プロパティーを 001 と 002 に設定するとします。これらのインスタンスのコンポーネント ID、FTPRA001 および FTPRA002 は、短いので固有性を保つことができ、別のアダプター・インスタンスとして区別することができます。しかし、もっと長いアダプター ID プロパティーのインスタンスの場合、互いを区別できなくなります。例えば、2 つのインスタンスのアダプター ID プロパティーを、Instance01 および Instance02 に設定するとします。この場合、各アダプター・インスタンスのログおよびトレース情報を個別に調べることはできなくなります。これは、両方のインスタンスのコンポーネント ID が FTPRAInstance に切り詰められるためです。</p> <p>Inbound 処理の場合、このプロパティーの値は、リソース・アダプター・レベルで設定されます。Outbound 処理の場合は、リソース・アダプター・レベルと Managed Connection Factory レベルの両方で設定できます。外部サービス・ウィザードを使用して、アダプターを Outbound 処理用に構成した後、リソース・アダプター・プロパティーおよび Managed Connection Factory プロパティーを個別に設定できます。WebSphere Integration Developer アセンブリー・エディターまたは管理コンソールを使用して、これらのプロパティーを再設定する場合、ログおよびトレース・エントリーのマーキングが不整合にならないように、矛盾がない設定になっていることを確認してください。</p>
グローバル化	はい

表 17. 「アダプター ID」の詳細 (続き)

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グローバル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 18. 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 (Enable high availability support) (enableHASupport)

このプロパティーは変更しないでください。true に設定してください。

ログ・ファイルおよびトレース・ファイルで、ユーザー・データを「XXX」と記述する (HideConfidentialTrace) プロパティー

このプロパティーは、ログおよびトレース・ファイル中のユーザー・データを「X」のストリングに置換し、潜在的な機密データが許可なく外部に漏れないようにします。

表 19. ログ・ファイルおよびトレース・ファイルで、ユーザー・データを「XXX」と記述するの詳細

必須	いいえ
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	このプロパティーを True に設定すると、アダプターでは、ログおよびトレース・ファイルに書き込む時に、ユーザー・データを「X」のストリングに置換します。 Inbound 処理の場合、このプロパティーの値は、リソース・アダプター・レベルで設定されます。Outbound 処理の場合は、リソース・アダプター・レベルと Managed Connection Factory レベルの両方で設定できます。外部サービス・ウィザード を使用して、アダプターを Outbound 処理用に構成した後、リソース・アダプター・プロパティーおよび Managed Connection Factory プロパティーを個別に設定できます。WebSphere Integration Developer アセンブリー・エディターまたは管理コンソールを使用して、これらのプロパティーを再設定する場合、ログおよびトレース・エントリーのマーキングが不整合にならないように、矛盾がない設定になっていることを確認してください。
グローバル化	いいえ
BIDI 対応	いいえ

ログ・ファイルの最大サイズ (Log file maximum size) (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 20. 「ログ・ファイルの最大サイズ (Log file maximum size)」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 21. 「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 (Log number of files) (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 22. 「ログ・ファイルの数 (Log number of files)」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・ファイルがその最大サイズに到達すると、アダプターは別のログ・ファイルの使用を開始します。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 (Trace file maximum size) (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 23. 「トレース・ファイルの最大サイズ (Trace file maximum size)」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 24. 「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 (Trace number of files) (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・ファイルがその最大サイズに到達すると、アダプターは別のトレース・ファイルの使用を開始します。

表 25. 「トレース・ファイルの数 (Trace number of files)」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

Managed (J2C) Connection Factory プロパティ

Managed Connection Factory 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。

Managed Connection Factory 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、Managed Connection Factory 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 Managed Connection Factory 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 26. Managed Connection Factory 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	
アダプター ID	AdapterID	PMI イベントのアダプター・インスタンス、ロギングおよびトレースのアダプター・インスタンスを識別する場合に使用します。
181 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。
181 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 操作時に使用するファイル名を指定します。
182 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。
182 ページの『ログ・ファイルおよびトレース・ファイルで、ユーザー・データを「XXX」と記述する (HideConfidentialTrace) プロパティ』	HideConfidentialTrace	ログおよびトレース・ファイルへの書き込み時に、ユーザー・データを「X」のストリングで記述し、潜在的な機密データを隠すようにするかどうかを指定します。
サーバー検査を使用可能にする	EnableServerVerification	SFTP プロトコルのリモート・サーバー検査を使用可能にする
183 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。
ホスト鍵ファイル	HostKeyFile	トラステッド・サーバーのホスト鍵を含んだホスト鍵ファイルの絶対パス

表 26. Managed Connection Factory プロパティ (続き)

184 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。
184 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を指定します。
185 ページの『「ホスト名」プロパティ (SocksProxyHost)』	SocksProxyHost	プロキシー・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。
パスフレーズ・プロパティ (passPhrase)	passPhrase	秘密鍵を暗号化して、拡張セキュリティのために使用されます
185 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。
186 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。
186 ページの『「パスワード」プロパティ (SocksProxyPassword)』	SocksProxyPassword	プロキシー・サーバーを認証するためのパスワードを指定します。
186 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。
186 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。
186 ページの『「ポート番号」プロパティ (SocksProxyPort)』	SocksProxyPort	プロキシー・サーバーのポート番号を指定します。
187 ページの『「秘密鍵ファイル」プロパティ (PrivateKeyFilePath)』	PrivateKeyFilePath	セキュア・シェル・サーバーに認証させるために使用される秘密鍵。
187 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。
187 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。
188 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。
188 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。
189 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。

表 26. Managed Connection Factory プロパティ (続き)

189 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。
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に使用するユーザー名を指定します。
190 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。

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) を識別します。

表 27. 「アダプター ID」の詳細

必須	はい
デフォルト	001
プロパティ・タイプ	String

表 27. 「アダプター ID」の詳細 (続き)

<p>使用法</p>	<p>このプロパティは、ログおよびトレース・ファイル内のアダプター・インスタンスを識別し、また、アダプターのモニター時にアダプター・インスタンスを識別する場合に役立ちます。アダプター ID は、アダプター固有の ID、FTPRA と共に使用され、Log and Trace Analyzer ツールによって使用されるコンポーネント名を構成します。例えば、アダプター ID プロパティが、001 に設定されている場合、コンポーネント ID は、FTPRA001 となります。</p> <p>同じアダプターを複数インスタンス実行する場合、アダプター ID プロパティの最初の 8 文字は、必ずインスタンスごとに固有にし、ログおよびトレース情報を特定のアダプター・インスタンスに相互に関連付けられるようにしてください。アダプター ID プロパティの最初の 7 文字を固有にすることにより、そのアダプターの複数インスタンスのコンポーネント ID も固有になり、アダプターの特定インスタンスにログおよびトレース情報を相互に関連付けることができます。</p> <p>アダプター ID プロパティの長さが、ログおよびトレース・ファイルのフィルター処理にどのように関係するかを示すため、2 つのインスタンスの WebSphere Adapter for FTP のアダプター ID プロパティを 001 と 002 に設定するとします。これらのインスタンスのコンポーネント ID、FTPRA001 および FTPRA002 は、短いので固有性を保つことができ、別のアダプター・インスタンスとして区別することができます。しかし、もっと長いアダプター ID プロパティのインスタンスの場合、互いを区別できなくなります。例えば、2 つのインスタンスのアダプター ID プロパティを、Instance01 および Instance02 に設定するとします。この場合、各アダプター・インスタンスのログおよびトレース情報を個別に調べることはできなくなります。これは、両方のインスタンスのコンポーネント ID が FTPRAInstance に切り詰められるためです。</p> <p>Inbound 処理の場合、このプロパティの値は、リソース・アダプター・レベルで設定されます。Outbound 処理の場合は、リソース・アダプター・レベルと Managed Connection Factory レベルの両方で設定できます。外部サービス・ウィザードを使用して、アダプターを Outbound 処理用に構成した後、リソース・アダプター・プロパティおよび Managed Connection Factory プロパティを個別に設定できます。WebSphere Integration Developer アセンブリー・エディターまたは管理コンソールを使用して、これらのプロパティを再設定する場合、ログおよびトレース・エントリーのマーキングが不整合にならないように、矛盾がない設定になっていることを確認してください。</p>
<p>グローバル化</p>	<p>はい</p>
<p>BIDI 対応</p>	<p>いいえ</p>

「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 28. 「カスタム・パーサー・クラス名」プロパティ特性

<p>必須</p>	<p>いいえ</p>
<p>デフォルト</p>	<p>なし</p>
<p>プロパティ・タイプ</p>	<p>String</p>
<p>グローバル化されるかどうか</p>	<p>いいえ</p>

「デフォルト・ターゲット・ファイル名」プロパティ (Filename)

Outbound 操作時に使用するファイル名。

表 29. 「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
使用法	WebSphere Application Server 環境変数を使用して、ファイル名ディレクトリーを表すことができます。\$ 記号の後に、環境変数の名前を中括弧で囲んで指定します。例えば、\${FILENAME} です。この資料の 71 ページの『WebSphere Application Server 環境変数の定義』についてのトピックを参照してください。
グローバル化されるかどうか	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 30. 「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
使用法	WebSphere Application Server 環境変数を使用して、出力ディレクトリーを表すことができます。\$ 記号の後に、環境変数の名前を中括弧で囲んで指定します。例えば、\${OUTPUT_DIRECTORY} です。この資料の 71 ページの『WebSphere Application Server 環境変数の定義』についてのトピックを参照してください。
グローバル化されるかどうか	はい

ログ・ファイルおよびトレース・ファイルで、ユーザー・データを「XXX」と記述する (HideConfidentialTrace) プロパティ

このプロパティは、ログおよびトレース・ファイル中のユーザー・データを「X」のストリングに置換し、潜在的な機密データが許可なく外部に漏れないようにします。

表 31. ログ・ファイルおよびトレース・ファイルで、ユーザー・データを「XXX」と記述するの詳細

必須	いいえ
使用可能な値	True False
デフォルト	False
プロパティ・タイプ	Boolean

表 31. ログ・ファイルおよびトレース・ファイルで、ユーザー・データを「XXX」と記述するの詳細 (続き)

使用法	このプロパティを True に設定すると、アダプターでは、ログおよびトレース・ファイルに書き込む時に、ユーザー・データを「X」のストリングに置換します。 Inbound 処理の場合、このプロパティの値は、リソース・アダプター・レベルで設定されます。Outbound 処理の場合は、リソース・アダプター・レベルと Managed Connection Factory レベルの両方で設定できます。外部サービス・ウィザード を使用して、アダプターを Outbound 処理用に構成した後、リソース・アダプター・プロパティおよび Managed Connection Factory プロパティを個別に設定できます。WebSphere Integration Developer アセンブリー・エディターまたは管理コンソールを使用して、これらのプロパティを再設定する場合、ログおよびトレース・エントリーのマーキングが不整合にならないように、矛盾がない設定になっていることを確認してください。
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 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 32. 「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
グローバル化されるかどうか	いいえ

サーバー検査を使用可能にする プロパティ (EnableServerVerification)

このプロパティは、SFTP プロトコルのリモート・サーバー検査を使用可能にするために使用されます。

表 33. サーバー検査を使用可能にする プロパティの詳細

必須	いいえ
使用可能な値	True False
デフォルト	False
プロパティ・タイプ	Boolean
使用法	このプロパティの設定内容によって、次のようになります。 <ul style="list-style-type: none"> • True に設定されると、サーバー認証は使用可能です。 • False に設定されると、サーバー認証は使用不可です。 アダプターは、トラステッド・サーバーのホスト鍵を含んだファイルのパスの中の、HostKeyFile プロパティを検査します。
グローバル化	はい
BIDI 対応	いいえ

ホスト鍵ファイル プロパティ (HostKeyFile)

このプロパティは、トラステッド・サーバーのホスト鍵を含むホスト鍵ファイルの絶対パスを指定します。

表 34. ホスト鍵ファイル プロパティの特性

必須	EnableServerVerification が使用可能の場合、このプロパティを指定する必要があります。
デフォルト	なし
プロパティ・タイプ	String
使用法	これは、このファイルに指定されているトラステッド・サーバーのホスト鍵でリモート・サーバーのホスト鍵を検査するために、アダプターが使用します。
グローバル化されるかどうか	はい
BIDI 対応	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 35. 「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
グローバル化されるかどうか	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 36. 「ホスト名」プロパティ特性

必須	はい
----	----

表 36. 「ホスト名」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
使用法	FTP サーバーのホスト名または IP アドレス (例えば 9.20.13.159) が入ります
グローバル化されるかどうか	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 37. 「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
グローバル化されるかどうか	はい

パスフレーズ・プロパティ (passPhrase) (passPhrase)

このプロパティは、秘密鍵を暗号化して、拡張セキュリティのために使用されます。

表 38. パスフレーズ・プロパティ (passPhrase) プロパティの特性

必須	いいえ
使用可能な値	任意のストリング値
デフォルト	なし
プロパティ・タイプ	String
使用法	拡張セキュリティに使用されます。これは、SFTP 構成の中で秘密鍵を暗号化することによって、秘密鍵を保護します。
グローバル化されるかどうか	はい
BIDI 対応	いいえ

「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。

表 39. 「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
グローバル化されるかどうか	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 40. 「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
グローバル化されるかどうか	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 41. 「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
グローバル化されるかどうか	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 42. 「ポート番号」プロパティ特性

必須	はい
デフォルト	FTP は 21、FTPS は 990、SFTP は 22。
プロパティ・タイプ	Integer
グローバル化されるかどうか	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 43. 「ポート番号」プロパティ特性

必須	はい
デフォルト	FTP は 21、FTPS は 990。
プロパティ・タイプ	Integer
グローバル化されるかどうか	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 44. 「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
グローバル化されるかどうか	いいえ

秘密鍵ファイル・プロパティ (PrivateKeyFilePath)

このプロパティによって、秘密鍵を参照して選択することができます。その秘密鍵は、セキュア・シェル・サーバーに対する認証に使用されます。

表 45. 「秘密鍵」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
使用法	秘密鍵を格納するファイルの絶対パス。ユーザーをセキュア・シェル・サーバーに認証させるために使用されます。
例	c:%temp%key.ppk
グローバル化されるかどうか	はい
BIDI 対応	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

FTP over SSL 接続: FTPS

FTP over SSH 接続: SFTP

表 46. 「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
グローバル化されるかどうか	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

FTP over SSL 接続: FTPS

表 47. 「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
グローバル化されるかどうか	いいえ

「2 次サーバー・ディレクトリー」プロパティ (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。これは、ファイルの転送先のリモート・イベント・ディレクトリーです。

表 48. 「2 次サーバー・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
使用法	FTP サーバーにあり、Outbound 操作で使用されるディレクトリーは、FTP ディレクトリーの絶対パスを表します。これにはホスト名または URL 情報は含まれません。例えば、/home/usr/output です。
グローバル化されるかどうか	はい

「シーケンス・ファイル」プロパティ (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 49. 「シーケンス・ファイル」プロパティ特性

必須	いいえ
デフォルト	なし

表 49. 「シーケンス・ファイル」プロパティ特性 (続き)

プロパティ・タイプ	String
使用法	重要: 同時に要求が出されるとバッチ要求の処理で遅延の原因となるため、2 つのアダプター・インスタンスがクラスターの一部でない限り、2 つのアダプター・インスタンスを同じシーケンス・ファイルにアクセスすることはお勧めできません。
グローバル化されるかどうか	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 50. 「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
使用法	WebSphere Application Server 環境変数を使用して、ステージング・ディレクトリーを表すことができます。\$ 記号の後に、環境変数の名前を中括弧で囲んで指定します。例えば、\${STAGING_DIRECTORY} です。この資料の 71 ページの『WebSphere Application Server 環境変数の定義』についてのトピックを参照してください。
グローバル化されるかどうか	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 51. 「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
グローバル化されるかどうか	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 52. 「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
グローバル化されるかどうか	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 53. 「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
グローバル化されるかどうか	はい

Wrapper および対話仕様プロパティ

Wrapper プロパティは Wrapper ビジネス・オブジェクトの属性で、Wrapper ビジネス・オブジェクトによって、アプリケーション・プログラマーはラッパーの中のビジネス・オブジェクトの操作を制御できます。対話仕様プロパティは、全体のアダプターの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。ただし、Outbound 操作の一部のプロパティは変更できます。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。Wrapper プロパティは、WebSphere Integration Developer テスト・クライアントを使用して設定するか、実行時にプログラムで設定します。

以下の表は、Wrapper および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 54. 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	
Retrieve 操作のリモート・アーカイブ・ディレクトリー	ArchiveDirectoryForRetrieve	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。

表 54. 対話仕様プロパティ (続き)

ファイルが存在しない場合に新規ファイルを作成する	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。
FTP サーバー接続モード	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。
retrieve 操作後のファイルの削除	DeleteOnRetrieve	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。
FTP システム上のリモート・ディレクトリー	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。
ファイル内容のエンコード	FileContentEncoding	ファイルへの書き込み中に使用されるエンコード。
ローカル・ディレクトリー内のファイル	FileInLocalDirectory	Create 操作時に true に設定されている場合、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・パスからファイル内容が取り出される。
デフォルト・ターゲット・ファイル名	Filename	DirectoryPath プロパティで指定されたディレクトリー内のファイルの名前。
ファイル転送タイプ	FileTransferType	Outbound 操作時に使用されるファイル転送タイプ。
固有ファイルの生成	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。
ホスト名プロパティ	SecondServerHostName	2 次 FTP サーバーのホスト名。
「ファイル内のビジネス・オブジェクト間の区切り文字」プロパティ	IncludeEndBODelimiter	この値がファイルの内容に付加される。
Create 操作のローカル・アーカイブ・ディレクトリー	LocalArchiveDirForCreate	Create 操作時に LocalArchivingEnabledForCreate が true に設定されている場合、このディレクトリーのローカル・ワークステーションにファイルが保存される。
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	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。
ローカル・ディレクトリー	LocalDirectoryPath	このディレクトリーからファイルが取り出される。
ポート番号	SecondServerPortNumber	2 次 FTP サーバーのポート番号。
プロトコル	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。
スクリプト・ファイル・パラメーター	ScriptFileParameters	FTP スクリプト・ファイルに必要なパラメーター。
ディレクトリー	SecondServerDirectory	ServerToServerFileTransfer 操作時の 2 次 FTP サーバーのディレクトリー・パス。
パスワード	SecondServerPassword	ServerToServerFileTransfer 操作時の 2 次 FTP サーバーのパスワード。

表 54. 対話仕様プロパティ (続き)

ユーザー名	SecondServerUsername	ServerToServerFileTransfer 操作時の 2 次 FTP サーバーのユーザー名。
ファイル内容を分割するための基準の指定	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。
関数クラス名の分割	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。
ステージング・ディレクトリー	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 55. 「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
グローバル化されるかどうか	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 56. 「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
グローバル化されるかどうか	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。これは、ファイルの転送先のリモート・イベント・ディレクトリーです。

表 57. 「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
使用法	対話仕様プロパティーの場合、FTP サーバーにあって Outbound 操作で使用されるディレクトリーは、FTP ディレクトリーの絶対パスを表します。例えば、/home/usr/output です。これにはホスト名または URL 情報は含まれません。 wrapper ビジネス・オブジェクト・プロパティーの場合、Outbound 操作 ServerToServerFileTransfer の実行対象となる 2 次サーバーの URL。例えば、FTP URL を指定する場合の構文は ftp://[UserId:password@]FTPserver[:port]/DirectoryForSecondServer です。
グローバル化されるかどうか	はい

「FTP サーバー接続モード」プロパティー (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。このプロパティーは、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 58. 「FTP サーバー接続モード」プロパティー特性

必須	いいえ
デフォルト	active
プロパティー・タイプ	String
使用可能な値	active または passive
グローバル化されるかどうか	いいえ

retrieve 操作後のファイルの削除 (DeleteOnRetrieve)

Outbound Retrieve 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、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 59. 「retrieve 操作後のファイルの削除」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
グローバル化されるかどうか	いいえ

「FTP システム上のリモート・ディレクトリー」プロパティー (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

のローカル・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 60. 「FTP システム上のリモート・ディレクトリー」プロパティー特性

必須	はい
デフォルト	なし
プロパティー・タイプ	String
使用法	アダプターを始動する前に、アダプターを実行するマシン上に、DirectoryPath ディレクトリーを手動で作成する必要があります。アダプターはこのディレクトリーを自動的に作成しないためです。
グローバル化されるかどうか	はい

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

ファイルへの書き込み中に使用されるエンコード。このプロパティーを指定しない場合、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 61. 「ファイル内容のエンコード」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
グローバル化されるかどうか	いいえ

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

Outbound の create 操作時に、このプロパティーが 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、このプロパティーが 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 62. 「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

必須	いいえ
デフォルト	false
プロパティー・タイプ	Boolean
グローバル化されるかどうか	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 操作時に使用するファイル名。

表 63. 「デフォルト・ターゲット・ファイル名」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
グローバル化されるかどうか	はい

「ファイル転送タイプ」プロパティ (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 64. 「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
グローバル化されるかどうか	いいえ

固有ファイルの生成 (GenerateUniqueFile)

このプロパティが true に設定されていると、Outbound Create 操作の実行中、アダプターは固有ファイル名を作成します。このプロパティが true に設定されている場合、アダプターは、Filename プロパティに設定されているあらゆる値を無視します。

注: アダプターは、GenerateUniqueFile と StagingDirectory の両方のオプションを同時にはサポートしません。

表 65. 「固有ファイルの生成」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
グローバル化されるかどうか	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 66. 「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

表 66. 「ホスト名」プロパティ特性 (続き)

グローバル化されるかどうか	はい
---------------	----

「ファイル内のビジネス・オブジェクト間の区切り文字」プロパティ ー (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 67. 「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	<p>Create および Overwrite 操作の場合、デフォルト値は設定されていません。</p> <p>Append 操作の場合、デフォルト値は <EndB0> です。</p> <p>Append 操作の場合、次の規則が適用されます。</p> <ul style="list-style-type: none"> ビジネス・オブジェクト・ラッパーで区切り文字が null に設定されている場合、ビジネス・オブジェクトの分離に区切り文字は使用されません。 ビジネス・オブジェクト・ラッパーに IncludeEndBODelimiter プロパティが設定されておらず、対話仕様の中の値も null である場合、デフォルトは <EndB0> です。 ビジネス・オブジェクト・ラッパーに特定の区切り文字の値が指定されていれば、指定されている値が付加されます。 ビジネス・オブジェクト・ラッパーと対話仕様の両方に値が指定されていれば、ビジネス・オブジェクト・ラッパー値が優先されます。
プロパティ・タイプ	String
グローバル化されるかどうか	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 ー (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 68. 「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
使用法	<p>アダプターを始動する前に、アダプターを実行するマシン上に、LocalArchiveDirForCreate ディレクトリーを手動で作成する必要があります。アダプターはこのディレクトリーを自動的に作成しないためです。</p>
グローバル化されるかどうか	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」プロパティー (LocalArchivingEnabledForCreate)

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、このプロパティーが 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 69. 「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」プロパティー特性

必須	いいえ
デフォルト	false
プロパティー・タイプ	Boolean
グローバル化されるかどうか	いいえ

「ローカル・ディレクトリー」プロパティー (LocalDirectoryPath)

Outbound の create 操作時に、FileInLocalDirectory プロパティーが 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 プロパティーが 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 70. 「ローカル・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
使用法	アダプターを始動する前に、アダプターを実行するマシン上に、LocalDirectoryPath ディレクトリーを手動で作成する必要があります。アダプターはこのディレクトリーを自動的に作成しないためです。
グローバル化されるかどうか	はい

「ポート番号」プロパティー (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 71. 「ポート番号」プロパティー特性

必須	はい
デフォルト	FTP は 21、FTPS は 990
プロパティー・タイプ	Integer
グローバル化されるかどうか	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。

表 72. 「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
グローバル化されるかどうか	いいえ
例	例: 通常接続: FTP FTP over SSL 接続: FTPS

「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 73. 「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
グローバル化されるかどうか	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 74. 「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
グローバル化されるかどうか	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 75. 「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
グローバル化されるかどうか	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 76. 「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
グローバル化されるかどうか	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。

表 77. 「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

表 77. 「分割関数クラス名」プロパティ特性 (続き)

グローバル化されるかどうか	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 78. 「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
使用法	アダプターを始動する前に、アダプターを実行するマシン上に、StagingDirectory ディレクトリーを手動で作成する必要があります。アダプターはこのディレクトリーを自動的に作成しないためです。
グローバル化されるかどうか	はい

Inbound 構成プロパティ

WebSphere Adapter for FTP には、オブジェクトやサービスを生成したり作成したりするときに、外部サービス・ウィザードを使用して設定する、いくつかの種類の Inbound 接続構成プロパティがあります。WebSphere Integration Developer または管理コンソール 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。

プロパティの詳細についてのガイド

WebSphere Adapter for FTP を構成するときに使用されるプロパティは、リソース・アダプター・プロパティや Managed Connection Factory プロパティなど、それぞれの構成プロパティのトピックに記載されている表で詳細に説明されています。これらの表を使用しやすくするため、参照する各行の情報を以下に説明します。

次の表では、構成プロパティの表に表示される場合がある各行の意味を説明します。

行	説明
必須	<p>アダプターが動作するためには、必須フィールド (プロパティ) に値が必要です。必須プロパティに対しては、外部サービス・ウィザードがデフォルト値を提供する場合があります。</p> <p>外部サービス・ウィザードの必須フィールドからデフォルト値を除去しても、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入っていないと、外部サービス・ウィザードは、その割り当て済みのデフォルト値を使用してフィールドを処理し、そ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>可能な値は「はい」および「いいえ」です。</p> <p>プロパティは、他のプロパティが特定の値の場合のみ必須となることがあります。その場合は、表にこの依存関係が記載されます。以下に例を示します。</p> <ul style="list-style-type: none"> • EventQueryType プロパティが Dynamic に設定された場合は「はい」 • Oracle データベースの場合は「はい」
使用可能な値	プロパティで選択可能な値をリストして説明します。
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る事前定義値。プロパティが必須の場合は、デフォルト値を受け入れるか、ユーザーが値を指定する必要があります。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>None という語は、受け入れ可能なデフォルト値です。デフォルト値がないという意味ではありません。</p>
計測単位	プロパティの計測単位を指定します (例: キロバイト、秒)。
プロパティ・タイプ	<p>プロパティ・タイプを示します。有効なプロパティ・タイプとしては、以下のものがあります。</p> <ul style="list-style-type: none"> • Boolean • String • Integer
使用法	<p>プロパティに適用される場合がある使用の条件または制限について記述します。制限の記載例を以下に示します。</p> <p>Rational Application Developer for WebSphere Software バージョン 6.40 またはそれ以前では、パスワードに以下の制限があります。</p> <ul style="list-style-type: none"> • 大文字である必要があります • 長さが 8 文字である必要があります <p>Rational Application Developer for WebSphere Software バージョン 6.40 よりも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"> • 大文字小文字を区別しません • 長さが 40 文字まで可能です <p>このセクションでは、このプロパティに影響を及ぼす他のプロパティ、またはこのプロパティによって影響を受ける他のプロパティをリストし、その条件付き関係の内容を説明します。</p>
例	<p>プロパティ値のサンプルを示します。例:</p> <p>「言語が JA (日本語) に設定された場合、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	グローバル化されたプロパティには各国語サポートが備わっているため、値を各国語で設定できます。 有効な値は「はい」および「いいえ」です。
BIDI 対応	プロパティが双方向 (bidi) 処理でサポートされるかどうかを示します。双方向処理とは、1 つのファイルに左から右 (ヘブライ語やアラビア語など) と右から左 (URL やファイル・パスなど) の両方の意味内容を含むデータを処理するタスクを指します。 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

アダプター・タイプ・プロパティは、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・プロパティのいずれかを設定する場合、そのプロパティ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・プロパティおよびその目的を以下の表に示します。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。後続セクションのプロパティ詳細表の見方について詳しくは、169 ページの『プロパティの詳細についてのガイド』を参照してください。

表 79. アダプター・タイプ・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	
203 ページの『「説明」プロパティ (Description)』	Description	アダプターの説明。
203 ページの『「表示名」プロパティ (DisplayName)』	DisplayName	アダプター表示名。
203 ページの『「ID」プロパティ (ID)』	ID	アダプター・タイプの ID。
203 ページの『「ベンダー」プロパティ (Vendor)』	Vendor	アダプターを提供するベンダーの名前。

表 79. 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 80. 「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 81. 「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 82. 「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 83. 「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 84. 「バージョン」プロパティ特性

必須	はい
----	----

表 84. 「バージョン」プロパティ特性 (続き)

デフォルト	6.2
プロパティ・タイプ	String

リソース・アダプター・プロパティ

リソース・アダプター・プロパティは、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は、バージョン 6.1.x 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・プロパティおよびその目的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 85. Adapter for FTP 用のリソース・アダプター・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	
アダプター ID	AdapterID	PMI イベントのアダプター・インスタンス、ロギングおよびトレースのアダプター・インスタンスを識別する場合に使用します。
206 ページの『EISEncoding (EISEncoding)』	EISEncoding	FTP サーバーのエンコード。
(なし)	enableHASupport	このプロパティは変更しないでください。
206 ページの『ログ・ファイルおよびトレース・ファイルで、ユーザー・データを「XXX」と記述する (HideConfidentialTrace) プロパティ』	HideConfidentialTrace	ログおよびトレース・ファイルへの書き込み時に、ユーザー・データを「X」のストリングで記述し、潜在的な機密データを隠すようにするかどうかを指定します。
(なし)	ログ・ファイル最大サイズ	非推奨
(なし)	LogFilename	非推奨
(なし)	ログ・ファイル数	非推奨

表 85. Adapter for FTP 用のリソース・アダプター・プロパティー (続き)

プロパティー名		説明
ウィザード内	管理コンソール内	
(なし)	トレース・ファイル最大サイズ	非推奨
(なし)	トレース・ファイル名	非推奨
(なし)	トレース・ファイル数	非推奨

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) を識別します。

表 86. 「アダプター ID」の詳細

必須	はい
デフォルト	001
プロパティー・タイプ	String
使用法	<p>このプロパティーは、ログおよびトレース・ファイル内のアダプター・インスタンスを識別し、また、アダプターのモニター時にアダプター・インスタンスを識別する場合に役立ちます。アダプター ID は、アダプター固有の ID、FTPRA と共に使用され、Log and Trace Analyzer ツールによって使用されるコンポーネント名を構成します。例えば、アダプター ID プロパティーが、001 に設定されている場合、コンポーネント ID は、FTPRA001 となります。</p> <p>同じアダプターを複数インスタンス実行する場合、アダプター ID プロパティーの最初の 8 文字は、必ずインスタンスごとに固有にし、ログおよびトレース情報を特定のアダプター・インスタンスに相互に関連付けられるようにしてください。アダプター ID プロパティーの最初の 7 文字を固有にすることにより、そのアダプターの複数インスタンスのコンポーネント ID も固有になり、アダプターの特定インスタンスにログおよびトレース情報を相互に関連付けることができます。</p> <p>アダプター ID プロパティーの長さが、ログおよびトレース・ファイルのフィルター処理にどのように関係するかを示すため、2 つのインスタンスの WebSphere Adapter for FTP のアダプター ID プロパティーを 001 と 002 に設定するとします。これらのインスタンスのコンポーネント ID、FTPRA001 および FTPRA002 は、短いので固有性を保つことができ、別のアダプター・インスタンスとして区別することができます。しかし、もっと長いアダプター ID プロパティーのインスタンスの場合、互いを区別できなくなります。例えば、2 つのインスタンスのアダプター ID プロパティーを、Instance01 および Instance02 に設定するとします。この場合、各アダプター・インスタンスのログおよびトレース情報を個別に調べることはできなくなります。これは、両方のインスタンスのコンポーネント ID が FTPRAInstance に切り詰められるためです。</p> <p>Inbound 処理の場合、このプロパティーの値は、リソース・アダプター・レベルで設定されます。Outbound 処理の場合、リソース・アダプター・レベルと Managed Connection Factory レベルの両方で設定できます。外部サービス・ウィザードを使用して、アダプターを Outbound 処理用に構成した後、リソース・アダプター・プロパティーおよび Managed Connection Factory プロパティーを個別に設定できます。WebSphere Integration Developer アセンブリー・エディターまたは管理コンソールを使用して、これらのプロパティーを再設定する場合、ログおよびトレース・エントリーのマーキングが不整合にならないように、矛盾がない設定になっていることを確認してください。</p>
グローバル化	はい

表 86. 「アダプター ID」の詳細 (続き)

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グローバル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 87. 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 (Enable high availability support) (enableHASupport)

このプロパティーは変更しないでください。true に設定してください。

ログ・ファイルおよびトレース・ファイルで、ユーザー・データを「XXX」と記述する (HideConfidentialTrace) プロパティー

このプロパティーは、ログおよびトレース・ファイル中のユーザー・データを「X」のストリングに置換し、潜在的な機密データが許可なく外部に漏れないようにします。

表 88. ログ・ファイルおよびトレース・ファイルで、ユーザー・データを「XXX」と記述するの詳細

必須	いいえ
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	このプロパティーを True に設定すると、アダプターでは、ログおよびトレース・ファイルに書き込む時に、ユーザー・データを「X」のストリングに置換します。 Inbound 処理の場合、このプロパティーの値は、リソース・アダプター・レベルで設定されます。Outbound 処理の場合は、リソース・アダプター・レベルと Managed Connection Factory レベルの両方で設定できます。外部サービス・ウィザードを使用して、アダプターを Outbound 処理用に構成した後、リソース・アダプター・プロパティーおよび Managed Connection Factory プロパティーを個別に設定できます。WebSphere Integration Developer アセンブリー・エディターまたは管理コンソールを使用して、これらのプロパティーを再設定する場合、ログおよびトレース・エントリーのマーキングが不整合にならないように、矛盾がない設定になっていることを確認してください。
グローバル化	いいえ
BIDI 対応	いいえ

ログ・ファイルの最大サイズ (Log file maximum size) (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 89. 「ログ・ファイルの最大サイズ (Log file maximum size)」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 90. 「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 (Log number of files) (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 91. 「ログ・ファイルの数 (Log number of files)」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・ファイルがその最大サイズに到達すると、アダプターは別のログ・ファイルの使用を開始します。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 (Trace file maximum size) (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 92. 「トレース・ファイルの最大サイズ (Trace file maximum size)」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 93. 「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 (Trace number of files) (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・ファイルがその最大サイズに到達すると、アダプターは別のトレース・ファイルの使用を開始します。

表 94. 「トレース・ファイルの数 (Trace number of files)」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

活動化仕様プロパティ

活動化仕様プロパティは、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です。

活動化仕様プロパティは、エンドポイント・アクティベーション時に使用され、アダプターに適切なイベント・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 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、活動化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 95. 活動化仕様プロパティ

プロパティ名		説明
ウィザード内	管理コンソール内	
213 ページの『イベント送達は 1 回のみ (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。
213 ページの『「自動作成イベント・テーブル」プロパティ (EP_CreateTable)』	EP_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
213 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
214 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
214 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
214 ページの『「データベース・スキーマ名」プロパティ (EP_SchemaName)』	EP_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
215 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
215 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	前の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る
215 ページの『送達のタイプ (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。

表 95. 活動化仕様プロパティ (続き)

216 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	前の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る
216 ページの『「イベント・リカバリー DataSource (JNDI) 名」プロパティ (EP_DataSource_JNDIName)』	EP_DataSource_JNDIName	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 사용되는データ・ソースの JNDI 名
217 ページの『「イベント・リカバリー・テーブル名」プロパティ (EP_EventTableName)』	EP_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
217 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。
217 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
218 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
218 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 (FTPRenameExt)』	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
218 ページの『「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)』	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
219 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
219 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。
219 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
220 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
失敗したイベントの再試行制限 (FailedEventRetryLimit)	FailedEventRetryLimit	アダプターの再送信の試行回数で、この回数に達すると失敗とマークされます。
221 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 (FTPScriptFileExecutedAfterInbound)』	ftpScriptFileExecutedAfterInbound	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する

表 95. 活動化仕様プロパティ (続き)

221 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 (FTPScriptFileExecutedBeforeInbound)』	ftpScriptFileExecutedBeforeInbound	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する
222 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
222 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	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る
222 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 (LocalArchiveDirectory)』	LocalArchiveDirectory	ローカル・アーカイブ・ディレクトリーの絶対パス
223 ページの『「ローカル・ディレクトリー」プロパティ (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
223 ページの『最大接続数 (Maximum connections) (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
223 ページの『最小接続数 (Minimum connections) (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
224 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 (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
パスフレーズ・プロパティ (passPhrase)	passPhrase	秘密鍵を暗号化して、拡張セキュリティのために使用されます
224 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ユーザーのパスワード
224 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 (EP_Password)』	EP_Password	イベント・パーシスタンス時に使用されるパスワード
225 ページの『ポーリング期間の間隔 (ポーリング間隔)』	ポーリング間隔	ポーリング期間中にアダプターが待機する時間の長さ
225 ページの『ポーリング期間内の最大イベント数 (ポーリング数量)』	ポーリング数量	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
226 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
226 ページの『「秘密鍵ファイル・プロパティ (PrivateKeyFilePath)』	PrivateKeyFilePath	セキュア・シェル・サーバーに認証させるために使用される秘密鍵
226 ページの『「プロトコル」プロパティ (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
228 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 (EventFileMask)』	EventFileMask	イベント・ファイル用のフィルター

表 95. 活動化仕様プロパティ (続き)

開始時に EIS 接続を再試行する	RetryConnectionOnStartup	アダプターが始動時に FTP サーバー に接続できない場合に、接続を再試行するかどうかを指定します。
229 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
229 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。
227 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
228 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
サーバー検査を使用可能にする	EnableServerVerification	SFTP プロトコルのリモート・サーバー検査を使用可能にする
ホスト鍵ファイル	HostKeyFile	トラステッド・サーバーのホスト鍵を含んだホスト鍵ファイルの絶対パス
231 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
231 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
231 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
231 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
232 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
232 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
233 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
234 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。
234 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
235 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
235 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 (EP_UserName)』	EP_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

表 95. 活動化仕様プロパティ (続き)

ルール・エディター	ruleTable	イベントのフィルター処理に使用されるルールのコレクション
-----------	-----------	------------------------------

イベント送達は 1 回のみ (AssuredOnceDelivery)

このプロパティでは、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 96. 「イベント送達は 1 回のみ」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・タイプ	Boolean
使用法	<p>このプロパティを 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 (EP_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 97. 「自動作成イベント・テーブル」プロパティ特性

必須	いいえ
デフォルト	true
プロパティ・タイプ	Boolean
グローバル化されるかどうか	いいえ

「テーブル作成」プロパティ (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 98. 「テーブル作成」プロパティ特性

必須	いいえ
デフォルト	true
プロパティ・タイプ	Boolean
グローバル化されるかどうか	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 99. 「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
グローバル化されるかどうか	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 100. 「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
グローバル化されるかどうか	はい

「データベース・スキーマ名」プロパティ (EP_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 101. 「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
グローバル化されるかどうか	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 102. 「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
グローバル化されるかどうか	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 103. 「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
グローバル化されるかどうか	いいえ

送達のタイプ (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 104. 送達のタイプの詳細

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">• 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。• 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。
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 と活動化仕様レベルの 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、活動化仕様レベルの 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、複数の活動化仕様を使用し、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、活動化仕様レベルの EISEncoding を設定する場合、活動化仕様レベルの値は、FTP サーバーとの通信中に制御接続で設定されます。値は活動化仕様レベルであるため、これが適用されるのは、その活動化仕様のみです。
- アダプター・レベルの EISEncoding と活動化仕様レベルの EISEncoding の両方を設定する場合、活動化仕様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 105. 「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
グローバル化されるかどうか	はい

「イベント・リカバリー DataSource (JNDI) 名」プロパティ (EP_DataSource_JNDIName)

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 106. 「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
グローバル化されるかどうか	はい

「イベント・リカバリー・テーブル名」プロパティ (EP_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。複数の活動化仕様を使用する場合、この値は活動化仕様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 107. 「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
グローバル化されるかどうか	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 108. 「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、値: Customer,Order を指定します。
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 109. 「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

表 109. 「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 (続き)

グローバル化されるかどうか	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 110. 「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
グローバル化されるかどうか	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 (FTPRenameExt)

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 111. 「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
グローバル化されるかどうか	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 112. 「内容ではなくファイル名およびディレクトリーのみの受け渡し」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
グローバル化されるかどうか	いいえ

「ファイル転送タイプ」プロパティー (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 113. 「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
グローバル化されるかどうか	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 114. 「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 115. 「一度に取得するファイル数」プロパティー特性

必須	はい
----	----

表 115. 「一度に取得するファイル数」プロパティ特性 (続き)

デフォルト	10
プロパティ・タイプ	Integer
グローバル化されるかどうか	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 (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 116. 「各ダウンロード間のポーリング期間の数」プロパティ特性

必須	はい
デフォルト	5
プロパティ・タイプ	Integer
グローバル化されるかどうか	いいえ

失敗したイベントの再試行制限 (FailedEventRetryLimit)

このプロパティは、アダプターがイベントの再送信を試みる回数を指定します。この回数に達すると失敗とマークされます。

表 117. 「失敗したイベントの再試行制限」の詳細

必須	いいえ
使用可能な値	整数
デフォルト	5
プロパティ・タイプ	Integer

表 117. 「失敗したイベントの再試行制限」の詳細 (続き)

使用法	<p>このプロパティは、この回数以上は失敗とマークする場合の、アダプターがイベントを送信する回数を指定する時に使用します。以下のいずれかの値を取ります。</p> <p>デフォルト このプロパティが設定されない場合、アダプターは、イベント送信失敗の後、さらに 5 回イベントの送信を試み、それでも送達できない場合にイベントを失敗とマーク付けします。</p> <p>0 アダプターは、回数無制限でイベントの送信を試行します。このプロパティが 0 に設定されると、イベントはイベント・ストアに残されたままになり、イベントが失敗とマークされることはなくなります。</p> <p>> 0 正の整数の場合、アダプターは、指定した回数再試行を行った後、イベントを失敗とマークします。</p> <p><0 負の整数の場合、アダプターは失敗したイベントの送信を再試行しません。</p>
グローバル化	いいえ
BIDI 対応	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 (FTPScriptFileExecutedAfterInbound)

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 118. 「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
グローバル化されるかどうか	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 (FTPScriptFileExecutedBeforeInbound)

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。このフィーチャーを使用して、各ポーリング周期の前に FTP サーバーに対して追加アクションを実行できます。ftpScriptFileExecutedAfterInbound

表 119. 「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
グローバル化されるかどうか	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 120. 「テーブル作成」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
グローバル化されるかどうか	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 121. 「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
プロパティ・タイプ	String
グローバル化されるかどうか	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 122. 「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
使用法	<p>WebSphere Application Server 環境変数を使用して、ローカル・アーカイブ・ディレクトリーを表すことができます。\$ 記号の後に、環境変数の名前を中括弧で囲んで指定します。例えば、\${LOCALARCHIVE_DIRECTORY} です。この資料の 71 ページの『WebSphere Application Server 環境変数の定義』についてのトピックを参照してください。</p> <p>注: アダプターを始動する前に、アダプターを実行するマシン上に、LocalArchiveDirectory を手動で作成する必要があります。アダプターはこのディレクトリーを自動的に作成しないためです。</p>
グローバル化されるかどうか	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 123. 「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
使用法	WebSphere Application Server 環境変数を使用して、ローカル・イベント・ディレクトリーを表すことができます。\$ 記号の後に、環境変数の名前を中括弧で囲んで指定します。例えば、\${LOCAL_DIRECTORY} です。この資料の 71 ページの『WebSphere Application Server 環境変数の定義』についてのトピックを参照してください。 注: アダプターを始動する前に、アダプターを実行するマシン上に、 LocalEventDirectory を手動で作成する必要があります。アダプターはこのディレクトリーを自動的に作成しないためです。
グローバル化されるかどうか	はい

最大接続数 (Maximum connections) (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 124. 「最大接続数 (Maximum connections)」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。アダプターは、1 より小さい正の入力値を 1 であるとみなします。このプロパティに対して負の値または 1 を入力すると、実行時エラーが発生することがあります。
グローバル化	いいえ
BIDI 対応	いいえ

最小接続数 (Minimum connections) (MinimumConnections)

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小数を指定します。

表 125. 「最小接続数 (Minimum connections)」の詳細

必須	いいえ
デフォルト	1
プロパティ・タイプ	Integer

表 125. 「最小接続数 (Minimum connections)」の詳細 (続き)

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。このプロパティに対して負の値または 1 を入力すると、実行時エラーが発生することがあります。
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 (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。このプロパティは、LocalArchiveDirectory が有効で存在する場合にのみ使用されます。

表 126. 「ローカル・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	original
プロパティ・タイプ	String
グローバル化されるかどうか	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。EventDirectory 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、このプロパティに値を指定する必要はありません。

表 127. 「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
グローバル化されるかどうか	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 (EP_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 128. 「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
グローバル化されるかどうか	はい

ポーリング期間の間隔 (ポーリング間隔)

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 129. 「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 (ポーリング数量)

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 130. 「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場合があります。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

パスフレーズ・プロパティ (passPhrase) (passPhrase)

このプロパティは、秘密鍵を暗号化して、拡張セキュリティのために使用されます。

表 131. パスフレーズ・プロパティ (passPhrase) プロパティの特性

必須	いいえ
使用可能な値	任意のストリング値
デフォルト	なし
プロパティ・タイプ	String

表 131. パスフレーズ・プロパティ (passPhrase) プロパティの特性 (続き)

使用法	拡張セキュリティに使用されます。これは、SFTP 構成の中で秘密鍵を暗号化することによって、秘密鍵を保護します。
グローバル化されるかどうか	はい
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 132. 「ポート番号」プロパティ特性

必須	はい
デフォルト	FTP は 21、FTPS は 990、SFTP は 22。
プロパティ・タイプ	Integer
グローバル化されるかどうか	いいえ

秘密鍵ファイル・プロパティ (PrivateKeyFilePath)

このプロパティによって、秘密鍵を参照して選択することができます。その秘密鍵は、セキュア・シェル・サーバーに対する認証に使用されます。

表 133. 「秘密鍵」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
使用法	秘密鍵を格納するファイルの絶対パス。ユーザーをセキュア・シェル・サーバーに認証させるために使用されます。
例	c:%temp%key.ppk
グローバル化されるかどうか	はい
BIDI 対応	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

FTP over SSL 接続: FTPS

FTP over SSH 接続: SFTP

表 134. 「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
グローバル化されるかどうか	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 135. 「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
使用法	<p>WebSphere Application Server 環境変数を使用して、リモート・アーカイブ・ディレクトリーを表すことができます。\$ 記号の後に、環境変数の名前を中括弧で囲んで指定します。例えば、\${REMOTEARCHIVE_DIRECTORY} です。この資料の 71 ページの『WebSphere Application Server 環境変数の定義』についてのトピックを参照してください。</p> <p>FTP サーバーにあり、Inbound 構成で 사용되는アーカイブ・ディレクトリーは、アーカイブ・ディレクトリーの絶対パスを表します。これにはホスト名または URL 情報は含まれません。このディレクトリーは、イベント・ディレクトリーが存在するのと同じ FTP サーバーにあります。例えば、/home/archive です。</p> <p>注: アダプターを始動する前に、アダプターを実行するマシン上に、FTPArchiveDirectory を手動で作成する必要があります。アダプターはこのディレクトリーを自動的に作成しないためです。</p>
グローバル化されるかどうか	はい

「リモート・ディレクトリー」プロパティ (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 136. 「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
使用法	WebSphere Application Server 環境変数を使用して、リモート・ディレクトリーを表すことができます。\$ 記号の後に、環境変数の名前を中括弧で囲んで指定します。例えば、\${REMOTE_DIRECTORY} です。この資料の 71 ページの『WebSphere Application Server 環境変数の定義』についてのトピックを参照してください。 注: アダプターを始動する前に、アダプターを実行するマシン上に、EventDirectory を手動で作成する必要があります。アダプターはこのディレクトリーを自動的に作成しないためです。
グローバル化されるかどうか	はい

「このパターンを持つファイルの取得」プロパティ (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 137. 「このパターンを持つファイルの取得」プロパティ特性

必須	はい
デフォルト	*.*
プロパティ・タイプ	String
グローバル化されるかどうか	はい

開始時に EIS 接続を再試行する (RetryConnectionOnStartup)

このプロパティは、アダプターが始動時に FTP サーバー に接続できない場合に、再度接続を試みるかどうかを指定します。

表 138. 「開始時に EIS 接続を再試行する」の詳細

必須	いいえ
使用可能な値	True False
デフォルト	False
プロパティ・タイプ	Boolean

表 138. 「開始時に EIS 接続を再試行する」の詳細 (続き)

使用法	<p>このプロパティは、アダプターの始動時に、FTP サーバー に接続できない場合に、接続を再試行するかどうかを指定します。</p> <ul style="list-style-type: none"> アダプターが、FTP サーバー に対する接続を確立できたかどうかに関するフィードバックを直ちに必要とする場合、例えば、アダプターからのイベントを受信するアプリケーションを作成し、テストしている場合は、このプロパティを <code>False</code> に設定します。アダプターが接続できない場合、アダプターは、ログおよびトレース情報を書き込んで、停止します。管理コンソールは、アプリケーション状況を <code>Stopped</code> と表示します。この場合、接続の問題を解決後、手動でアダプターを始動してください。 接続に関するフィードバックをすぐに必要としない場合は、このプロパティは <code>True</code> に設定します。アダプターが始動時に接続できない場合、アダプターはログおよびトレース情報を書き込んでから、<code>RetryInterval</code> プロパティで再試行の頻度を判別して再接続を試み、<code>RetryLimit</code> プロパティの値で指定された値に達するまで、再試行を複数回行います。管理コンソールは、アプリケーション状況を <code>Started</code> と表示します。
グローバル化	いいえ
BIDI 対応	いいえ

接続が失敗した場合の再試行間隔 (RetryInterval)

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 139. 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・タイプ	Integer
使用法	<p>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 <code>Inbound</code> 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。</p>
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 140. 「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	0 および正の整数
デフォルト	0
プロパティ・タイプ	Integer

表 140. 「システム接続を再試行する回数」の詳細 (続き)

使用法	このプロパティは、アダプターが FTP サーバー に接続して Inbound 処理を実行できない場合に、接続を再試行する回数を指定します。値が 0 の場合は、再試行回数が無制限になることを指定します。 アダプターの当初始動時に、FTP サーバー に接続できない場合、アダプターが再試行するかどうかを制御するには、RetryConnectionOnStartup プロパティを使用します。
グローバル化	はい
BIDI 対応	いいえ

サーバー検査を使用可能にする プロパティ (EnableServerVerification)

このプロパティは、SFTP プロトコルのリモート・サーバー検査を使用可能にするために使用されます。

表 141. サーバー検査を使用可能にする プロパティの詳細

必須	いいえ
使用可能な値	True False
デフォルト	False
プロパティ・タイプ	Boolean
使用法	このプロパティの設定内容によって、次のようになります。 <ul style="list-style-type: none"> • True に設定されると、サーバー認証は使用可能です。 • False に設定されると、サーバー認証は使用不可です。 アダプターは、トラステッド・サーバーのホスト鍵を含んだファイルのパスの中の、HostKeyFile プロパティを検査します。
グローバル化	はい
BIDI 対応	いいえ

ホスト鍵ファイル プロパティ (HostKeyFile)

このプロパティは、トラステッド・サーバーのホスト鍵を含むホスト鍵ファイルの絶対パスを指定します。

表 142. ホスト鍵ファイル プロパティの特性

必須	EnableServerVerification が使用可能な場合、このプロパティを指定する必要があります。
デフォルト	なし
プロパティ・タイプ	String
使用法	これは、このファイルに指定されているトラステッド・サーバーのホスト鍵でリモート・サーバーのホスト鍵を検査するために、アダプターが使用します。
グローバル化されるかどうか	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 143. 「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
グローバル化されるかどうか	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 144. 「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
グローバル化されるかどうか	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 145. 「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
グローバル化されるかどうか	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 146. 「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
グローバル化されるかどうか	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、活動化仕様の `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 147. 「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	ソートなし (= ソートされません)
プロパティ・タイプ	String
グローバル化されるかどうか	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。

注: 複数の COBOL コピーブック・レコードを含む入力ファイルの場合、サイズによってファイル分割ができるように、それぞれのレコードの正しい長さを指定する必要があります。それぞれのレコードのサイズを判別するには、次のいずれかの方式を使用します。

1. テキスト・エディターでビジネス・オブジェクトを開きます。
 - a. 例:

```

<element name="CustomerNumber">
  <annotation>
    <appinfo source="http://www.ibm.com/cam/2005/typedescriptor">
      <td:typeDescriptorElement>
        <td:initialValue kind="SPACE"/>
        <td:simpleInstanceTD accessor="readWrite" attributeInBit="false"
          contentSize="5" offset="0" size="5">
          <td:sharedType>
            <td:stringTD addrUnit="byte" alignment="byte" characterSize="1"
              lengthEncoding="fixedLength" paddingCharacter=" "
              prefixLength="0" width="5"/>
            </td:sharedType>
          </td:simpleInstanceTD>
        </td:typeDescriptorElement>
      </appinfo>
    </annotation>
    <simpleType>
      <restriction base="string">
        <maxLength value="5"/>
      </restriction>
    </simpleType>
  </element>

```

ビジネス・オブジェクト内の各エレメントには、対応する <element> エレメントがあります。

- b. 各エレメント・タグの制限タグを見つけます (COBOL データ・バインディングには固定長のデータ・ハンドラーが必要です)。
 - c. 各エレメントの maxLength 属性値を加算します。この例では、値は 5 です。maxLength 値の合計は、タイプが DFHCOMMAREA の各レコードのサイズです。
2. テキスト・エディターでビジネス・オブジェクトを開きます。
- a. 名前属性にビジネス・オブジェクト名の値を持つ複合タイプ・タグを見つけます。次の例では、ビジネス・オブジェクト名は DFHCOMMAREA です。
 - b. 名前空間が追加された、aggregateInstanceTD という名前のタグを見つけ、属性 contentSize の値を使用します。この例では、値は 117 です。これは、タイプ DFHCOMMAREA のそれぞれのレコードのサイズです。

```

<complexType name="DFHCOMMAREA">
  <annotation>
    <appinfo source="http://www.ibm.com/cam/2005/typedescriptor">
      <td:typeDescriptorCT>
        <td:aggregateInstanceTD accessor="readWrite" attributeInBit="false"
          contentSize="117" offset="0" size="117">

```

表 148. 「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
グローバル化されるかどうか	はい

「分割関数クラス名」プロパティ

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.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.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` プロパティで指定します。
`EventContentType` 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 149. 「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	<code>com.ibm.j2ca.utils.filesplit.SplitBySize</code>
プロパティ・タイプ	<code>String</code>
グローバル化されるかどうか	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 150. 「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	<code>True</code> <code>False</code>
デフォルト	<code>False</code>
プロパティ・タイプ	<code>Boolean</code>
使用法	このプロパティを <code>True</code> 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 <code>False</code> 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 151. 「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
グローバル化されるかどうか	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 152. 「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
グローバル化されるかどうか	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 (EP_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 153. 「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
グローバル化されるかどうか	はい

ルール・ベースのイベント・フィルター処理 (ruleTable)

このプロパティは、一定のルール・セットに基づいてイベント・ファイルをフィルター処理する場合に使用されます。

表 154. ルール・ベースのイベント・フィルター処理

必須	任意指定
デフォルト	なし
プロパティ・タイプ	String
使用法	Inbound 処理時に、ルール・テーブルの値が指定されている場合、指定されたルールに基づいてフィルター処理後、イベント・ファイルを取り出してから、これらのイベント・ファイルのポーリングを行います。
グローバル化	はい

表 154. ルール・ベースのイベント・フィルター処理 (続き)

BIDI 対応	いいえ
---------	-----

グローバル化

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、グローバル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。

グローバル化および双方向変換

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。

グローバル化

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 では、Windows の標準形式が使用されますが、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では、異なる形式が使用される場合があります。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 は、ILYNN (暗黙、左から右、オン、オフ、公称) の双方向形式を使用します。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、これらの各属性に値を割り当てます。属性と設定値を次の表に示します。

表 155. 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	スキーマの配列	I	暗黙 (論理的)	I
		V	表示	
2	方向	L	左から右	L
		R	右から左	
		C	コンテキスト上の左から右	
		D	コンテキスト上の右から左	
3	対称スワッピング	Y	対称スワッピングのオン	Y
		N	対称スワッピングのオフ	
4	テキストの形状指定	S	テキストの形状を指定する	N
		N	テキストの形状を指定しない (名目)	
		I	語頭形の指定	
		M	語中形の指定	
		F	語尾形の指定	
		B	独立形の指定	
5	数字の形状指定	H	各国 (ヒンディ語)	N
		C	コンテキストによる形状指定	
		N	数字の形状を指定しない (名目)	

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 156. 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。

表 156. 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
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。
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。
- **Managed (J2C) Connection Factory 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。Managed Connection Factory 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。
- **活動化仕様プロパティ:** 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 (活動化

仕様) トランザクションと Outbound (Managed Connection Factory) 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・アダプターと活動化仕様の両方の EIS タイプ双方向プロパティに値を設定した場合、Inbound トランザクションに起因する変換では、活動化仕様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。

ビジネス・オブジェクト内の双方向変換

Outbound 処理の場合、WebSphere Adapter for FTP ビジネス・オブジェクトの Wrapper プロパティと、コンテンツ固有のビジネス・オブジェクトまたは汎用ビジネス・オブジェクトのデータの双方向変換を可能にするために、ビジネス・オブジェクトを変更できます。

複合タイプのビジネス・オブジェクトに注釈を追加して、次のビジネス・オブジェクトのファイルに双方向形式属性を指定する必要があります。

- 汎用ビジネス・オブジェクトの場合、FTPFile.xsd ファイルを変更します。
- ユーザー定義のビジネス・オブジェクトの場合、カスタム・ラッパー (例えば、CustomWrapper.xsd ファイル および Customer.xsd) を変更します。
- UnstructuredContent ビジネス・オブジェクトの場合、UnstructuredContent.xsd を変更します。

次のセクションで、注釈の例を示します。

ビジネス・オブジェクトの双方向形式属性

次の注釈は双方向のコンテキスト情報を含んでおり、FTP ビジネス・オブジェクトのすべての属性に適用できます。FTPFileBaseDataBinding は、エレメント BiDiContext の中の双方向情報を使用して、すべての属性を変換します。

```
<xsd:complexType name="Customer">
<xsd:annotation>
  <xsd:appinf
    source="http://www.ibm.com/xmlns/prod/websphere/j2ca/datatrans
formation/databindingm
apping">
  <dtm:DataBindingMapping
    xsi:type="dtm:DataBindingMapping"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:dtm="http://www.ibm.com/xmlns/prod/websphere/j2ca/da
tatransformation/databindingmapping">
  <BiDiContext>
    <orientation>rtl</orientation>
    <textShape>nominal</textShape>
    <orderingScheme>visual</orderingScheme>
    <symmetricSwapping>true</symmetricSwapping>
    <numeralShapes>nominal</numeralShapes>
```

```

        </BiDiContext>
    </dtm:DataBindingMapping>
</xsd:appinfo>
</xsd:annotation>

```

ラッパーの双方向形式属性

ユーザー定義タイプのビジネス・オブジェクトのラッパーに、注釈を追加できます。汎用 (FTPFile) およびユーザー定義タイプ (CustomerWrapper) といった Wrapper ビジネス・オブジェクトの中の注釈は、ラッパー属性の双方向変換を行うために使用されます。Wrapper ビジネス・オブジェクト内部で使用されるコンテンツ固有のビジネス・オブジェクトは、Wrapper ビジネス・オブジェクトの中の注釈を使用して変換されることはありません。コンテンツ固有のビジネス・オブジェクトを変換するには、個々のビジネス・オブジェクト定義を編集して、上記のビジネス・オブジェクトの双方向形式属性の例に示す注釈を追加する必要があります。

次の注釈は、ラッパー用の例です。

```

<complexType name="CustomerWrapper">
<annotation>
    <appinfo
        source="http://www.ibm.com/xmlns/prod/websphere/j2ca/
datatransformation/databindingmapping">
        <dtm:DataBindingMapping
            xsi:type="dtm:DataBindingMapping"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:dtm="http://www.ibm.com/xmlns/prod/websphere/j2ca/
datatransformation/databindingmapping">
            <BiDiContext>
                <orientation>rtl</orientation>
                <textShape>nominal</textShape>
                <orderingScheme>visual</orderingScheme>
                <symmetricSwapping>true</symmetricSwapping>
                <numeralShapes>nominal</numeralShapes>
            </BiDiContext>
        </dtm:DataBindingMapping>
    </appinfo>
</annotation>

```

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た後、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。

Managed (J2C) Connection Factory 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。

- Username
- Password
- Directory
- FileName
- StagingDirectory

- SecondServerUsername
- SecondServerPassword
- SecondServerDirectory
- SocksProxyUsername
- SocksProxyPassword
- FileSequenceLog

活動化仕様プロパティ

以下の活動化仕様プロパティは、双方向変換を制御します。

- Username
- Password
- EventDirectory
- EventFileMask
- FTPArchiveDirectory
- LocalEventDirectory
- LocalArchiveDirectory
- FTPScriptFileExecutedBeforeInbound
- FTPScriptFileExecutedAfterInbound
- FTPRenameExt
- FailedArchiveExt
- OriginalArchiveExt
- SuccessArchiveExt
- 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
- FtpServerEventDirectory
- SecondServerDirectory

- SecondServerUsername
- SecondServerPassword
- LocalDirectoryPath
- LocalArchiveDirForCreate
- StagingDirectory
- ArchiveDirectoryForRetrieve

アダプター・メッセージ

WebSphere Adapter for FTP によって送出されたメッセージを以下の場所に表示します。

メッセージのリンク先は <http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.wbit.620.help.messages.doc/messages.html> です。

表示される Web ページには、メッセージ接頭語のリストがあります。メッセージ接頭語をクリックすると、以下に示すように、その接頭語があるすべてのメッセージを参照できます。

- 接頭語 CWYFT があるメッセージの送出元は WebSphere Adapter for FTP です。
- 接頭語 CWYBS があるメッセージの送出元はアダプター・ファウンデーション・クラスで、これらのクラスはすべてのアダプターによって使用されます。

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 の関連情報が含まれています。

サンプルおよびチュートリアル

WebSphere Adapters を使用する際に役立つように、サンプルおよびチュートリアルがビジネス・プロセス・マネージメントのサンプルおよびチュートリアルの Web サイトに掲載しています。サンプルおよびチュートリアルには、次のいずれかの方法でアクセスできます。

- WebSphere Integration Developer を始動すると表示される「ようこそ」ページ。WebSphere Adapter for FTP のサンプルおよびチュートリアルを表示するには、「取得」をクリックします。表示されたカテゴリーをブラウズして、選択を行います。
- Web 上の <http://publib.boulder.ibm.com/bpcsamp/index.html> のページ。

情報リソース

- WebSphere Business Process Management の情報リソース Web ページ (<http://www14.software.ibm.com/webapp/wsbroker/redirect?version=pix&product=wps-dist&topic=bpmroadmaps>) には、記事、Redbooks、資料、および研修用資料へのリンクが組み込まれており、WebSphere Adapters を習得するのに役立ちます。

- WebSphere Adapters ライブラリーのページ (<http://www.ibm.com/software/integration/wbiadapters/library/infocenter/>) には、資料の全バージョンへのリンクが組み込まれています。

関連製品の情報

- WebSphere Business Process Management バージョン 6.2 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/v6r2mx/index.jsp>
- WebSphere Adapters バージョン 6.1.x インフォメーション・センター: <http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.wbit.612.help.adapter.emd.ui.doc/topics/tcreatecmps.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

developerWorks® リソース

- WebSphere Adapter Toolkit
- WebSpherebusiness 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 テクニカル・ノート: <http://www.ibm.com/support/search.wss?tc=SSMKUK&rs=695&rank=8&dc=DB520+D800+D900+DA900+DA800+DB560&dtm>。「**Product category**」リストで、アダプターの名前を選択して、「**Go**」をクリックします。

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すが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。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認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名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、さまざま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 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、および `ibm.com` は、International Business Machines Corporation の米国およびその他の国における商標です。これらおよび他の IBM 商標に、この情報の最初に現れる個所で商標表示 (® または ™) が付されている場合、これらの表示は、この情報が公開された時点で、米国において、IBM が所有する登録商標またはコモン・ロー上の商標であることを示しています。このような商標は、その他の国においても登録商標またはコモン・ロー上の商標である可能性があります。現時点での IBM の詳しい商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

Linux は、Linus Torvalds 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。

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。

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
アセンブリー・エディター 117
アダプター実装 23
アダプターのアーキテクチャー 2
アダプターのパフォーマンス 147
アダプター用パッケージ・ファイル 155
アダプター・アプリケーション
開始 146
停止 146
アダプター・アプリケーションの開始 146
アダプター・アプリケーションの停止 146
アダプター・テクニカル・ノート 243
アダプター・パターン・ウィザード 73
アダプター・メッセージ 242
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
隠蔽、重要データ 29
エンドポイント 117

[カ行]

外部サービス接続プロパティ 171, 202
外部サービス・ウィザード
開始 83
始動 101
接続プロパティ 101
概要 1
カスタム・ビジネス・オブジェクト 168
カスタム・プロパティ
活動化仕様 139, 144
リソース・アダプター 135, 141
Managed Connection Factory 137, 142
活動化仕様プロパティ 209
管理コンソールでの設定 139, 144
リスト 209
関数セレクター 12
関連情報 242
関連製品、情報 242
技術概説 2
技術情報 2
機密データ、隠蔽 29
機密トレース 29

区切り文字 14, 15
組み込みアダプター
活動化仕様プロパティ、設定 139
使用する場合の考慮事項 32
説明 30
リソース・アダプター・プロパティ、設定 135
Managed Connection Factory プロパティ、設定 137
クラスター化された環境
説明 33
デプロイ 33
Inbound 処理 34
Outbound 処理 35
研修、WebSphere Adapters 242
公開鍵 28
高可用性環境
説明 33
デプロイ 33
Inbound 処理 34
Outbound 処理 35
構成
トレース 154
ロギング 154
Performance Monitoring Infrastructure (PMI) 148
後方互換性
プロジェクト 39
プロジェクト交換ファイル 39
互換性マトリックス 2
コンポーネントの接続 125

[サ行]

再試行制限プロパティ 229
サポート
概要 153
セルフ・ヘルプ・リソース 161
テクニカル 243
サポート、Log and Trace Analyzer 153
サポートされる Outbound 操作 4
参照バインディング 92
参照バインディング、Inbound 117
サンプル 41
事前定義ビジネス・オブジェクト 168
実行時環境
認証 29
EAR ファイルのデプロイ先 129
実装環境、Java 126
紹介 1
スタンドアロン参照 92
接続 117
スタンドアロン・アダプター
活動化仕様プロパティ、設定 144

スタンドアロン・アダプター (続き)
 使用する場合の考慮事項 33
 説明 30
 リソース・アダプター・プロパティ、設定 141
 Managed Connection Factory プロパティ、設定 142
 成果物、生成 98
 成果物の生成 98
 成果物の生成、Inbound 121
 製品概要 1
 セキュア FTP 23
 セキュリティー 23
 重要データの隠蔽 29
 セキュリティー、Java 2 30
 接続 92
 接続プロパティの設定 84, 101
 説明 172, 203
 セルフ・ヘルプ・リソース 161
 ソフトウェア要件 2

[夕行]

ターゲット・コンポーネント 125
 対話仕様プロパティ 190
 変更 123
 チャンク機能 15
 チュートリアル 41
 注釈 3
 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 3
 定義、WebSphere Application Server 環境変数 71
 テクニカル・サポート 243
 テクニカル・ノート 161, 243
 テクニカル・ノート、WebSphere Adapters 242
 テスト環境
 デプロイ先 125, 127
 モジュールの追加先 127
 モジュールのテスト 128
 デバッグ
 セルフ・ヘルプ・リソース 161
 org.xml.sax.SAXParseException 例外 160
 XAResourceNotAvailableException 例外 160
 デプロイメント
 オプション 30
 環境 125
 実稼働環境への 129
 テスト環境への 125
 トラストストア 24, 26
 トラブルシューティング
 概要 153
 セルフ・ヘルプ・リソース 161
 org.xml.sax.SAXParseException 例外 160
 XAResourceNotAvailableException 例外 160
 トレース
 管理コンソールを使用したプロパティの構成 155
 トレース・ファイル
 使用可能化 155

トレース・ファイル (続き)
 詳細レベル 155
 使用不可化 155
 場所 157
 ファイル名の変更 156

[ナ行]

認証 28
 実行時 29
 説明 29
 認証別名 30, 45

[ハ行]

バージョン 172, 203
 ハードウェア要件 2
 ハードウェア要件とソフトウェア要件 2
 バススルー処理 14
 パターン 73
 バッシブ FTP モード 23
 パフォーマンスに関する統計 151
 パフォーマンスのモニター 147
 ビジネス・オブジェクト 20, 163
 命名規則 166
 ビジネス・オブジェクト、カスタム 168
 ビジネス・オブジェクト、事前定義 47, 48
 ビジネス・オブジェクト、COBOL コピーブック・ファイルか
 ら変換 61
 ビジネス・オブジェクト、COBOL コピーブック・ファイルへ
 の変換 51
 ビジネス・オブジェクトおよびサービスの選択: Outbound 87
 ビジネス・オブジェクト属性プロパティ 167
 ビジネス・オブジェクトの構造 163
 ビジネス・オブジェクトの選択、Inbound 115
 ビジネス・オブジェクトの操作サポート 167
 ビジネス・オブジェクトの命名規則 166
 ビジネス・グラフ 3
 ビジネス・フォールト 158
 表示名 (DisplayName) 172, 203
 ファイアウォール 23
 ファイル
 SystemOut.log ログ・ファイル 156
 trace.log トレース・ファイル 156
 ファイル分割 15
 フォールト
 説明 158
 プロジェクト、作成 83
 プロジェクト交換 (PI) ファイル
 マイグレーションなしでの更新 39
 プロパティ
 活性化仕様 139, 144
 リスト 209
 構成プロパティ
 Inbound 200

プロパティ (続き)
構成プロパティ (続き)
 Outbound 169
リソース・アダプター 135, 141
Inbound 構成 200
Managed (J2C) Connection Factory 137, 142
 Outbound 構成 169
別名、認証 45
ベンダー (Vendor) 172, 203

[マ行]

マイグレーションに関する考慮事項 36
マトリックス、互換性 2
メッセージ、アダプター 242
文字セット 84, 101
モジュール、作成 47
モジュールの構成のためのロードマップ 43
問題判別
 セルフ・ヘルプ・リソース 161
 org.xml.sax.SAXParseException 例外 160
 XAResourceNotAvailableException 例外 160

[ヤ行]

要件、ハードウェアおよびソフトウェア 2

[ラ行]

ラッパー・ビジネス・オブジェクト 3
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
 サーバーへのインストール 129
 説明 129
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 135, 141
 詳細 173, 204
例外
 org.xml.sax.SAXParseException 160
 XAResourceNotAvailableException 160
連邦情報処理標準 26
ロギング
 管理コンソールを使用したプロパティの構成 155
ログ・アナライザー 155
ログ・ファイル
 使用可能化 155
 詳細レベル 155
 使用不可化 155
 場所 157
 ファイル名の変更 156
ログ・ファイルとトレース・ファイル 153

A

Adapter for FTP
 管理 135
Adapter for FTP モジュール
 開始 146
 停止 146
 EAR ファイルとしてのエクスポート 131
 EAR ファイルのサーバーへのインストール 132
Append 4
ArchiveDirectoryForRetrieve 192

C

CEI (Common Event Infrastructure) 150
COBOL コピーブック・ファイル、ビジネス・オブジェクトからの変換 51
COBOL コピーブック・ファイル、ビジネス・オブジェクトへの変換 61
Common Event Infrastructure (CEI) 150
Create 4
CreateFileIfNotExists 192
CustomerWrapper 3
CustomerWrapper ビジネス・オブジェクト 163
CustomerWrapperBG 3, 163

D

DataConnectionMode 193
Delete 4
DeleteOnRetrieve 193
developerWorks 243
developerWorks リソース、WebSphere Adapters 242
Directory Path 193

E

EAR ファイル
 エクスポート 131
 サーバーへのインストール 132
EAR ファイルとしてのモジュールのエクスポート 131
EAR ファイルのインストール 132
EmbeddedNameFunctionSelector 12
enableHASupport プロパティ 34
ExecuteFTPScript 4
Exists 4

F

FFDC (First Failure Data Capture) 158
FileContentEncoding 194
FileInLocalDirectory 194
Filename 194
FilenameFunctionSelector 12
FileTransferType 195

First Failure Data Capture (FFDC) 158
FTPFile 3
FTPFileBG 3
FTPFileBG ビジネス・オブジェクト 163

G

GenerateUniqueFile 195

I

IBM WebSphere Adapter Toolkit 243
ID 172, 203
Inbound イベント処理 10
Inbound 構成プロパティ 200
Inbound 処理 10
IncludeEndBODelimiter 196

J

J2EE コンポーネント 92
Java 2 セキュリティー 30
Java 実装環境 126

L

List 4
LocalArchiveDirForCreate 196
LocalArchivingEnabledForCreate 197
LocalDirectoryPath 197

M

Managed (J2C) Connection Factory プロパティ 178
管理コンソールでの設定 137, 142

O

org.xml.sax.SAXParseException 160
Outbound 構成プロパティ 169
Outbound 処理 3
Overwrite 4

P

Performance Monitoring Infrastructure (PMI)
構成 148
説明 147
パフォーマンスに関する統計の表示 151
PMI (Performance Monitoring Infrastructure)
構成 148
説明 147
パフォーマンスに関する統計の表示 151

R

RAR (リソース・アダプター・アーカイブ) ファイル
サーバーへのインストール 129
説明 129
Redbooks、WebSphere Adapters 242
Retrieve 4

S

ScriptFileParameters 198
SecondServerDirectory 192
SecondServerHostName 195
SecondServerPassword 198
SecondServerPortNumber 197
SecondServerProtocol 198
SecondServerUsername 198
Secure Sockets Layer (SSL) 24
ServerToServerFileTransfer 4
SFTP 26
SplitByDelimiter 15
SplitBySize 15
SplitCriteria 14, 15, 199
SplittingFunctionClassName 199
SSH over FTP 26
SSL 通信 23
StagingDirectory 200
SystemOut.log ファイル 156

T

trace.log ファイル 156

W

WebSphere Adapters バージョン 6.0 情報 243
WebSphere Adapters、バージョン 6.0.2.x、 243
WebSphere Application Server 環境変数 21
WebSphere Application Server 情報 243
WebSphere Business Integration Adapters 情報 243
WebSphere Business Process Management、バージョン
6.1.x、 243
WebSphere Enterprise Service Bus
情報 243
WebSphere Extended Deployment 34
WebSphere Integration Developer
開始 47, 48, 83
情報 243
テスト環境 125
WebSphere Process Server
情報 243
WebSphere Process Server または WebSphere Enterprise Service
Bus
デプロイ先 129

X

XAResourceNotAvailableException 160

xsd ファイル 163



Printed in Japan