





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、225 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 Flat Files 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。

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

原典： WebSphere® Adapters
Version 6 Release 2
WebSphere Adapter for Flat Files 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 Flat

Files の概要	1
ハードウェア要件とソフトウェア要件	1
Adapter for Flat Files の技術概要	2
Outbound 処理	3
Inbound 処理	15
ビジネス・オブジェクト	27
WebSphere Application Server 環境変数	28
外部サービス・ウィザード	29

第 2 章 アダプター実装の計画 31

始める前に	31
セキュリティー	31
ログおよびトレース・ファイル内の機密ユーザー・データの保護に対するサポート	31
デプロイメント・オプション	32
クラスター環境での WebSphere Adapter	35
バージョン 6.2 へのマイグレーション	37
マイグレーションに関する考慮事項	37
マイグレーションの実行	39
バージョン 6.0.2.x またはバージョン 6.1.x プロジェクトをマイグレーションせずに更新する	41
WebSphere Adapter のバージョン 6.2 とともに使用するための、WebSphere ビジネス・インテグレーション・アプリケーションのマイグレーション	42
WebSphere InterChange Server からアプリケーションをマイグレーションするためのロードマップ	42
WebSphere ビジネス・インテグレーション・アダプターのマイグレーションに関する考慮事項	44
アプリケーション成果物の WebSphere InterChange Server からのマイグレーション	44
アダプター固有の成果物のマイグレーション	45
マイグレーション後のインポート、エクスポート、および WSDL ファイルの変更	48

第 3 章 サンプルとチュートリアル 51

第 4 章 デプロイメントのためのモジュールの構成 53

モジュールの構成のためのロードマップ	53
必須のローカル・フォルダーの作成	55
モジュールの作成	56
WebSphere Application Server 環境変数の定義	58
ビジネス・オブジェクトの定義	62
Outbound 処理時のビジネス・オブジェクトから COBOL コピーブック・ファイルへの変換	64
Inbound 処理時の COBOL コピーブック・ファイルからビジネス・オブジェクトへの変換	73
アダプター・パターン・ウィザードを使用した単純なサービスの作成	82

プロジェクトの作成	87
Outbound 処理のモジュールの構成	89
デプロイメントおよびランタイム・プロパティの設定	90
操作およびデータ・タイプの選択	92
データ・バインディングの構成	95
データ・ハンドラーの構成	96
対話プロパティの設定およびサービスの生成	102
Inbound 処理のモジュールの構成	105
デプロイメントおよびランタイム・プロパティの設定	105
操作およびデータ・タイプの選択	112
データ・バインディングの構成	114
データ・ハンドラーの構成	116
デプロイメント・プロパティの設定およびサービスの生成	119

第 5 章 アセンブリー・エディターの使用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
テストするモジュールのデプロイ	125
Inbound 処理のテスト用ターゲット・コンポーネントの生成および配線	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
実動用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	138
組み込みアダプターの活動化仕様プロパティの設定	140
スタンドアロン・アダプターの構成プロパティの変更	142
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	142
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	144

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	145
アダプターを使用するアプリケーションの開始	147
アダプターを使用するアプリケーションの停止	148
Performance Monitoring Infrastructure を使用したパ フォーマンスのモニター	148
Performance Monitoring Infrastructure の構成	149
パフォーマンスに関する統計の表示	151
Common Event Infrastructure (CEI) によるトレース の使用可能化	152

第 8 章 トラブルシューティングおよび サポート **155**

Log and Trace Analyzer のサポート	155
ロギングおよびトレースの構成	156
ロギング・プロパティの構成	156
ログ・ファイル名およびトレース・ファイル名の 変更	158
First Failure Data Capture (FFDC) のサポート	159
ビジネス・フォールト	160
XAResourceNotAvailableException	161
org.xml.sax.SAXParseException	162
セルフ・ヘルプ・リソース	162

第 9 章 参照情報 **165**

ビジネス・オブジェクト情報 165

ビジネス・オブジェクトの構造	165
属性プロパティ	168
命名規則	169
カスタム・ファイル分割	170
Outbound 構成プロパティ	171
ウィザードの接続プロパティ	173
Managed Connection Factory プロパティ	177
リソース・アダプター・プロパティ	183
対話仕様プロパティ	186
Inbound 構成プロパティ	192
ウィザードの接続プロパティ	194
活動化仕様プロパティ	198
リソース・アダプター・プロパティ	216
グローバリゼーション	218
グローバリゼーションおよび双方向データ変換	218
ビジネス・オブジェクト内の双方向変換	220
双方向データ変換で使用可能なプロパティ	222
アダプター・メッセージ	223
関連情報	223

特記事項 **225**

プログラミング・インターフェース情報	227
商標	227

索引 **229**

第 1 章 WebSphere Adapter for Flat Files の概要

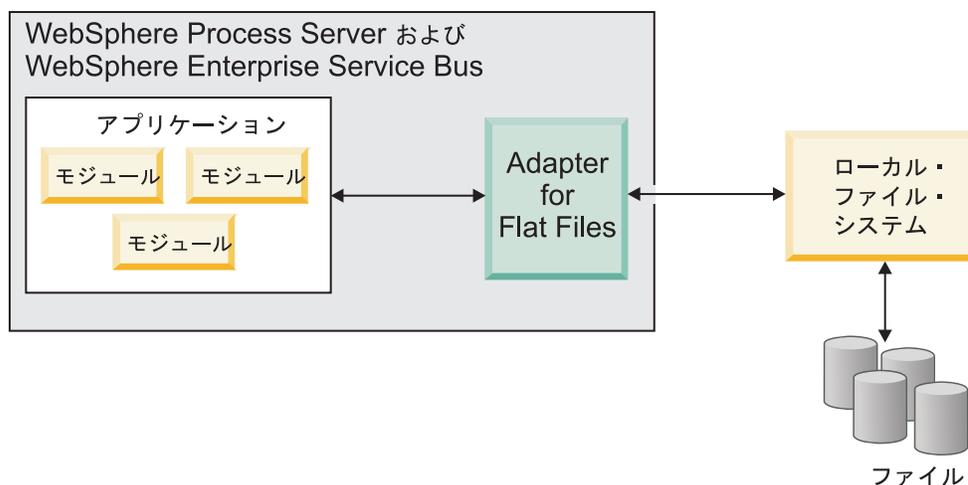
WebSphere Adapter for Flat Files を使用すれば、特別なコーディングを行わなくても、ローカル・ファイル・システムとのデータ交換を含む統合プロセスを作成することができます。

アダプターを使用してローカル・ファイル・システムのファイルからデータを読み取り、そのデータを WebSphere Process Server または WebSphere Enterprise Service Bus で実行中のアプリケーションで使用してから、ローカル・ファイル・システムに送信して戻すことができます。また、アダプターを使用してローカル・ファイル・システムのディレクトリーをポーリングして新しいファイルを探し、それらをアプリケーションに送信して処理することもできます。

アダプターを使用すると、ローカル・ファイル・システムに保管されているすべてのタイプのファイルを読み書きすることができます。実行可能な操作は以下のとおりです。

- 新規ファイルの作成
- 既存のファイルへの付加または上書き
- 特定のファイル内容の検索、ディレクトリー内ファイル名リストの検索、またはファイルの削除
- 特定のファイルが存在するかどうかの確認
- 新規ファイルを探すためのディレクトリーのポーリング、および処理するためのアプリケーションへのファイルの送信

以下の図は、SOA の実装の一部としてアダプターを示したものです。



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 のハードウェアおよび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 のハードウェアおよび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

以下のリンクでは、アダプターの構成およびデプロイに必要なとある場合がある追加情報を参照できます。

- WebSphere Business Integration Adapters および WebSphere Adapters の互換性マトリックスは、アダプターに必要なソフトウェアのサポートされるバージョンを識別します。この資料を表示するには、WebSphere Adapters のサポート・ページ (<http://www.ibm.com/software/integration/wbiadapters/support/>) にアクセスして、「**Additional support links**」セクションの「**Related**」の見出しの下にある「**Compatibility Matrix**」をクリックしてください。
- WebSphere Adapters のテクニカル・ノートには、製品資料に含まれていない予備手段および追加情報が記載されています。アダプターのテクニカル・ノートを表示するには、Web ページ <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**」リストから、ご使用のアダプターの名前を選択し、検索アイコンをクリックしてください。

Adapter for Flat Files の技術概要

IBM WebSphere Adapter for Flat Files を使用すると、WebSphere Process Server または WebSphere Enterprise Service Bus で実行されるサービスが、ローカル・ファイル・システムとの間でデータを交換できるようになります。

サービスは、アダプターを使用して、以下の 2 つの方法でローカル・ファイル・システムとデータを交換できます。

- *Outbound 処理* により、WebSphere Process Server または WebSphere Enterprise Service Bus で実行されるサービスは、アダプターを使用してローカル・ファイル・システムにあるファイルに対して操作 (例えば、注文文書の更新) を実行します。
- *Inbound 処理* により、WebSphere Process Server または WebSphere Enterprise Service Bus で実行されるサービスは、アダプターを使用してローカル・ファイル・システムからイベント (例えば、顧客レポートが更新されたことの通知) を受け取ります。

WebSphere Integration Developer で起動する外部サービス・ウィザードを使用して、この処理を行うようにアダプターを構成します。外部サービス・ウィザードを使用して、モジュールを作成します。モジュールは、WebSphere Integration Developer のプロジェクトと、WebSphere Process Server または WebSphere Enterprise Service Bus に対するデプロイメントの単位で構成されています。各モジュールには、サービスを構成するコンポーネントと、インポート または エクスポート が含まれています。

- インポート は、SCA モジュールが外部サービス (SCA モジュール外のサービス) に、ローカル・アクセスを行うようにアクセスするポイントです。インポートは、SCA モジュールとサービス・プロバイダーの間の対話を定義します。インポートには、バインディングおよび 1 つ以上のインターフェースがあります。

- エクスポート (エンドポイントとも言う) は、外部にビジネス・サービスを提供する Service Component Architecture (SCA) モジュールの公開インターフェースです。エクスポートには、サービス要求元からのサービス (Web サービスなど) へのアクセス方法を定義するバインディングが含まれます。

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 または WebSphere Enterprise Service Bus にデプロイされます。

モジュールとローカル・ファイル・システムの間で交換されるファイルを表すため、アダプターはビジネス・オブジェクトを使用します。ビジネス・オブジェクトは、アダプターで処理されるデータを含む論理データ・コンテナです。ビジネス・オブジェクトを作成するには、WebSphere Integration Developer で、外部サービス・ウィザードを使用するか、ビジネス・オブジェクト・エディターを使用します。

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、Inbound および Outbound の処理中に、ある形式から別の形式にデータを変換します。データ・バインディング は基本的に、ビジネス・オブジェクトのフォーマット方法を定義するマップです。データ・バインディングにより、ビジネス・オブジェクト内のフィールドが読み取られ、ファイル内の対応するフィールドが記入されます。使用されるデータ・バインディングは、ファイルの内部形式に応じて異なります。データ・タイプにはそれぞれ同等のデータ・バインディングがあります。この 外部サービス・ウィザード を使用して、データ・バインディングを構成します。

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形式の間の変換を実行します。ビジネス・オブジェクトを含むデータ・タイプを選択する場合は、変換を行うデータ・ハンドラーを指定する必要があります。データ・ハンドラーは、WebSphere Process Server または WebSphere Enterprise Service Bus によって提供されます。

Outbound 処理

アダプターは、Outbound 処理中に、ローカル・ファイル・システム 内のファイル上で処理を実行するために、ビジネス・オブジェクトの形式でモジュールから要求を受け取ります。アダプターは要求された操作を実行して、該当する場合は、操作の結果を表すビジネス・オブジェクトをコンポーネントに返します。

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

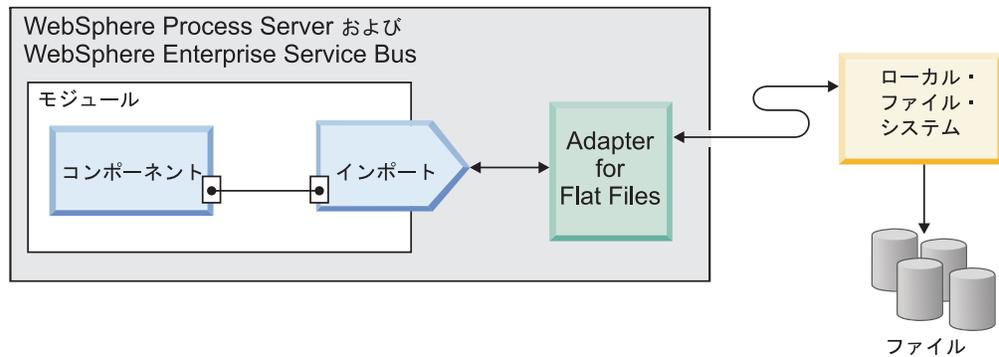


図 1. Outbound 処理

Outbound 操作

操作は、Outbound 処理時にアダプターがローカル・ファイル・システムで実行可能なアクションです。操作名は通常、アダプターが実行するアクションのタイプを示します。

このアダプターは、Outbound 処理時の以下の操作をサポートします。

Append 操作:

Append 操作によって、指定したファイルにコンテンツが追加されます。Append 操作中に新規ファイルが作成されます。

外部サービス・ウィザードで「操作の応答タイプを有効にする」を選択した場合、ファイル名はビジネス・オブジェクトのコンポーネントに返されます。

CreateFileIfNotExists プロパティを true に設定すると、アダプターは新規ファイルを作成します。GenerateUniqueFile プロパティを true に設定すると、アダプターは固有ファイルを生成し、ファイル名プロパティの値を無視します。

注: GenerateUniuFile プロパティは非推奨になりました。現行でこのプロパティを設定することはできませんが、アダプターでは、常にこのプロパティの値を false として処理します。

付加するファイルが存在せず、CreateFileIfNotExists プロパティが false に設定されている場合、アダプターは RecordNotFoundException エラーを生成します。

Filename プロパティに値がないと、アダプターは MissingDataException エラーを生成します。

注: wrapper ビジネス・オブジェクトの場合、ラッパーの CreateFileIfNotExists プロパティに値が設定されていないと、対話仕様プロパティに設定されている値が使用されます。

Create 操作:

Create 操作によって、指定した名前のファイルが作成されます。異なるプロパティを指定して、作成したファイル名を変更できます。例えば、ファイルにシーケンス番号を付加できます。

外部サービス・ウィザードで「操作の応答タイプを有効にする」を選択した場合、ファイル名はビジネス・オブジェクトのコンポーネントに返されます。指定した名前のファイルが既に存在している場合、アダプターは DuplicateRecordException エラーを生成します。ファイルは作成されません。

GenerateUniqueFile プロパティを true に設定すると、アダプターは固有の名前でファイルを生成し、ファイル名プロパティの値を無視します。アダプターによって生成される固有ファイルの名前は、ビジネス・オブジェクト名のプレフィックスに乱数が付き、ファイル拡張子を .tmp にした形式となります。例: Customer23423.tmp。

注: wrapper ビジネス・オブジェクトの場合、ラッパーの GenerateUniqueFile プロパティに値が設定されていないと、対話仕様プロパティに設定されている値が使用されます。

FileSequenceLog 管理接続プロパティを指定すると、アダプターは、シーケンス・ファイルで指定された出力ファイル名にシーケンス番号を付加します。例えば、要求内の出力ファイル名が Customer.txt の場合は、Customern.txt という名前のファイルが作成されます。n は特定の要求のシーケンス番号で、1 から始まります。出力ファイル名が Order.txt となっている別の要求を受け取ると、シーケンス番号が 1 増分され、Order.2.txt が生成されます。個別のファイル名ごとに新たなシーケンス番号が作成されるわけではありません。出力ファイル名に拡張子がない場合は、ファイル名の末尾にシーケンスが付加されます。例えば、要求内の出力ファイル名が Customer の場合は、Customern という名前のファイルが作成されます。

要求ごとにビジネス・オブジェクトで出力ディレクトリーとファイル名を設定する手間を省くため、管理接続レベルで出力ディレクトリーとファイル名を設定することにより、特定タイプの要求に対するファイル・シーケンスを生成できます。アダプターはファイルを作成する要求を受け取ると、ファイル・シーケンス・ログを検査し、その名前を持つファイルが既に存在するかどうかを検査します。既に存在する場合、アダプターはファイル・シーケンス番号を使用して新規ファイル名を作成します。

注: ビジネス・オブジェクトで指定されるディレクトリー・パスとファイル名は、管理接続プロパティの値よりも優先されます。

クラスター環境 (複数のシステム上で稼働するアダプターのインスタンスが 1 つ存在する環境) では、FileSequenceLog プロパティによって指定されるシーケンス・ファイルが、クラスター内のすべてのノードからアクセス可能なマップされたドライブに存在する必要があります。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許可が必要です。この許可がない場合は、IOException エラーが返されます。

FileSequenceLog プロパティが指定されており、GenerateUniqueFile プロパティが有効になっている場合は、GenerateUniqueFile プロパティが FileSequenceLog プロパティより優先されます。シーケンス番号は、アダプターの再始動後まで増分

され続けます。シーケンス・ファイルを手動で削除すると、順序付けが 1 から再開されます。シーケンスは、シーケンス・ファイル内のシーケンス値を変更することによってリセットできます。

Delete 操作:

Delete 操作によって、指定したファイルが削除されます。

Delete

オプションで、ビジネス・オブジェクトのコンポーネントに delete 操作の出力を返すようにすることができます。外部サービス・ウィザード で「**操作の応答タイプを有効にする**」チェック・ボックスを選択した場合、アダプターは、ファイルが正常に削除されると true を返します。delete のファイル・アクセス権がない場合、アダプターは false を返します。

そのファイルが存在しない場合、アダプターは RecordNotFoundException エラーを生成します。

Exists 操作:

Exists 操作によって、指定されたファイルが存在するかどうかを検査されます。

Exists

指定されたファイルが存在する場合は、成功応答がビジネス・オブジェクトの形式でコンポーネントに返されます。ビジネス・オブジェクトには、ファイルが存在する場合には true に、ファイルが存在しない場合には false に設定される 1 つの属性があります。ファイルが存在しない場合、またはディレクトリーが存在しない場合、アダプターは false を返します。

List 操作:

List 操作によって、指定されたディレクトリー内のファイル名がリストされます。

List

そのディレクトリーが存在しない場合、アダプターは RecordNotFoundException エラーを生成します。

Overwrite 操作:

Overwrite 操作によって、指定されたファイルは要求で指定された内容で上書きされます。

外部サービス・ウィザード で「**操作の応答タイプを有効にする**」を選択した場合、ファイル名はビジネス・オブジェクトのコンポーネントに返されます。ステージング・ディレクトリーが StagingDirectory プロパティーで指定されている場合、上書きされるファイルは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、内容はステージング・ディレクトリーにあるファイルで上書きされます。その後、出力ディレクトリーにファイルが戻されます。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、出力ディレクトリーにあるファイルで内容が上書きされます。

注: ステージング・ディレクトリーは、Overwrite 操作が結果の値を返す前にそのファイル内容が書き込まれる場合にのみ構成できます。Overwrite 操作が出力ストリームを返し、コンポーネントがこのストリームに書き込む場合は、ステージング・ディレクトリーを使用できません。

Input 要求を FlatFileOutputStreamRecord レコードとして受け取ると、アダプターは出力ストリームを返します。

CreateIfFileNotExists プロパティーを true に設定すると、アダプターは新規ファイルを作成します。GenerateUniueFile プロパティーは非推奨になりました。現行でこのプロパティーを設定することはできますが、アダプターでは、常にこのプロパティーの値を false として処理します。

更新するファイルが存在しないときに、CreateFileIfNotExists プロパティーを false に設定すると、アダプターは RecordNotFoundException エラーを生成します。

注: wrapper ビジネス・オブジェクトの場合、ラッパーの CreateFileIfNotExists プロパティーに値が設定されていないと、対話仕様プロパティーに設定されている値が使用されます。

Retrieve 操作:

Retrieve 操作によって、指定されたファイルのコンテンツが取り出され、ビジネス・オブジェクトの形式で返されます。Outbound 処理中に、Retrieve 操作中に返されたファイルを削除およびアーカイブすることもできます。

Retrieve 操作中に Retrieve 要求で指定したファイルのコンテンツが取り出されて、汎用またはコンテンツ固有のビジネス・オブジェクトの形式で返されます。アダプターはファイル分割機能を使用して、大容量ファイルをもっと小さなチャンクに分割し、それらが別々に取り出されます。ファイルのコンテンツは、対話仕様で定義される SplittingFunctionClassName プロパティーおよび SplitCriteria プロパティーに従って分割されます。これらのプロパティーには、アダプターがファイル・システムとインターフェースをとるために使用する Outbound 接続プロパティーが含まれています。データ・ハンドラーが構成されている場合、アダプターはコンテンツ固有のビジネス・オブジェクトを返します。それ以外の場合は、汎用ビジネス・オブジェクトを返します。

ファイルを取り出した後に元のファイルを削除するには、対話仕様で DeleteOnRetrieve プロパティーを設定します。ファイルをアーカイブしてから削除するには、ArchiveDirectoryForDeleteOnRetrieve プロパティーを設定します。

Retrieve 操作中に Retrieve 要求で指定したファイルが存在しないと、アダプターは RecordNotFoundException エラーを生成します。

注: wrapper ビジネス・オブジェクトの場合、ラッパーの DeleteOnRetrieve プロパティーに値が設定されていないと、対話仕様プロパティーに設定されている値が使用されます。

Outbound データ変換

Outbound 処理中、アダプターは、外部サービス・ウィザードでのアダプターの Outbound 処理の構成時に選択したアダプター固有のデータ・バインディングおよびデータ・ハンドラーに基づいて、データ変換を行います。

データ変換を伴う Outbound 処理

Outbound 処理中、アダプターは、アプリケーションで予測されるデータ形式にビジネス・オブジェクトを変換します。この処理は、Outbound 処理用にモジュールを構成した際に選択したアダプター固有のデータ・バインディングおよびデータ・ハンドラーによって制御されます。

図 2 は、Outbound 処理中にデータが変換される方法を示したものです。

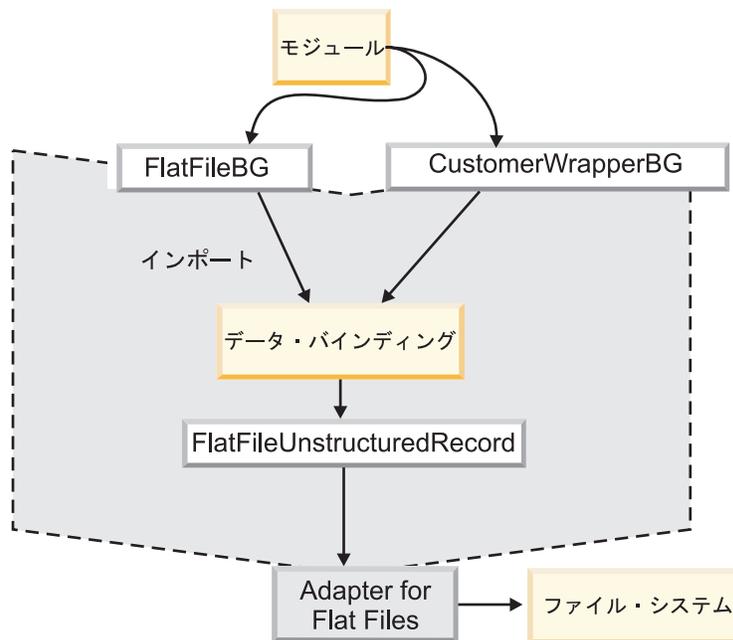


図 2. Outbound 処理中のデータ変換

以下のステップで、データ変換を伴う Outbound 処理について説明します。

1. Retrieve 以外のすべての操作では、アダプターは、入力タイプおよび構成済みデータ・ハンドラーに基づいて、データ変換を行います。入力が汎用タイプ (FlatFile または FlatFileBG) 以外の場合、アダプターはデータを変換します。Retrieve 操作では、アダプターは、データ・バインディングのデータ・ハンドラー・プロパティーが構成済みである場合にのみ、データを変換します。
2. 構成済みのデータ・バインディングが呼び出され、ビジネス・オブジェクトを処理します。
3. データ・バインディングによって、データ・バインディング・プロパティーのデータ・ハンドラー・プロパティーに対して指定された値が検査され、データ・ハンドラー・プロパティーに設定された値に基づいて、コンテンツ固有のデータ・ハンドラーが呼び出されます。

4. アダプターは、ファイルに対して要求された操作を行い、以下のような応答ビジネス・オブジェクトを返す場合があります。
- Create、Append、および Overwrite 操作では、出力が構成済みである場合、応答ビジネス・オブジェクトにはファイル名が含まれます。
 - List 操作では、応答ビジネス・オブジェクトには指定されたディレクトリー内のファイル・リストが含まれます。
 - Exists 操作では、応答ビジネス・オブジェクトには true または false のいずれかの値が含まれます。
 - Retrieve 操作では、取り出されたファイルのコンテンツが汎用またはコンテンツ固有の応答ビジネス・オブジェクトの形式で返されます。
 - Delete 操作では、出力が構成済みであると、応答ビジネス・オブジェクトに true または false のいずれかの値が含まれます。

データ変換を伴わない Outbound 処理

Retrieve 以外のすべての操作では、入力データが汎用タイプ (FlatFile または FlatFileBG) の場合、アダプターはデータ変換を伴わない Outbound 処理を行います。Retrieve 操作では、データ・バイndィングのデータ・ハンドラー・プロパティに値が設定されていない場合、データ変換は実行されません。このタイプの処理中は、UnstructuredContent という特殊なデータ構造を使用してコンテンツが保持されます。

10 ページの図 3 はデータ変換を伴わない Outbound 処理を示したものです。

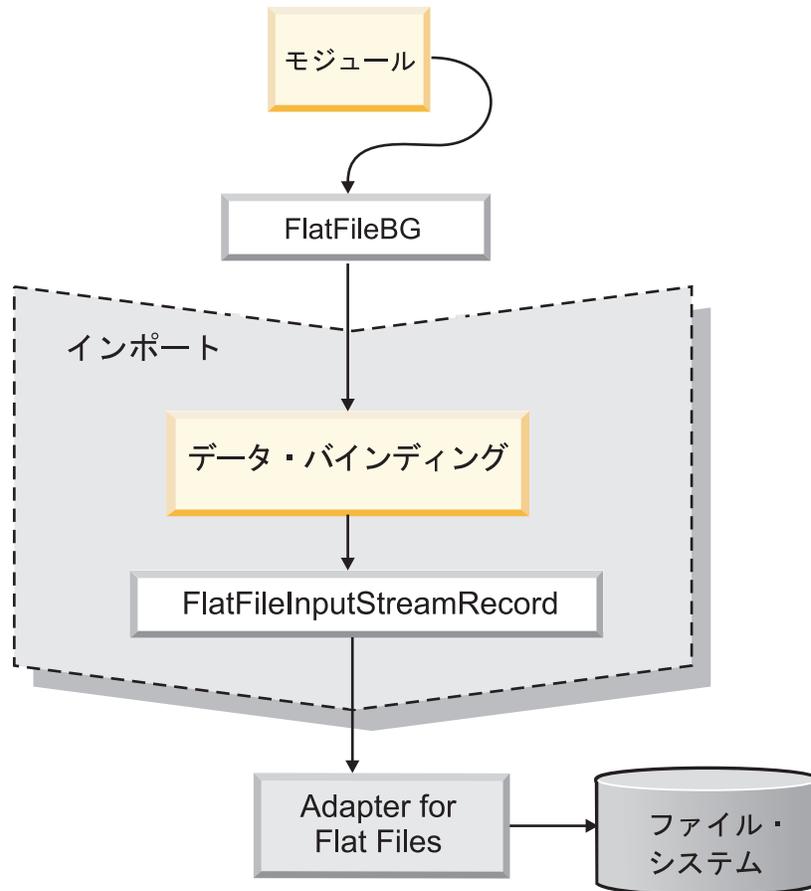


図3. データ変換を伴わない Outbound 処理

以下のステップで、データ変換を伴わない Outbound 処理について説明します。

1. Retrieve 以外のすべての操作では、アダプターは要求データ・オブジェクトの入力タイプを検査します。入力が汎用タイプ (FlatFile または FlatFileBG) の場合、アダプターは着信オブジェクトに対してデータ変換を行いません。Retrieve 操作では、アダプターはデータ・ハンドラー・プロパティーを検査します。値を指定しなかった場合、データ変換が行われます。
2. 構成済みのデータ・バインディングが呼び出され、ビジネス・オブジェクトを処理します。
3. Retrieve 操作では、アダプターはデータ・ハンドラー・プロパティーを検査します。データ・ハンドラーに値が設定されていない場合は、アダプターはデータ変換を行いません。
4. アダプターは、ファイルに対して要求された操作を行い、以下のような応答ビジネス・オブジェクトを返す場合があります。
 - Create、Append、および Overwrite 操作では、出力が構成済みである場合、応答ビジネス・オブジェクトにはファイル名が含まれます。
 - List 操作では、応答ビジネス・オブジェクトには指定されたディレクトリー内のファイル・リストが含まれます。
 - Exists 操作では、応答ビジネス・オブジェクトには true または false のいずれかの値が含まれます。

- Retrieve 操作では、取り出されたファイルのコンテンツが汎用またはコンテンツ固有の応答ビジネス・オブジェクトの形式で返されます。
- Delete 操作では、出力が構成済みであると、応答ビジネス・オブジェクトに true または false のいずれかの値が含まれます。

ファイル分割

複数のレコードを含むファイルに対応するため、アダプターではオプションでファイル分割機能が提供されます。この機能を使用する場合、アダプターは大きなファイルを小さいチャンクに分割して、別々に取得します。

ファイルは、内包するコンテンツのタイプに応じて、区切り文字またはサイズで分割できます。

- ビジネス・オブジェクトのコンテンツが明確な構造を持つ場合 (例えば、名前、住所、および市などの要素を持つ場合など) は、ファイルは区切り文字で分割されます。
- ビジネス・オブジェクトに含まれるのがプレーン・テキストまたはバイナリー・ファイルなどの非構造化データの場合、ファイルはサイズによって分割されます。

デフォルトでは、アダプターはファイルをサイズで分割します。

SplitCriteria プロパティーで指定する値によって、使用する方式が決まります。

SplitCriteria プロパティーのデフォルト値がゼロの場合、分割は行われません。分割を必要としない場合は、**SplitCriteria** プロパティーおよび **SplittingFunctionClassName** プロパティーの値を空のままにしておいてもかまいません。

オプションでカスタム・ファイルのスプリッター・クラスを指定できます。クラス名に **SplittingFunctionClassName** プロパティーを設定してください。

区切り文字によるファイル分割

ファイル内のビジネス・オブジェクトを区切るのに、コンマ (,)、セミコロン (;)、引用符 ("、')、中括弧 ({}), またはスラッシュ (/ ¥) などの 1 つ以上の文字 (区切り文字) が使用されるとき、アダプターは区切り文字に基づいて、ファイルを小さいチャンクに分割できます。ファイルのビジネス・オブジェクトを分離する区切り文字は、**SplitCriteria** プロパティーに定義します。

区切り文字によるファイル分割を使用可能にするには、**SplittingFunctionClassName** プロパティーの値を `com.ibm.j2ca.utils.filesplit.SplitByDelimiter` に指定します。

区切り文字の使用には、以下の規則が適用されます。

- 区切り文字内のすべての改行は、プラットフォーム固有の改行文字で示す。プラットフォーム固有の改行文字を表 1 に示します。

表 1.

プラットフォーム	改行文字
Macintosh	¥r
Microsoft® Windows®	¥r¥n
UNIX®	¥n

- 複数の区切り文字がある場合、それぞれの区切り文字はセミコロン (;) で分離する必要があります。区切り文字は、指定された順に突き合わせが行われます。セミコロンが区切り文字の一部である場合、¥; のようにしてエスケープする必要があります。例えば、区切り文字が ###¥;## の場合、##;## として処理されます。
- 区切り文字の一部であるコンテンツをスキップする場合、直前に 2 つのセミコロン (;;) を指定すると、区切り文字の間のコンテンツがスキップされます。例えば、以下の形式のビジネス・オブジェクトがイベント・ファイルに含まれていて、区切り文字が ##;\$\$\$ の場合、アダプターは、###\$\$\$ を区切り文字と見なし、「content skipped by the adapter」の部分をスキップします。

```
Name=Smith
Company=IBM
##content skipped by the adapter$$$
```

- 区切り文字には任意の値を使用することができ、制限はありません。区切り文字は、有効なストリング、改行文字 (¥n など)、および区切り文字が複数ある場合はセミコロンの分離文字を組み合わせたものです。区切り文字に改行文字およびセミコロンが含まれていなくてもかまいません。改行文字が使用されるのは、ファイルのコンテンツを分割するときに改行文字が考慮される場合に限られます。有効な区切り文字の例は以下のとおりです。

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

- 区切り文字がファイルの最後に配置された場合、SplitCriteria プロパティーは END_OF_FILE を使用して、ファイルの物理的な末尾を決定します。

一般的なシナリオの例と推奨される区切り文字の形式を以下に示します。

表 2.

データ・バイインディング	BO コンテンツ	推奨される区切り文字の形式
XML	<pre><?xml version="1.0" encoding="UTF-8"?> <customer:Customer xsi:type="customer:Customer" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:customer="http://www.ibm.com/xmlns/prod/websphere/ j2ca/flatfile/customer"> <CustomerName>Deepa</CustomerName> <Address>IBM</Address> <City>Bangalore</City> <State>KA</State> </customer:Customer></pre>	¥n

サイズによるファイル分割

`SplittingFunctionClassName` プロパティーで指定する値によって、ファイルがサイズで分割されるかどうかが決まります。`SplittingFunctionClassName` プロパティーを `com.ibm.j2ca.extension.utils.filesplit.SplitBySize` に設定した場合、`SplitCriteria` プロパティーには最大ファイル・サイズをバイト単位で表す有効な数字が含まれている必要があります。ファイルが `SplitCriteria` プロパティーで指定した値よりも大きい場合、ファイルはチャンクに分割され、それぞれのチャンクはインポートに別々に送られます。ファイルが `SplitCriteria` の値よりも小さい場合、ファイル全体がインポートに送られます。

イベント・ファイルがチャンクに分割されると、それぞれのチャンクがビジネス・オブジェクトとなります。すなわち、`PollQuantity` プロパティーに指定した値と、インポートに送達したビジネス・オブジェクトの数が、異なってもかまいません。アダプターは `PollQuantity` の値に従ってポーリングを行います。実際にはファイル内のビジネス・オブジェクトは一度に 1 つずつ処理します。例えば、イベント・ファイルが 3 つのチャンクに分けられた場合、1 つのファイルがポーリングされて 3 つのビジネス・オブジェクトがインポートに送達されます (それぞれのチャンクが 1 つのビジネス・オブジェクトを作成しているため)。

インポートでは、アダプターはチャンク・データを単一ファイルに再組み立てすることはしませんが、WebSphere Process Server または WebSphere Enterprise Service Bus でこれらを単一ファイルに再組み立てできるように、チャンクについての情報を提供します。チャンク情報は `FlatFileInputStreamRecord` レコードの `ChunkFileName` プロパティーに組み込まれ、バイト単位のチャンク・サイズおよびイベント ID が含まれます。チャンクのイベント ID は、`eventFileLocation/_timestampStr/_MofN` の形式を使用します (M は現在のチャンク番号、N は総チャンク数)。イベント ID の例は、

`C:%flatfile%eventdir%eventfile.in/_2005_01_10_10_17_49_864/_3of5` のようになります (timestampStr は `year_month_day_hour_minutes_seconds_milliseconds` の形式になります)。

固有ファイル名の生成

`Create` 操作中に固有ファイル名を生成するには、永続シーケンス番号をデフォルトのファイル名に追加するか、乱数を使用してファイル名を生成します。 `Append` および `Overwrite` 操作中は、乱数を用いた方法を使用してください。

`Create` 操作中に固有ファイル名を生成する方法は、次の 2 とおりです。

1. 永続シーケンス番号をデフォルト・ファイル名に追加する。このメソッドは特にクラスター化された環境の場合にお勧めします。
2. 乱数を使用して、永続的ではない固有ファイル名を生成する。

`Append` および `Overwrite` 操作の場合は、乱数メソッドを使用してください。

永続シーケンス番号を使用する固有ファイル名の生成

永続シーケンス番号を使用して固有ファイル名を生成するには、以下を指定します。

- シーケンス・ファイル。シーケンス番号を保管するファイルの完全パスです。

- デフォルト・ターゲット・ファイル名

アダプターは、デフォルトのターゲット・ファイル名にシーケンス番号が追加された構成のファイル名を生成します。

固有ファイル名の生成を制御するプロパティは、次の 3 つの場所にあります。

- **Managed Connection Factory** プロパティ (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)
- **対話仕様** プロパティ (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)
- **ラッパー・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は対話仕様のプロパティよりも優先され、対話仕様のプロパティは **Managed Connection Factory** プロパティよりも優先されます。個々のビジネス・オブジェクトにプロパティを指定する場合を除き、**Managed Connection Factory** でプロパティを使用して、ファイル名の生成を制御してください。

デフォルト・ファイル名に拡張子が付いている場合、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** 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。

要求があるたびに、アダプターはシーケンス・ファイル内の番号を増分し、入力タイプがシーケンス・ファイルに現在保管されているシーケンス番号を取得します。シーケンス番号は、異なる入力データ・タイプに対して別個に保持されていません。

入力データ・タイプごとに別々のシーケンス番号を保持していた、前のバージョンのアダプターで生成されたシーケンス・ファイルとの互換性を保つために、アダプターは以前のフォーマット (`<dirPath>/Customer.txt = 2`。ここで、`Customer.txt` はデフォルト・ファイル名を表し、`2` はアダプターが同一ファイルに対する別の **Create** 要求を受け取った場合に使用されるシーケンス番号を表す) を持つファイル内のすべてのエントリーをチェックします。アダプターは、入力タイプごとにこのようなすべてのシーケンス番号を検索し、最も大きいシーケンス番号を次の入力タイプのシーケンス番号として使用します。次に、アダプターは新しい (増分した) シーケンス番号でファイル全体を上書きします。

重要: それらがクラスターの一部でない場合、2 つのアダプター・インスタンスは同じシーケンス・ファイルにアクセスしてはなりません。これにより、バッチ要求の処理が遅延する可能性があるためです。

乱数を使用する固有ファイル名の生成

乱数を使用して固有ファイル名を生成するには、対話仕様で「固有ファイルの生成 (**GenerateUniqueFile**)」プロパティを設定するか、ビジネス・オブジェクトを `true` に設定します。アダプターは、`ffa[RandomNumber].tmp` という形式で固有ファイル名を生成します。ここで、`RandomNumber` はアダプターが生成した乱数を表します。例えば、`ffa23423.tmp` のようになります。

Inbound 処理

Adapter for Flat Files は Inbound イベント処理をサポートします。これは、イベント（ファイルの作成や変更など）に指定された間隔で、ローカル・ファイル・システムのポーリングを行います。イベントを検出すると、イベント・データをビジネス・オブジェクトに変換し、これをモジュールに送信して処理します。

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

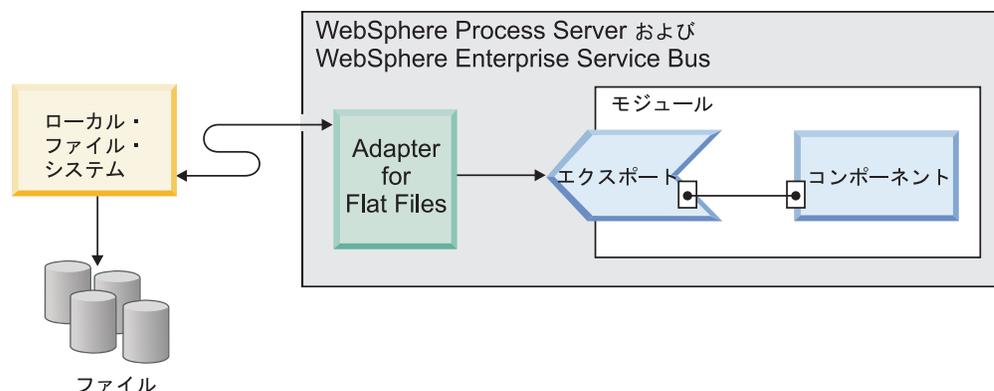


図4. Inbound 処理

ローカル・ファイル・システムで変更が生じた場合、新規または変更済みファイルであるイベント・ファイルが特定のディレクトリーに作成されます。このディレクトリーをアダプターのイベント・ディレクトリーとして構成します。イベント・ファイルはファイル・システム内の 1 つ以上のイベントを表すことが可能ですが、アダプターへの 1 転送単位のみを形成します。

アダプターは、ファイル・システムのイベント・ディレクトリーを定期的な間隔でポーリングします。この間隔は、PollPeriod プロパティーで設定した値に基づいています。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の内容をエクスポートに送信します。ファイル内容は一括で送信されるか、いくつかのビジネス・オブジェクト、あるいはチャンクに分割されます。アダプターは、関数セクターを使用して、ビジネス・オブジェクトをエクスポートに送信します。関数セクターは、コンポーネントで起動する操作を選択し、正しいデータ・バインディングを提供します。

Inbound 処理フローは以下のとおりです。

1. イベントは、ファイルの形式でファイル・システム内に生成されます。
2. アダプターが、イベント・ディレクトリーをポーリングします。
3. アダプターが、各イベントにイベント ID を割り当て、そのイベント ID をイベント・ストアに保管します。イベント・ストアは、ポーリング・アダプターがイベント・レコードを処理できるようになるまでイベント・レコードが保存されるパーシスタント・キャッシュです。アダプターを構成する前に、このデータベースを作成する必要があります。データベースのデフォルト名は FFDB です。

- アダプターは各イベント・ファイルをバイトとして読み取ります。ファイル分割が有効な場合、アダプターは `SplittingFunctionClassName` および `SplitCriteria` プロパティーに設定された値に基づいて、イベント・ファイルを解析します。
 - 分割が区切り文字に基づく場合、この機能を実行するクラスと分割基準が提供されます。
 - 分割がファイル・サイズに基づく場合、この機能を実行するクラス名が提供されます。
- 構成されたデータ・タイプがオブジェクト固有 (`CustomerWrapper` など) である場合、データ・ハンドラーが `DataBinding` に構成され、アダプターがそのデータを変換します。構成されたデータ・タイプが `FlatFile` または `FlatFileBG` である場合、アダプターはファイルの内容を `FlatFile` ビジネス・オブジェクト内のバイト配列として渡し、変換は行われません。

注: ファイル分割が有効な場合、ビジネス・オブジェクトにはファイル・サイズおよびイベント ID が含まれます。

- アダプターは、関数セクターを使用して、ビジネス・オブジェクトをエクスポートに送信します。関数セクターは、コンポーネントで起動する操作を選択し、正しいデータ・バインディングを提供します。ビジネス・オブジェクトがエクスポートに到達すると、イベントはイベント・ストアから削除されます。アーカイブ機能が有効な場合は、イベントは削除される前にアーカイブ・テーブルに移されます。

サブディレクトリー内のファイルのポーリング

デフォルトでは、アダプターがイベント・ディレクトリー内のファイルをポーリングする際、アダプターはルート・ディレクトリー内のファイルのみをポーリングし、サブディレクトリー内のファイルを無視します。活動化仕様の `PollSubDirectories` プロパティーを `True` に設定すると、アダプターはまずルート・ディレクトリー内のファイルをポーリングし、その後にサブディレクトリー内のファイルをポーリングします。すべてのファイルを取得した後に、アダプターは `SortEventFiles` プロパティーに設定されている値に従ってファイルをソートします。次に、アダプターは `PollQuantity` プロパティーに設定されている値に従ってファイルを処理し、ビジネス・オブジェクトをダウンストリーム・コンポーネントに送信します。

イベント・アーカイブ

正しくポーリングされたイベントを追跡するために、外部サービス・ウィザードの `ArchiveDirectory` 活動化仕様プロパティーを使用して、ファイル・システムにアーカイブ・ディレクトリーを構成できます。ファイルは、活動化仕様に明記されているとおり、`success` 拡張子または `fail` 拡張子のいずれかが付加されてアーカイブ・ディレクトリーにコピーされます。

イベント・ファイル・ロック

ファイル・ロックの動作はオペレーティング・システムに応じて異なります。`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た任意のファイルが他のアプリケーションで使用されており、イベント・ディレクトリーへのコピー処理中である場合は、アダプターでの処理に使用できません。

ただし、AIX® などの UNIX 環境の場合は、アプリケーションが書き込みを受けているファイルにアクセスできなくなるようなファイル・ロック・メカニズムはありません。別のアプリケーションでイベント・ディレクトリーにコピーされるファイルは、アダプターでの処理に使用できますが、正常な結果は得られません。Java™ には、ファイルが書き込みを受けているかどうかを確認する、プラットフォームに依存しない方法はありません。

まずイベント・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、move コマンドを使ってそのファイルをイベント・ディレクトリーに移動することで、この状態を回避できます。UNIX スクリプトのいくつかのサンプルが、アダプターの一部として提供されています。CheckIfFileIsOpen.sh という名前のスクリプト・ファイルを、アダプター・インストーラーの Unix-script-file フォルダーから使用できます。

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
- Directory
- 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、イベント・ファイル・マスクの基準でフィルターに掛けられたファイルに対して適用されます。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3. 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_REGEXP	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
Directory	MATCHES_REGEXP	Java 正規表現	pollSubDirs = true

表3. メタデータのフィルター・プロパティー (続き)

プロパティー	有効な演算子	値	前提条件
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

イベント・パーシスタンス

アダプターは突然終了した場合の Inbound 処理のイベント・パーシスタンスをサポートしています。イベント・パーシスタンス (または 1 回の送達) とは、障害時に、イベントがエクスポートに確実に 1 回だけ送達されるようにする手段です。アダプターは、イベント処理中に、データ・ソースにあるイベント・ストアにイベントの状態を永続させます。このデータ・ソースは、イベント・ストアが作成可能になる前に、WebSphere Process Server または WebSphere Enterprise Service Bus を使用してセットアップする必要があります。WebSphere Process Server または WebSphere Enterprise Service Bus で提供されているリカバリー機能を使用するには、活動化仕様で AssuredOnceDelivery プロパティーを true に設定します。このリカバリー機能はデフォルトでは有効になっています。

また、アダプターはイベント・ストアのメモリー内表記を使用して、イベント・パーシスタンスを提供することもできます。この機能を使用すると、JNDI データ・ソースまたは外部イベント・ストアを作成する必要がなく、イベント処理が高速になります。ただし、この機能ではイベント・リカバリーがサポートされていません。サーバーに障害が発生すると、メモリー内のイベント・ストアは失われます。サーバー障害時にイベントが失われることを回避するために、データベース・イベント・ストアを使用するアプローチをお勧めします。

アダプターのメモリー内イベント・パーシスタンス機能を使用するには、AssuredOnceDelivery プロパティーを false に設定する必要があります。設定しない場合、アダプターは警告メッセージをログに記録します。

関連資料

51 ページの『第 3 章 サンプルとチュートリアル』

ユーザーが、WebSphere Adapters を円滑に使用できるようにするため、

「Business Process Management Samples and Tutorials」Web サイトからサンプルおよびチュートリアルが利用できます。

イベント・ストア

イベント・ストアは、ポーリング・アダプターがイベント・レコードを処理できるようになるまでイベント・レコード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 イベントがシステム内を通過するときに、イベント・ストアを使用してその Inbound イベントを追跡します。ファイルが作成、更新、または削除されるたびに、アダプターはイベント・ストア内のイベントの状況を更新します。イベントがエクスポートに送信されるまでは、各イベントの状況は、リカバリーのためにアダプターによって継続的に更新されます。

Inbound モジュール用のイベント・ストアがローカル・ファイル・システム内に存在しないことを検出すると、アダプターはアプリケーションがランタイムにデプロイされるときに自動的にイベント・ストアを作成します。アダプターによって作成された各イベント・ストアは、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・ストア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。

ローカル・ファイル・システムをポーリングするときに、アダプターは活動化仕様プロパティーで指定した検索条件に一致する各イベントについて、イベント・ストア内にエントリーを作成します。アダプターは、新しい各エントリーの状況を NEW として記録します。

イベントが正常に送られた場合、イベント・ストア項目は削除されます。失敗したイベントの場合、エントリーはイベント・ストアに残されます。オプションとして、アダプターは、正常にポーリングされたイベント・ファイルをアーカイブ・ディレクトリーにアーカイブできます。

注: 失敗したイベントによって、イベント・ファイル内のデータが正確でなくなる場合があります。例えば、fname という名前のフィールドが fnam と示される場合があります。これを訂正するには、正しいデータを含むイベント・ファイルを再送するしかありません。

アダプターは、イベントの送達を 1 回としています。つまり、各イベントは 1 回のみ配信されます。AssuredOnceDelivery 活動化仕様プロパティーを True に設定した場合、アダプターは各イベントの XID (トランザクション ID) の値をイベント・ストアに保管します。イベントが処理のために取得されると、以下の操作が実行されます。

1. イベントの XID 値がイベント・ストア内で更新されます。
2. イベントが対応するエクスポートに送達されます。
3. イベントがイベント・ストアから削除されます。

次の図はアダプター用のイベント管理フローを示しています。

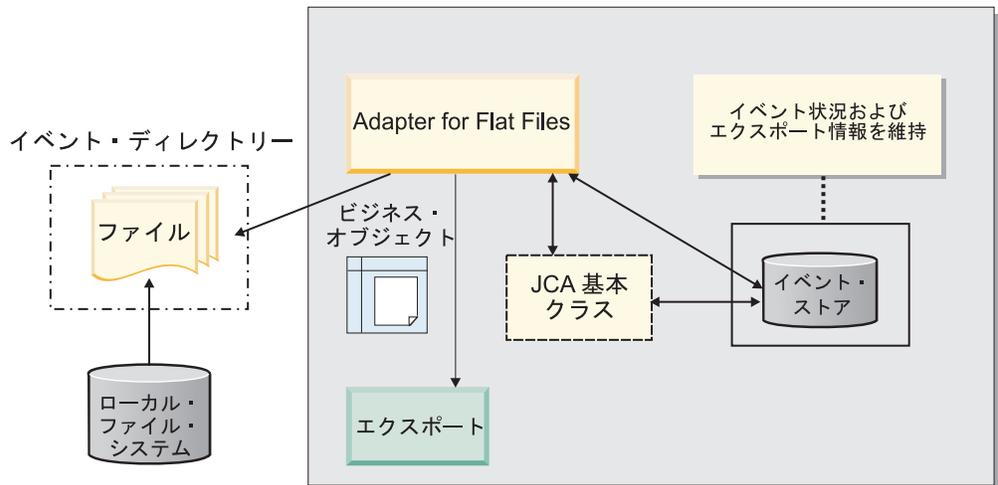


図5. イベント管理フロー

イベント・ストア構造:

イベント・ストアはイベントを追跡するために、アダプターによって使用されます。次の表に、イベントごとに保管される値を示します。

表4. イベント・ストア構造

列名	タイプ (長さ)	説明
EVNTID	Varchar (255)	Inbound 処理中のイベントの追跡に使用されます。各イベントには、トラッキングのため、イベント ID が必要です。これはテーブル内での固有 ID である必要があります。
EVNTSTAT	Integer	<p>イベントの状況。アダプターはこの状況を使用して、イベントが新規か処理中かを判断します。</p> <p>イベント状況の値:</p> <p>NEW(0)</p> <p>イベントの処理の準備ができています。</p> <p>PROCESSED (1)</p> <p>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)</p> <p>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>

表 4. イベント・ストア構造 (続き)

列名	タイプ (長さ)	説明
XID	Varchar (255)	イベント送達およびイベント・リカバリーが行われたかどうかを確認するために、アダプターによって使用されます。
EVNTDATA	Varchar (255)	失敗したイベントを追跡して、これらがリカバリー中に再処理されないようにするために使用されます。失敗したイベントには「ARCHIVED」のマークが付けられます。

イベント・アーカイブ値:

ディレクトリー内の処理済みイベント・ファイルをアーカイブするようにアダプターを構成すると、それにアクセスして処理済みイベントのリストを入手することができます。ファイル拡張子には、アーカイブされたイベントが正常に実行されたかどうか反映されます。

指定されたアーカイブ・ディレクトリー内のアーカイブされたイベントは、すべて success、failure、および original のいずれかのファイル拡張子を付けて保管されます。success 拡張子は、イベント処理が成功した場合に使用されます。イベント処理に失敗した場合は、ファイルは failure 拡張子および original の拡張子付きでアーカイブされます。イベント・ファイルに複数のビジネス・オブジェクトがあり、その一部が正常に処理された場合、そのファイルにも success 拡張子が付けられます。

アーカイブ拡張子は、活動化仕様プロパティー FailedArchiveExt、OriginalArchiveExt、および SuccessArchiveExt に基づいて構成できます。

以下の表は、アダプターが使用するアーカイブ拡張子をリストしたものです。

表 5. イベント・アーカイブ値

拡張子	定義	フォーマット
SUCCESS	イベント・ファイルはエクスポートに送達されました。	<filename>_<timestamp>.SUCCESS
FAIL	イベント・ファイルはエクスポートに送達されませんでした。	<filename>_<timestamp>.FAIL
ORIGINAL	処理するように選択された元のイベント・ファイル。	<filename>_<timestamp>.ORIGINAL

関数セレクター

Inbound 処理中に、関数セレクターはサービスで呼び出される適切な操作を返しません。外部サービス・ウィザードで Inbound 処理用にアダプターを構成した場合は、

関数セクターを選択してください。アダプターは、`FilenameFunctionSelector` および `EmbeddedNameFunctionSelector` の 2 つの関数セクターを提供します。

FilenameFunctionSelector

`FilenameFunctionSelector` は、ファイル名にマップする正規表現に基づくオブジェクトの名前解決を提供する、ルール・ベースの関数セクターです。正規表現は、一定の構文規則に従って、一連のストリングを記述または突き合わせる場合に使用するストリングです。

以下の表に、突き合わせルールの例を示します。このルールは `ObjectName` フィールドおよび `Rule` フィールドで構成されます。

表 6. `FilenameFunctionSelector` の突き合わせルールの例

FileName	ObjectName	Rule
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 行目のルールは同じ名前に解決されていますが、2 行目のルールでは、特定の数値や文字のシーケンスを含んでいることがファイル名が一致していると思なされる条件であるため、比較的短い突き合わせとなります。一方、3 行目のルールでは、ファイル名に「OR」という文字が含まれるすべてのオブジェクト名に解決されます。「.*」という文字の組み合わせは、任意の文字が任意の回数使用される可能性があることを示しています。

固有の関数名を生成する場合、関数セクターは、指定されたオブジェクト名の先頭に `emit` を追加します。例えば、オブジェクト名が `Customer` の場合、関数セクターは `emitCustomer` という関数名を返します。オブジェクト名としては、`Customer` または `Order` のようなペイロード・オブジェクト名を指定する必要があります。ラッパーまたはビジネス・グラフ名は指定できません。パススルーのシナリオの場合は、オブジェクト名として `FlatFile` を使用してください。

それぞれがオブジェクト名とファイル名と突き合わせる正規表現を含む複数のルールを使用して、`FilenameFunctionSelector` を構成することができます。複数のルールが一致する場合、関数セクターは最初に一致したルールに基づいてオブジェクト名を戻します。ルールが一致しない場合は、アダプターはエラーを生成します。構成にルールが存在しない場合、関数セクターは `emitFlatFile` という関数名を使用します。

正規表現の使用に対して適用されるルールについては、<https://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html> に記載されている Java クラス・パターンの文書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ますが、この場合、オブジェクト名はイベント・ファイルに埋め込まれています。`EmbeddedNameFunctionSelector` は、ラッパーではなく、必要なコンテンツ・デ

ータに基づく関数名を戻します。例えば、コンテンツ固有のビジネス・オブジェクトが `CustomerWrapperBG` の場合、関数セクターで戻される関数は `emitCustomer` です。

`EmbeddedNameFunctionSelector` はデータ・ハンドラーと一緒に構成する必要があります。データ・バインディングは、アダプター固有の `WrapperDataBinding` でなければならず、関数セクターと一緒に構成されたものと同じデータ・ハンドラーを使用するように構成する必要があります。

関連資料

173 ページの『ウィザードの接続プロパティ』

接続プロパティはサービス記述を作成して、組み込みの成果物を保存するために使用されます。これらのプロパティは、外部サービス・ウィザードで構成されます。

ファイル分割

イベント処理中のメモリー・ロードを削減するために、アダプターはファイル分割機能をオプションでサポートしています。この機能を使用する場合、アダプターは大きなイベント・ファイルを小さいチャンクに分割して、エクスポートに別々に送信します。

アダプターは `SplitCriteria` プロパティに指定された値に基づいて、大きなイベント・ファイルを複数のビジネス・オブジェクト (チャンクとも呼ばれる) に分割します。この値は、区切り文字またはチャンク・サイズとして指定できます。各ビジネス・オブジェクトは別々にエクスポートに送達されます。ビジネス・オブジェクトのコンテンツが明確な構造を持つ場合 (例えば、名前、住所、および市などの要素を持つ `Customer` ビジネス・オブジェクトなどが該当) は、ファイルを区切り文字で分割します。ビジネス・オブジェクトに含まれるのがプレーン・テキストまたはバイナリー・ファイルなどの非構造化データの場合、ファイルはサイズによって分割します。

イベント・ファイルがそのようなチャンクに分割される場合は、それぞれのチャンクがビジネス・オブジェクトを作成します。すなわち、`PollQuantity` プロパティに指定した値と、エクスポートに送達したビジネス・オブジェクトの数が、異なってもかまいません。区切り文字に基づいたファイル分割が有効な場合は、`PollQuantity` 活動化仕様プロパティにより、イベント・ストア内に存在する当該イベント・ファイルの数が指定され、`SplittingFunctionClassName` 活動化仕様プロパティでイベント・ファイルの分割に使用されるクラスが設定されます。

アダプターはチャンクに分けたデータの再組み立ては行いません。

`SplitCriteria` プロパティで指定する値によって、使用する方式が決まります。`SplitCriteria` プロパティのデフォルト値がゼロの場合、分割は行われません。分割を必要としない場合は、`SplitCriteria` プロパティおよび `SplittingFunctionClassName` プロパティの値を空のままにしておいてもかまいません。

オプションでカスタム・ファイルのスプリッター・クラスを指定できます。クラス名に `SplittingFunctionClassName` プロパティを設定してください。

区切り文字によるファイル分割

ファイル内のビジネス・オブジェクトを区切るために、コンマ (,)、セミコロン (;)、引用符 ("、')、中括弧 ({}), またはスラッシュ (/ ¥) などの 1 つ以上の文字 (区切り文字) が使用されている場合、アダプターは区切り文字に基づいて、ファイルを小さいチャンクに分割できます。それぞれのチャンクは、WebSphere Process Server または WebSphere Enterprise Service Bus に転送されたときにビジネス・オブジェクトを作成するために使用される論理単位です。ファイルのビジネス・オブジェクトを分離する区切り文字は、SplitCriteria プロパティに定義します。

区切り文字によるファイル分割で使用される PollQuantity 値の働きを示すため、2 つのイベント・ファイルについて考察します。第 1 のイベント・ファイルにはビジネス・オブジェクトが 1 つ、第 2 のイベント・ファイルにはビジネス・オブジェクトが 2 つあります。PollQuantity の値が 2 の場合、最初のビジネス・オブジェクトは第 1 のイベント・ファイルから、次のビジネス・レコードは第 2 のイベント・ファイルから、最初のポーリング周期で送信されます。第 2 のファイルの 2 番目のビジネス・オブジェクトは、次のポーリング周期で送信されます。

区切り文字の使用には、以下の規則が適用されます。

- 区切り文字内のすべての改行は、プラットフォーム固有の改行文字で示す。プラットフォーム固有の改行文字を表 7 に示します。

表 7.

プラットフォーム	改行文字
Macintosh	¥r
Microsoft Windows	¥r¥n
UNIX	¥n

- 複数の区切り文字がある場合、それぞれの区切り文字はセミコロン (;) で分離する必要があります。区切り文字は、指定された順に突き合わせが行われます。セミコロンが区切り文字の一部である場合、¥; のようにしてエスケープする必要があります。例えば、区切り文字が ###¥;## の場合、##;## として処理されます。
- 区切り文字の一部であるコンテンツをスキップする場合、直前に 2 つのセミコロン (;;) を指定すると、区切り文字の間のコンテンツがスキップされます。例えば、以下の形式のビジネス・オブジェクトがイベント・ファイルに含まれていて、区切り文字が ##;\$\$\$ の場合、アダプターは、###\$\$\$ を区切り文字であると見なし、「content skipped by the adapter」の部分スキップします。

```
Name=Smith
Company=IBM
##content skipped by the adapter$$
```

- 区切り文字には任意の値を使用することができ、制限はありません。区切り文字は、有効なストリング、改行文字 (¥n など)、および区切り文字が複数ある場合はセミコロンの分離文字を組み合わせたものです。区切り文字に改行文字およびセミコロンが含まれていなくてもかまいません。改行文字が使用されるのは、ファイルのコンテンツを分割するときに改行文字が考慮される場合に限られます。有効な区切り文字の例は以下のとおりです。

- ###¥;¥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最後に配置された場合、SplitCriteria プロパティーは END_OF_FILE を使用して、ファイルの物理的な末尾を決定します。

一般的なシナリオの例と推奨される区切り文字の形式を以下に示します。

表 8.

データ・バイ インディン グ	BO コンテンツ	推奨される区切り文字の形式
XML	<pre><?xml version="1.0" encoding="UTF-8"?> <customer:Customer xsi:type="customer:Customer" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:customer="http://www.ibm.com/xmlns/prod/websphere/ j2ca/flatfile/customer"> <CustomerName>Deepa</CustomerName> <Address>IBM</Address> <City>Bangalore</City> <State>KA</State> </customer:Customer></pre>	¥n

サイズによるファイル分割

SplittingFunctionClassName プロパティーで指定する値によって、ファイルがサイズで分割されるかどうかが決まります。SplittingFunctionClassName プロパティーを com.ibm.j2ca.extension.utils.filesplit.SplitBySize に設定した場合、SplitCriteria プロパティーには最大ファイル・サイズをバイト単位で表す有効な数字が含まれている必要があります。イベント・ファイルが SplitCriteria プロパティーで指定した値よりも大きい場合、ファイルは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別々に送られます。イベント・ファイルが SplitCriteria の値よりも小さい場合、イベント・ファイル全体がエクスポートに送られます。

イベント・ファイルがチャンクに分割されると、それぞれのチャンクがビジネス・オブジェクトとなります。すなわち、PollQuantity プロパティーに指定した値と、エクスポートに送達したビジネス・オブジェクトの数が、異なってもかまいません。アダプターは PollQuantity の値に従ってポーリングを行います。実際にはファイル内のビジネス・オブジェクトは一度に 1 つずつ処理します。例えば、イベント・ファイルが 3 つのチャンクに分けられた場合、1 つのファイルがポーリングされて 3 つのビジネス・オブジェクトがエクスポートに送達されます (それぞれのチャンクが 1 つのビジネス・オブジェクトを作成しているため)。

エクスポートでは、アダプターはチャンク・データを単一ファイルに再組み立てすることはしませんが、WebSphere Process Server または WebSphere Enterprise

Service Bus でこれらを単一ファイルに再組み立てできるように、チャンクについての情報を提供します。チャンク情報は FlatFileInputStreamRecord レコードの ChunkFileName プロパティに組み込まれ、バイト単位のチャンク・サイズおよび イベント ID が含まれます。チャンクのイベント ID は、eventFileLocation/_timestampStr/_MofN の形式を使用します (M は現在のチャンク番号、N は総チャンク数)。イベント ID の例は、

C:%flatfile%eventdir%eventfile.in/_2005_01_10_10_17_49_864/_3of5 のようになります (timestampStr は year_month_day_hour_minutes_seconds_milliseconds の形式になります)。

Inbound データ変換

Inbound 処理中、アダプターは、外部サービス・ウィザードでのモジュールの構成時に選択したアダプター固有のデータ・バインディングおよびデータ・ハンドラーに基づいて、データ変換を行います。

データ変換を伴う Inbound 処理

Inbound 処理中のデータ変換処理は、モジュール構成時に選択したアダプター固有のデータ・バインディングおよびデータ・ハンドラーによって制御されます。以下のステップで、データ変換を伴う Inbound 処理について説明します。

1. それぞれのイベントは、SplitCriteria プロパティに設定された値に基づいてイベント・ファイルから取り出されます。そのコンテンツはレコードで設定され、データ・バインディングに送信されます。
2. アダプターは、Inbound 操作の予測されるデータ・タイプを確認します。データが汎用タイプ (FlatFile または FlatFileBG) 以外である場合、アダプターはデータ・バインディング内のデータ・ハンドラー・プロパティを確認します。
3. データ・ハンドラーが設定されている場合は、アダプターはデータを変換します。データ・バインディングによって、データ・ハンドラーが呼び出され、コンテンツ固有のビジネス・オブジェクトが返されます。
4. アダプターは、このコンテンツ固有のビジネス・オブジェクトを、関数セレクターによって返されたメソッドを呼び出して、エンドポイントに渡します。

データ変換を伴わない Inbound 処理

コンテンツに対するデータ変換が不要な場合 (例えば、text/xml コンテンツを text/xml コンテンツとして保持する必要がある場合など)、イベント・データはビジネス・オブジェクトに変換されませんが、非構造のコンテンツとしてパススルーされます。

以下のステップで、データ変換を伴わない Inbound 処理について説明します。

1. それぞれのイベントは、SplitCriteria プロパティに設定された値に基づいてイベント・ファイルから取り出されます。そのコンテンツはレコードで設定され、データ・バインディングに送信されます。
2. データ・バインディングは、イベントの予測されるタイプを検査します。汎用タイプ (FlatFile または FlatFileBG) の場合、アダプターはデータを変換しません。
3. データ・バインディングによって、UnstructuredContent レコードでコンテンツが設定され、アダプターに戻されます。

4. アダプターは、このビジネス・オブジェクトを、関数セクターによって返されたメソッドを呼び出して、エンドポイントに渡します。

ビジネス・オブジェクト

ビジネス・オブジェクトは、アダプターで処理されるデータを表す論理データ・コンテナです。データは、ビジネス・エンティティー (送り状または従業員レコードなど) または非構造化テキスト (Eメールの本文またはワープロ文書) のいずれかを表すことができます。アダプターはビジネス・オブジェクトを使用してローカル・ファイル・システムにデータを送信するか、またはデータを取得します。

アダプターがビジネス・オブジェクトを使用する方法

Outbound 処理時に、アダプターは以下を行います。

1. ローカル・ファイル・システム内のファイル上で処理を行うために、要求を表すビジネス・オブジェクトをモジュールから受け取ります。
2. 必要に応じて、ローカル・ファイル・システムで認識できる形式にビジネス・オブジェクトを変換します。
3. 要求された操作を実行します。
4. 該当する場合は、操作の結果を表すビジネス・オブジェクトをモジュールに戻します。

Inbound 処理時に、アダプターは以下を行います。

1. ローカル・ファイル・システム上のイベント・ディレクトリーからファイルを取得します。
2. 必要に応じて、データを必要な形式に変換し、データからビジネス・オブジェクトを作成します。
3. ビジネス・オブジェクトをエクスポートに送信します。

ビジネス・オブジェクトの作成方法

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・エディターのいずれかを使用して作成できます。これらはどちらも WebSphere Integration Developer から起動できます。外部サービス・ウィザードを使用する場合、ウィザードはファイル・システム内のファイルを調べて、それらを表すビジネス・オブジェクトを生成します。また、アダプターに必要な他の成果物も生成します。

ビジネス・オブジェクト・エディターを使用する場合、ビジネス・オブジェクトは手動で作成します。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することができます。

外部サービス・ウィザードを実行すると、Adapter for Flat Filesは、コンテンツ固有および汎用という 2 つのタイプのビジネス・オブジェクトを生成します。アダプターは以下の汎用ビジネス・オブジェクト XSD ファイルを生成します。

- FlatFile.xsd
- FlatFileBG.xsd
- UnstructuredContent.xsd

- FileContent.xsd

コンテンツ固有のビジネス・オブジェクトの例として、Customer があります。Customer を選択した場合は、汎用 XSD ファイルのほかに、次のようなコンテンツ固有の XSD ファイルが生成されます。

- Customer.xsd
- CustomerWrapper.xsd
- CustomerWrapperBG.xsd

注: この例では、ビジネス・グラフ CustomerWrapperBG.xsd が生成されます。ビジネス・グラフの生成はオプションです。

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。

関連資料

165 ページの『ビジネス・オブジェクト情報』

ビジネス・オブジェクトの目的は、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と、ビジネス・オブジェクトの名前の両方を調べることによって判断できます。アプリケーション固有情報は、ローカル・ファイル・システムで実行可能な操作を示します。名前には通常、実行される操作とビジネス・オブジェクトの構造が反映されます。

WebSphere Application Server 環境変数

外部サービス・ウィザードで WebSphere Application Server 環境変数を使用して、ディレクトリー値を指定できます。

外部サービス・ウィザードを使用して Inbound または Outbound 処理のためにアダプターを構成する場合は、必要な様々なローカル・ファイルおよびディレクトリーに値を設定します。後で、WebSphere Process Server または WebSphere Enterprise Service Bus 管理コンソールから、デプロイされたアプリケーションでこれらの値を変更できます。

WebSphere Process Server または WebSphere Enterprise Service Bus バージョン 6.1.x を使用すると、ディレクトリーおよびファイルの値をハードコーディングする代わりに、それらを WebSphere Application Server 環境変数として宣言し、外部サービス・ウィザードの実行時に環境変数の名前を指定できます。アプリケーションをデプロイすると、環境変数は実際の値で置換され、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場合は、WebSphere Process Server または WebSphere Enterprise Service Bus 管理コンソールで環境変数を変更するだけですみます。

WebSphere Application Server 環境変数は、Inbound または Outbound 構成で設定されているすべてのストリングのプロパティー値に使用できます (ブール変数または整変数は不可)。

WebSphere Application Server 環境変数を作成する際、以下を設定します。

- 環境変数の名前。例: EVENT_DIRECTORY
- シンボル名が表す値。例: C:\flatfile\event
- 環境変数の有効範囲。これにより、管理コンソールで環境変数が表示されるレベルが決定されます。有効範囲のレベルとしては、サーバー、ノード、あるいはセルが可能です。
 - サーバー有効範囲は、指定したサーバーに表示を限定します。サーバー有効範囲は、環境変数の定義としては最も限定された有効範囲です。
 - ノード有効範囲は、指定したノード上のすべてのサーバーに表示を限定します。これはデフォルトの有効範囲です。
 - セル有効範囲は、指定したセル上のすべてのサーバーに表示を限定します。

WebSphere Application Server 環境変数を作成するには、WebSphere Process Server または WebSphere Enterprise Service Bus 管理コンソールを使用します。

関連タスク

58 ページの『WebSphere Application Server 環境変数の定義』

WebSphere Process Server または WebSphere Enterprise Service Bus 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。

関連資料

177 ページの『Managed Connection Factory プロパティー』

Managed Connection Factory プロパティーでは、アダプターがローカル・ファイル・システムとの Outbound 通信の実行時に必要とする情報を指定します。

198 ページの『活動化仕様プロパティー』

活動化仕様プロパティーは、エクスポート用の Inbound イベント処理の構成情報を保持しています。活動化仕様プロパティーは、外部サービス・ウィザードまたは管理コンソールのいずれかを使用して設定します。

外部サービス・ウィザード

アダプターが WebSphere Process Server または WebSphere Enterprise Service Bus にデプロイされる前に、外部サービス・ウィザードを使用してアダプターを構成します。ウィザードは、ローカル・ファイル・システム上のファイルを調べ、(指定した検索条件に基づいて) サービスを構築し、ビジネス・オブジェクトおよびインターフェースを生成します。

外部サービス・ウィザードは、ビジネス・オブジェクトの青写真を提供します。これによって目的の成果物を選択でき、デプロイ可能なサービス・オブジェクトおよび記述を生成できます。メタデータのツリー構造からメタオブジェクト・ノードを選択することにより、EIS またはデータベース・エンティティーのビジネス・オブジェクトを生成することができます。メタデータは、ビジネス・グラフとビジネス・オブジェクトで構成されるサービス・データ・オブジェクトに変換されます。

次の図は、外部サービス・ウィザードのフローを示しています。完了すると、アダプター・プロジェクトのすべての情報を収容する EAR ファイルが作成されます。次にこの EAR ファイルは、アプリケーション・サーバーにデプロイされます。

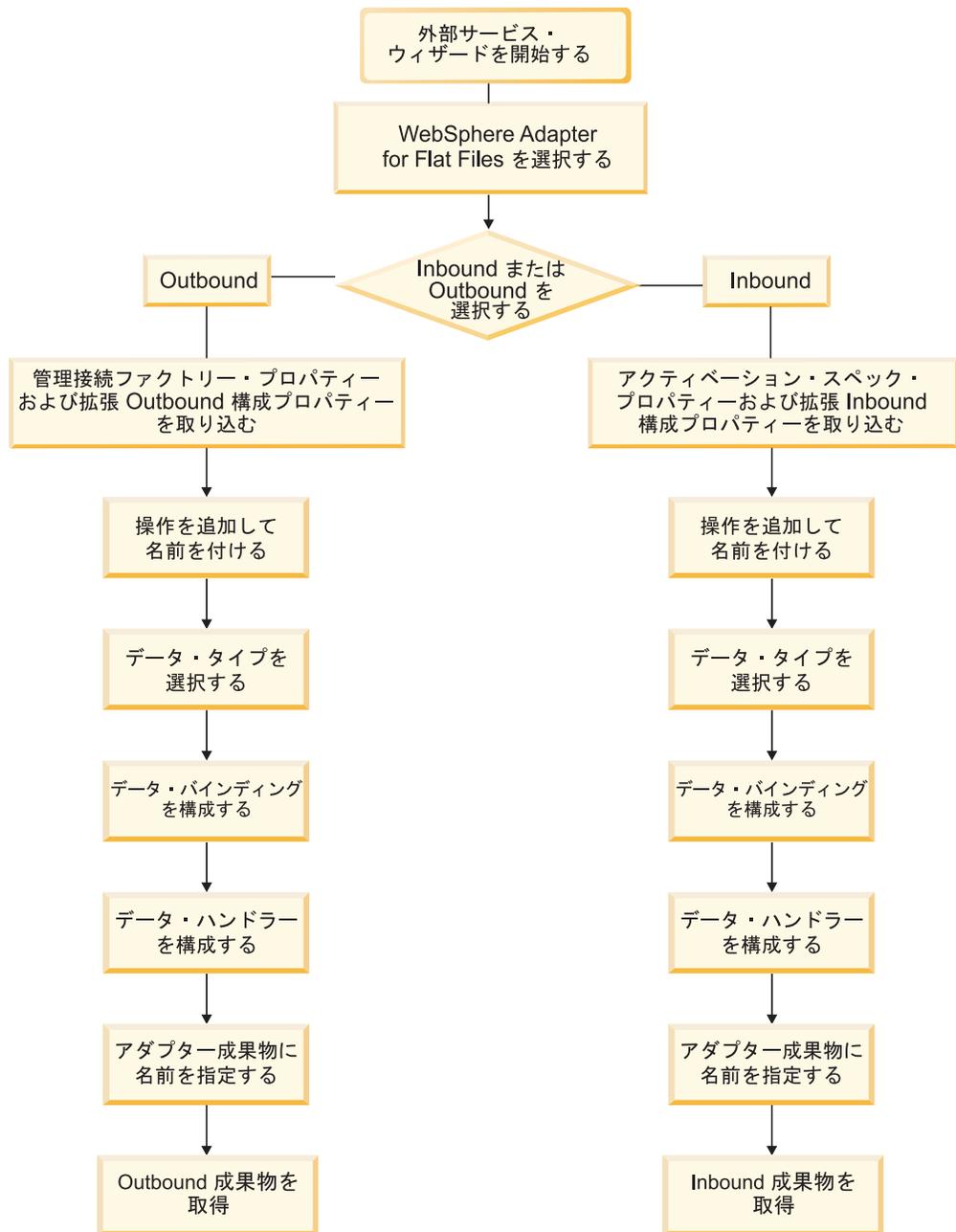


図6. 基本的な外部サービス・ウィザードのフロー

第 2 章 アダプター実装の計画

WebSphere Adapter for Flat Files を実装するには、Inbound および Outbound 処理の計画を立て、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。また、WebSphere Adapter for Flat Files を旧バージョンからマイグレーションする場合は、任意のマイグレーション作業を実行してください。

始める前に

アダプターのセットアップと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、およびソリューションを構築して使用する環境について、完全に理解しておく必要があります。

WebSphere Adapter for Flat Files を構成して使用するには、以下に示す概念、ツール、およびタスクを理解し、経験しておく必要があります。

- 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。
- ソリューションを構築するため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・サーバーを構成および管理する方法と、管理コンソールを使用して、プロパティ定義の設定および変更、接続の構成、およびイベントの管理を行う方法を知っておく必要があります。

セキュリティー

Adapter for Flat Files は、WebSphere Process Server を開始するユーザーの許可に依存します。

アダプターのユーザーは、アダプターがアクセス、読み取り、または変更を試行するディレクトリーおよびファイルに対して、十分なアクセス権限を持つ必要があります。

ログおよびトレース・ファイル内の機密ユーザー・データの保護に対するサポート

アダプターには、権限のない者にログおよびトレース・ファイル内の重要データまたは機密データを表示するのを回避する機能があります。

アダプターのログおよびトレース・ファイルには、ローカル・ファイル・システムのデータが含まれる可能性があり、その中に重要情報または機密情報が含まれる可能性があります。場合によっては、機密データを表示する許可を受けていない個人

がこれらのファイルを見なければなりません。例えば、サポート・スペシャリストは、問題のトラブルシューティングにログおよびトレース・ファイルを使用する必要があります。

このような状況でデータを保護するために、アダプターでは、機密ユーザー・データがアダプター・ログおよびトレース・ファイルに表示されるのを防ぐかどうか指定するようになっています。外部サービス・ウィザードでこのオプションを選択することも、HideConfidentialTrace プロパティーを変更することもできます。このプロパティーが有効になると、アダプターによって機密データは XXX に置換されません。

このオプション・プロパティーについては、177 ページの『Managed Connection Factory プロパティー』を参照してください。

次のタイプの情報は、機密データである可能性があると見なされ、隠蔽されます。

- ビジネス・オブジェクトの内容
- イベント・レコードのオブジェクト・キーの内容
- ユーザー名およびパスワード
- コンマ区切りバージョンのファイルなど、中間形式のビジネス・オブジェクト・データ

次のタイプの情報は、ユーザー・データとは見なされず、隠蔽されません。

- イベント・レコードのオブジェクト・キーの一部ではないイベント・レコードの内容。例: XID、イベント ID、ビジネス・オブジェクト名、イベント状況
- ビジネス・オブジェクト・スキーマ
- トランザクション ID
- 呼び出し順序

デプロイメント・オプション

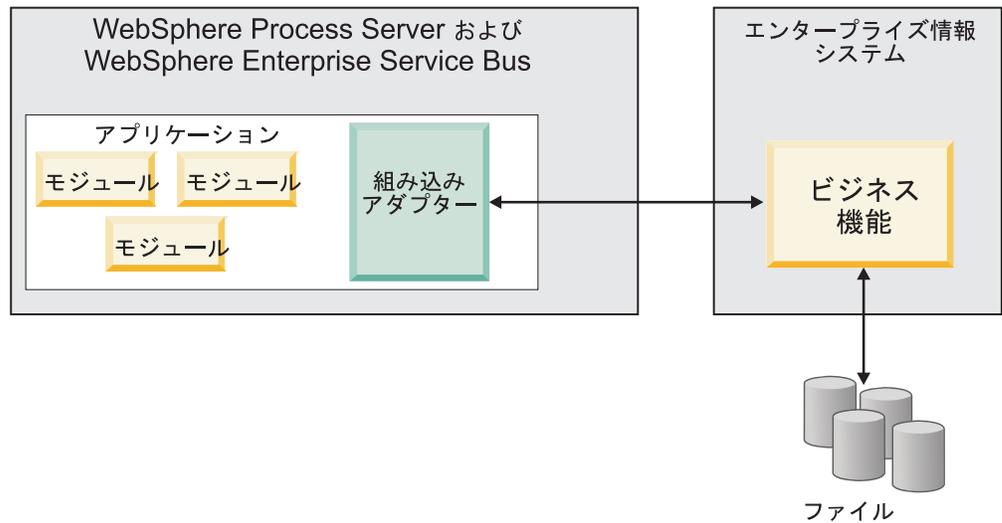
アダプターをデプロイする方法は、以下の 2 つの方法があります。デプロイされたアプリケーションの一部としてアダプターを組み込むか、アダプターをスタンドアロン RAR ファイルとしてデプロイできます。ご利用の環境の要件によって、選択するデプロイメント・オプションのタイプが異なります。

デプロイメント・オプションを以下に説明します。

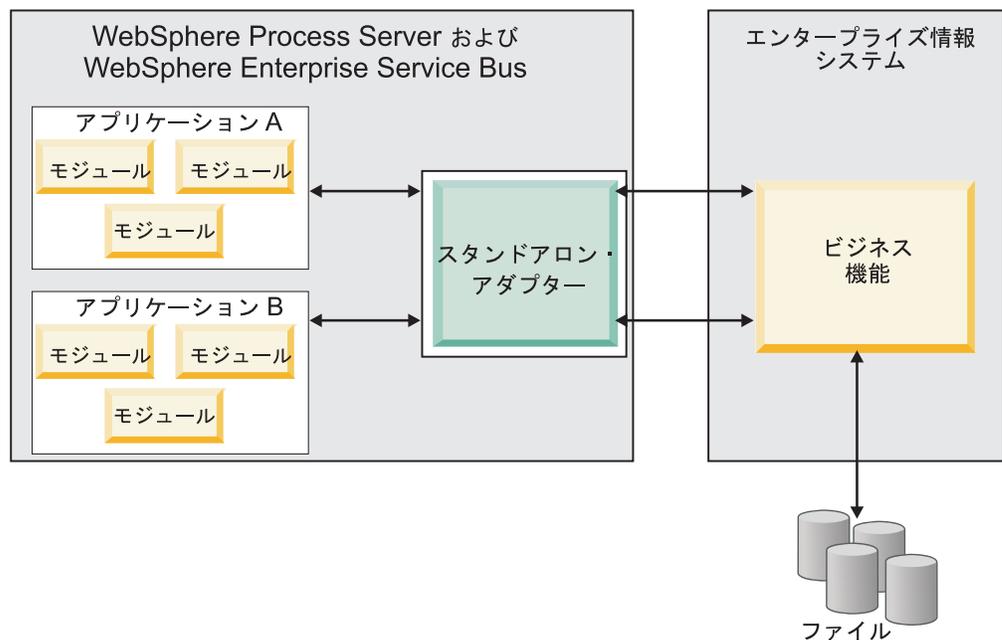
- 「**単一アプリケーションが使用するモジュールとともにデプロイする**」。アダプター・ファイルをモジュール内に組み込むと、モジュールをあらゆるアプリケーション・サーバーにデプロイすることができます。組み込みアダプターを使用するのは、組み込みアダプターを使用するモジュールが 1 つある場合か、複数のモジュールでバージョンの異なるアダプターを実行する必要がある場合です。組み込みアダプターを使用すると、他のモジュールのアダプター・バージョンを変更することで、それらのモジュールを不安定にするリスクを生じることなく、1 つのモジュール内でアダプターをアップグレードできます。
- 「**複数アプリケー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

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、同梱のデプロイ済みアプリケーションでのみ使用することができます。



スタンドアロン・アダプターを表すのは、スタンドアロンのリソース・アダプター・アーカイブ (RAR) ファイルです。これは、デプロイすると、サーバー・インスタンス内のすべてのデプロイ済みアプリケーションで使用できるようになります。



WebSphere Integration Developer を使用してアプリケーション用のプロジェクトを作成するときには、アダプターのパッケージ方法 [(EAR) ファイルでのバンドルか、スタンドアロンの (RAR) ファイルの形式か] を選択することができます。この選択に応じて、ランタイム環境におけるアダプターの使用方法や、管理コンソールにおけるアダプターのプロパティの表示方法が異なります。

アダプターをアプリケーションに組み込むか、スタンドアロン・モジュールとしてデプロイするかは、アダプターの管理方法に応じて選択します。アダプターを 1 コピー保持して、アダプターのアップグレード時に複数のアプリケーションを中断させなくても済むようにしたい場合は、アダプターをスタンドアロン・モジュールとしてデプロイした方が便利です。

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

アダプターのアプリケーションへの組み込みに関する考慮事項

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

- 組み込みアダプターには、クラス・ローダーの独立性があります。

クラス・ローダーは、アプリケーションのパッケージ化、およびランタイム環境にデプロイされたパッケージ済みアプリケーションの動作に影響を与えます。クラス・ローダーの独立性とは、アダプターが、他のアプリケーションまたはモジュールからクラスをロードできないということを意味します。クラス・ローダーの独立性により、異なるアプリケーション内の類似する名前を持つ 2 つのクラスは互いに干渉しなくなります。

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

スタンドアロン・アダプターを使用する際の考慮事項

スタンドアロン・アダプターを使用する場合は、以下の点を考慮してください。

- スタンドアロン・アダプターには、クラス・ローダーの独立性がありません。

スタンドアロン・アダプターにはクラス・ローダーの独立性がないため、指定された任意の 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およびシーケンスは確定されません。例えば、スタンドアロン・アダプターを使用する場合、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、1 つのサード・パーティーの JAR バージョンのみがあります。スタンドアロン・アダプターとしてデプロイされたアダプターはすべて、単一の AFC バージョンを共有し、1 つのアダプターのすべてのインスタンスは同じコードのバージョンを共有します。1 つのサード・パーティー・ライブラリーを使用するアダプター・インスタンスはすべて、そのライブラリーを共有しなければなりません。

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

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

- 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、スタンドアロン方式でデプロイされるすべての RAR ファイル内には、最新の AFC バージョンが必要です。

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

クラスター環境での WebSphere Adapter

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。モジュールのデプロイに使用するのがスタンドアロン・アダプターであるか組み込みアダプターであるかにかかわらず、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、高可用性とスケーラビリティを提供するために、一緒に管理されるサーバー・グループのことです。サーバー・クラスターをセットアップするときには、デプロイメント・マネージャー・プロファイルを作成してください。デプロイメント・マネージャーのサブコンポーネントである HAManager により、アダプター・インスタンスを活動状態にするよう 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して、WebSphere Application Server Network Deployment の機能を拡張します。動的作業負荷マネージャーは、要求による負荷の平衡化を動的に行うことによって、クラスター内のアダプター・インスタンスのパフォーマンスを最適化できます。これは、負荷の変動に応じて、アプリケーション・サーバー・インスタンスを自動的に停止したり始動したりできることを意味します。これにより、能力や構成が異なる複数のマシンが負荷の変動に一樣に対処できるようになります。WebSphere Extended Deployment の利点について詳しくは、リンク <http://publib.boulder.ibm.com/infocenter/wxinfo/v6r1/index.jsp> 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・インスタンスにて、Inbound 処理および Outbound 処理の両方を処理することができます。

制約事項: Inbound および Outbound 通信中、WebSphere Adapter for Flat Files は WebSphere Process Server または WebSphere Enterprise Service Bus クラスター・バックアップ・ノードとクラスターの 1 次ノードが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、これらのノード間でポーリングを切り替えることができません。例えば、アダプターがプライマリー Windows ノードでポーリングを開始する場合、バックアップ UNIX ノードに切り替えることはできず、進行中のイベントのディレクトリー保管に使用される Windows パスを処理することはできません。

Inbound 処理の高可用性

Inbound 処理は、ローカル・ファイル・システム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 Flat Files 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・ディレクトリーはファイル共有システムに存在しなければならず、いずれのクラスター・マシンに対してもローカルであってはなりません。

モジュールをクラスターにデプロイすると、JCA コンテナーにより、enableHASupport リソース・アダプター・プロパティーが検査されます。enableHASupport プロパティーの値が真である場合 (デフォルトの設定)、すべてのアダプター・インスタンスはポリシー N のうちの 1 つを持つ HAManager に登録されます。このポリシーは、アダプター・インスタンスのうちの 1 つのみがイベントのポーリングを開始することを意味します。クラスター内の他のアダプター・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、ご使用の環境に Outbound 要求のために WebSphere Adapter for Flat Files と対話するアプリケーションが複数ある場合、モジュールをクラスター化された環境にデプロイすることで、パフォーマンスが向上することがあります。クラスター化された環境では、複数の Outbound 要求が同じレコードを処理しようとしないう限り、複数の Outbound 要求を同時に処理することができます。

複数の Outbound 要求が同じレコード (顧客の住所など) を処理しようとした場合、その要求は WebSphere Application Server Network Deployment のワークロード管理機能によって、受信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。

バージョン 6.2 へのマイグレーション

WebSphere Adapter for Flat Files のバージョン 6.2 へマイグレーションすると、アダプターが前のバージョンから自動的にアップグレードされます。さらに、前のバージョンのアダプターが組み込まれたアプリケーションをマイグレーションして、バージョン 6.2 にあるフィーチャーや機能を使用できるようにすることができます。

マイグレーションに関する考慮事項

WebSphere Adapter for Flat Files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 Flat Files を使用するアプリケーションのマイグレーションを行う前に、既存のアプリケーションに影響を与える可能性のある要因について考慮する必要があります。

旧バージョンとの互換性

WebSphere Adapter for Flat Files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のアダプターと完全互換であり、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・バインディングと連携できます。

WebSphere Adapter for Flat Files 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、バージョン 6.0.2.x または WebSphere Adapter for Flat Files バージョン 6.1.0.x を使用していたすべてのアプリケーションは、バージョン 6.2 にアップグレードする際に、変更せずにそのまま実行することができます。ただし、アプリケーションで、バージョン 6.2 のアダプターで提供されるフィーチャーと機能を使用したい場合は、マイグレーション・ウィザードを実行してください。

マイグレーション・ウィザードにより、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 はバージョン 6.2 に置換 (アップグレード) され、バージョン 6.2 のフィーチャーと機能がアプリケーションで使用できるようになります。

注: マイグレーション・ウィザードは、バージョン 6.2 のアダプターで動作するマッパーやメディエーターなどのコンポーネントを作成したり、既存のコンポーネントを変更したりすることはありません。バージョン 6.2 にアップグレードする場合で、アプリケーションにバージョン 6.1.x 以前のアダプターが組み込まれている際に、バージョン 6.2 の機能をアプリケーションで利用する場合は、これらのアプリケーションを変更しなければならない場合があります。

単一モジュール内の成果物でバージョンに関する不整合がある場合、そのモジュール全体にマークが付けられ、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 Flat Files のバージョンに一致しない場合は、これらの製品を同時に実行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9. サポートされない製品の組み合わせ

WebSphere Integration Developer	WebSphere Adapter for Flat Files
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

アップグレードのみを行うか、アップグレードとマイグレーションの両方を行うことを決定

デフォルトでは、マイグレーション・ウィザードは、アダプターのアップグレードを実行してから、アプリケーション成果物をマイグレーションし、アプリケーションがバージョン 6.2 のアダプターの機能を使用できるようにします。プロジェクトを選択してアダプターをアップグレードするという選択を行った場合は、ウィザードが自動的にマイグレーション対象の関連付けられた成果物を選択します。

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が、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領域からアダプター成果物の選択を解除することでそのように設定できます。

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、アダプターがインストールされてアップグレードされますが、成果物はマイグレーションされません。また、バージョン 6.2 のアダプターで提供されるフィーチャーや機能をアプリケーションで利用できません。

テスト環境におけるマイグレーション・ウィザードの最初の実行

アダプター・マイグレーションを行うには、バージョン 6.2 の WebSphere Adapter for Flat Files を使用するアプリケーションの変更が必要になる場合があるため、必ず最初に開発環境でマイグレーションを行って、アプリケーションをテストしてから、そのアプリケーションを実稼働環境にデプロイしてください。

マイグレーション・ウィザードは開発環境と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、サポートされていても推奨されてはならず、廃止される可能性がある機能です。従来のバージョンの WebSphere Adapter for Flat Files の機能のうち、バージョン 6.1.x から非推奨になったもので、アプリケーションの変更が必要になる可能性があるのは、次のとおりです。

- 活動化仕様:
 - ArchivingProcessed
 - EventContentType
 - DefaultObjectName
- 対話仕様:
 - DefaultObjectName
- Wrapper プロパティー
 - RetrieveContentType
 - DefaultObjectName

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。ツールが終了するとマイグレーションは完了し、プロジェクトの処理またはモジュールのデプロイを行うことができます。

始める前に

「マイグレーションに関する考慮事項」に記載されている情報を確認します。

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、以下のような変化が起きます。

- モジュールは、前のバージョンの WebSphere Process Server または WebSphere Enterprise Service Bus または WebSphere Integration Developer との互換性を失ってなくなります。
- XML データ・ハンドラーがすべての操作に追加されます。このデータ・ハンドラーは、パススルー操作には必要ありませんので、パススルー操作に対してデータ・ハンドラーのないデータ・バインディングを 1 つ構成する必要があります。

注: 以下のステップでは、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・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を説明します。

このタスクの手順

1. 既存のプロジェクト用の PI (プロジェクト交換) ファイルまたはデプロイ済みアプリケーション用の EAR (エンタープライズ・アーカイブ) ファイルをワークスペースにインポートします。

2. Java EE パースペクティブに移動します。
3. モジュールを右クリックして、「更新」 → 「コネクタ・プロジェクトの更新」と選択します。

注: さらに、次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 パースペクティブの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」と選択します。
- 「問題」ビューから、マイグレーション固有のメッセージを右クリックし、「クイック・フィックス (Quick Fix)」を選択して問題を訂正します。

Java EE パースペクティブでコネクタ・プロジェクト・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と、デフォルトですべての依存関係のある成果物プロジェクトが選択されます。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。選択解除したプロジェクトはどれも後でマイグレーションすることが可能です。既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、およびエラーを含むプロジェクトは、マイグレーションで選択不可であるため、選択されません。

4. ウェルカム・ページにあるタスクおよび警告を確認してから、「次へ」をクリックします。
5. 警告ウィンドウが開き、「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それぞれの成果物に対して行われるマイグレーションによる変更を、+ 記号をクリックして確認できます。
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、ウィザードにより、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がバックアップされます。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。何らかの理由でマイグレーションが失敗したり、マイグレーション完了前にマイグレーションを取り消した場合には、ウィザードが変更したプロジェクトを削除し、一時フォルダー内に保管したプロジェクトで置換します。

マイグレーションが正常に完了した時点で、すべてのバックアップ・プロジェクトは削除されます。

8. 「問題」ビューを調べて、ストリング CWPAD で始まるマイグレーション・ウィザードからのメッセージを探します。
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および成果物を持つ新規の EAR ファイルを作成し、WebSphere Process Server または WebSphere Enterprise Service Bus にデプロイします。EAR ファイルのエクスポートおよびデプロイについて詳しくは、本書の該当するトピックを参照してください。

結果

プロジェクトまたは EAR ファイルは、バージョン 6.2 にマイグレーションされます。アダプター・マイグレーション・ウィザードを終了した後で、外部サービス・ウィザードを実行する必要はありません。

バージョン 6.0.2.x またはバージョン 6.1.x プロジェクトをマイグレーションせずに更新する

アダプターをバージョン 6.0.2.x からバージョン 6.1.x またはバージョン 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がバージョン 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、事前にバージョン 6.0.2.x またはバージョン 6.1.x プロジェクト内の成果物を更新して、新しい名前を使用する必要があります。マイグレーション・ウィザードを使用して、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へインポートします。
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」を選択します。アダプター・マイグレーション・ウィザードが開きます。
3. ウェルカム・ページで、「次へ」をクリックします。
4. 「プロジェクトの選択」ウィンドウで、依存関係のある成果物プロジェクトを選択解除して、「終了」をクリックします。
5. 警告ウィンドウが開き、「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 (Quick Fix)」ウィンドウで、修正「参照されたアダプターの名前変更 (Rename the referenced adapter)」が選択されていることを確認して、「OK」をクリックします。
7. エラーがまだ表示される場合、「プロジェクト」 → 「クリーン」とクリックし、先ほど更新したプロジェクトを選択して、「OK」をクリックします。

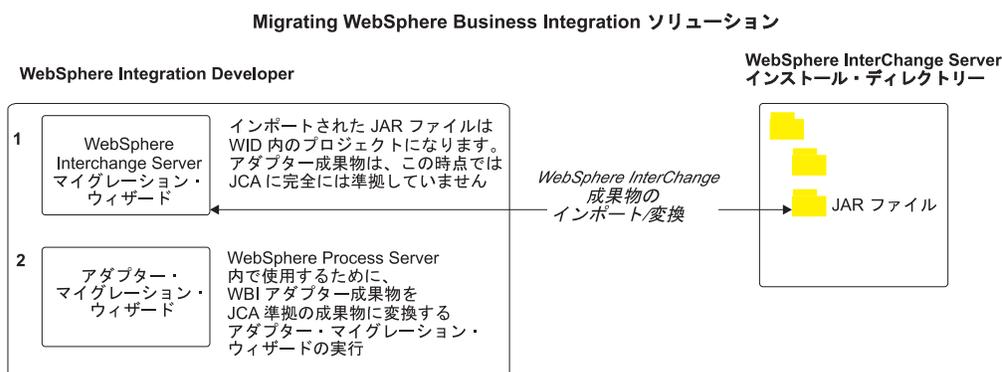
結果

プロジェクトは、WebSphere Adapter for Flat Filesバージョン 6.2 で使用できるようになりました。

WebSphere Adapter のバージョン 6.2 とともに使用するための、WebSphere ビジネス・インテグレーション・アプリケーションのマイグレーション

WebSphere Adapter のバージョン 6.2 とともに使用するための、WebSphere ビジネス・インテグレーション・アプリケーションのマイグレーション・プロセスは、複数のステップから成ります。最初のステップで、WebSphere InterChange Server からの成果物をマイグレーションおよび変換し、WebSphere Integration Developer に成果物用のプロジェクトを作成します。残りのステップで、アダプター固有の成果物をマイグレーションし、成果物をバージョン 6.2 のアダプターがサポートする JCA 準拠フォーマットに変換します。

次のダイアグラムは、WebSphere ビジネス・インテグレーション・ソリューションを WebSphere InterChange Server からマイグレーションするために使用するウィザードを示しており、この結果、これらのアプリケーションは Adapter のバージョン 6.2 とともに使用できるようになります。



WebSphere InterChange Server からアプリケーションをマイグレーションするためのロードマップ

バージョン 6.2 の WebSphere Adapter for Flat Files を WebSphere InterChange Server からのアプリケーションと一緒に使用するには、アプリケーション成果物をマイグレーションして、それらを WebSphere Process Server または WebSphere Enterprise Service Bus 上にデプロイして実行できるように変換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。

次の図は、マイグレーション作業の流れを示しています。図の後に示す手順で、この作業の概要を説明します。これらの各手順の実行方法について詳しくは、このロードマップの後に記載されているトピックを参照してください。

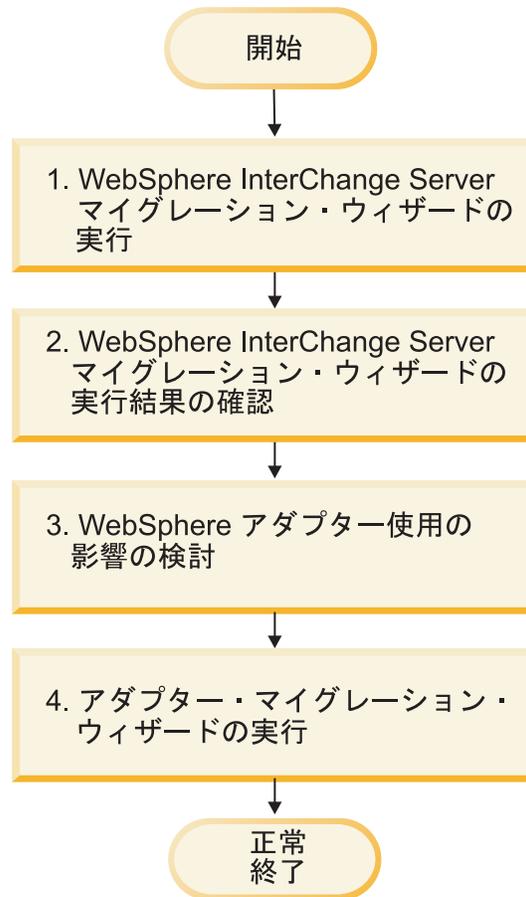


図7. *WebSphere InterChange Server* からアプリケーションをマイグレーションするためのロードマップ

WebSphere InterChange Server からのアプリケーションのマイグレーション

この作業は、次の手順から成ります。

1. WebSphere InterChange Server マイグレーション・ウィザードを実行します。

WebSphere InterChange Server マイグレーション・ウィザードにより、アプリケーション成果物が WebSphere Integration Developer に移動します。この作業が完了した時点では、マイグレーションされたアダプター成果物は完全には JCA に準拠していません。

2. WebSphere InterChange Server からのマイグレーションが成功したことを確認します。

マイグレーション結果ウィンドウですべてのメッセージを検討し、必要であれば処置をとります。

3. WebSphere Adapter for Flat Files のバージョン 6.2を使用した際の影響を考慮します。

WebSphere InterChange Server アプリケーションのマイグレーションに関する考慮事項に加え、WebSphere Adapter for Flat Files のバージョン 6.2 がどのようにマイグレーションされたアプリケーションと連動するか考慮する必要があります。WebSphere InterChange Server アプリケーションにサポートされているアダ

プター操作の中には、アダプターのバージョン 6.2 でサポートされていて、異なる方法で実装されている可能性があるものがあります。

4. アダプター・マイグレーション・ウィザードを実行します。

アダプター・マイグレーション・ウィザードを実行して、アダプターのバージョン 6.2 と共に使用できるように、スキーマおよびサービス定義ファイル (.import、.export、および .wsdl) など、アダプター固有の成果物を更新します。

WebSphere ビジネス・インテグレーション・アダプターのマイグレーションに関する考慮事項

マイグレーションを行うと、Java 2 Platform, Enterprise Edition (J2EE) Connector Architecture (JCA) に準拠し、サービス指向アーキテクチャー用に特に設計されたアダプターになります。

アダプター・マイグレーション・ウィザードを実行する前の WebSphere Interchange Server マイグレーション・ウィザードの実行

アダプター・マイグレーション・ウィザードを実行する前に、WebSphere Interchange Server マイグレーション・ウィザードを使用して、ビジネス・オブジェクト、マップ、コラボレーションなど、WebSphere ビジネス・インテグレーション・アダプター用のアプリケーション成果物を生成します。その後、アダプター・マイグレーション・ウィザードを実行して、スキーマおよびサービス定義ファイル (.import、.export、および .wsdl) などのアダプター固有の成果物を更新して、それらが JCA に準拠したフォーマットに適切に変換されるようにします。

テスト環境におけるマイグレーション・ウィザードの最初の実行

WebSphere ビジネス・インテグレーション・アダプターから WebSphere Adapter for Flat Files にマイグレーションすると、WebSphere Adapter for Flat Files のバージョン 6.2 を使用するそれらのアプリケーションへの変更が必要となる場合があるため、必ず最初に開発環境でマイグレーションを実行してアプリケーションをテストしてから、実稼働環境にアプリケーションをデプロイしてください。

注: WebSphere ビジネス・インテグレーション・アダプターをマイグレーションして、J2EE コネクター・アーキテクチャー (JCA) に準拠するようにすると、新規のアプリケーション固有情報 (ASI) プロパティをビジネス・オブジェクト・エディター内のビジネス・オブジェクトに追加できなくなります。マイグレーション後に新規の ASI プロパティを追加するには、テキスト・エディターでビジネス・オブジェクトを開き、ASI プロパティを手動で追加します。

アプリケーション成果物の WebSphere InterChange Server からのマイグレーション

アプリケーション成果物を WebSphere Integration Developer にマイグレーションするには、WebSphere InterChange Server マイグレーション・ウィザードを実行します。ウィザードにより、成果物のほとんどがインポートされ、WebSphere Process Server または WebSphere Enterprise Service Bus と互換性のあるフォーマットに変換されます。

始める前に

WebSphere Integration Developer から WebSphere InterChange Server マイグレーション・ウィザードを起動し、アプリケーション成果物を WebSphere InterChange Server フォーマットから WebSphere Process Server または WebSphere Enterprise Service Bus と互換性のある成果物にマイグレーションします。

このタスクを実行する理由および時期

成果物を WebSphere InterChange Server からマイグレーションするための準備の方法、およびマイグレーションを実行し、マイグレーションが成功したことを確認するための詳細な説明については、IBM WebSphere Business Process Management インフォメーション・センターにアクセスし、トピック『WebSphere InterChange Server からの WebSphere Process Server または WebSphere Enterprise Service Bus へのマイグレーション』を参照してください。

アダプター固有の成果物 (サービス記述子、サービス定義、ビジネス・オブジェクトなど) は、WebSphere InterChange Server マイグレーション・ウィザードを実行しても、WebSphere Process Server または WebSphere Enterprise Service Bus と互換性のある成果物に完全には変換されない場合があります。アダプター固有の成果物のマイグレーションを完了するには、WebSphere InterChange Server マイグレーション・ウィザードを正常に実行した後にアダプター・マイグレーション・ウィザードを実行してください。

結果

プロジェクトおよびアプリケーション成果物がマイグレーションされ、WebSphere Process Server と互換性のある成果物に変換されます。

次のタスク

アダプター・マイグレーション・ウィザードを実行して、アダプター固有の成果物をマイグレーションしてください。

アダプター固有の成果物のマイグレーション

アダプター・マイグレーション・ウィザードを使用して、プロジェクトをマイグレーションできます。アダプター・マイグレーション・ウィザードを使用すると、アダプターのバージョン 6.2 で使用できるように、スキーマおよびサービス定義ファイル (.import、.export、および .wsdl) などアダプター固有の成果物が更新されます。アダプター・マイグレーション・ウィザードの実行が終了すると、マイグレーションが完了し、プロジェクトで作業したり、モジュールをデプロイできます。

始める前に

アダプター・マイグレーション・ウィザードを実行する前に、次のことを行ってください。

- 「マイグレーションに関する考慮事項」トピックの情報を確認します。

- WebSphere InterChange Server マイグレーション・ウィザードを実行して、WebSphere Process Server または WebSphere Enterprise Service Bus とともに使用するために、プロジェクトをマイグレーションし、データ・オブジェクトを変換します。

このタスクを実行する理由および時期

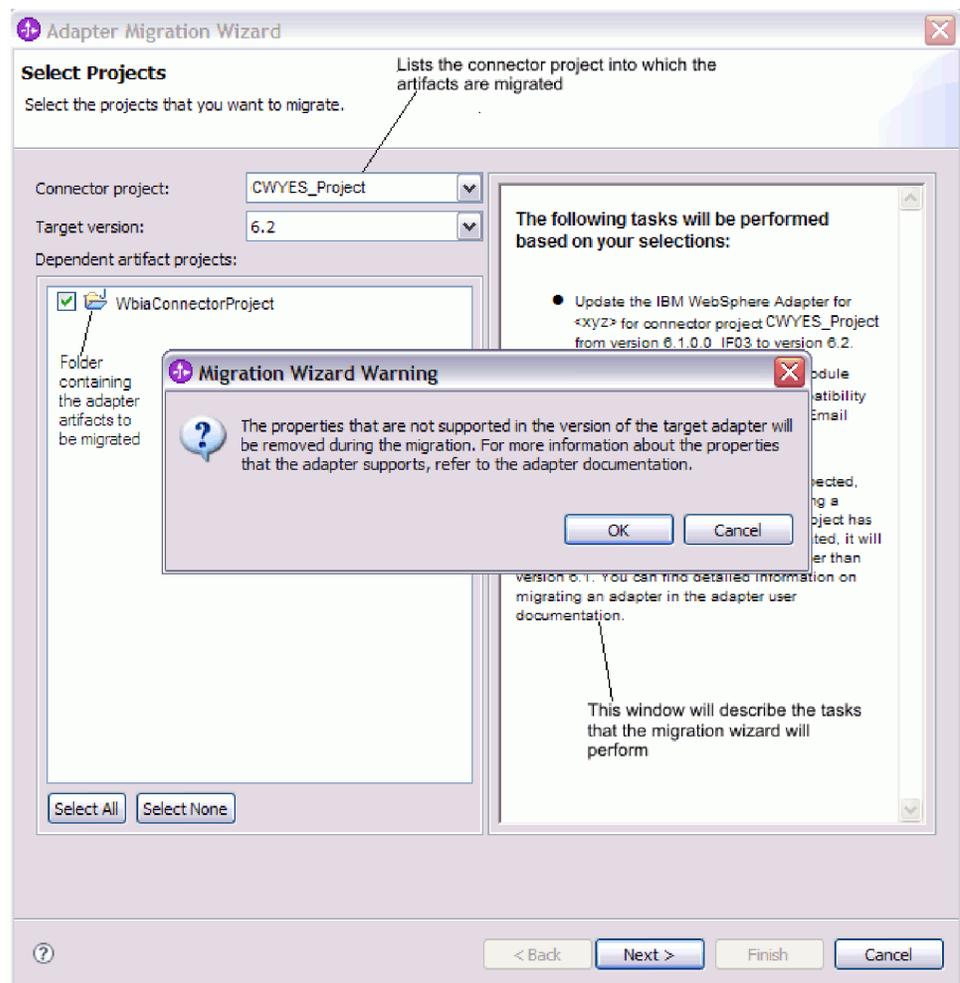
注: マイグレーションの完了後、モジュールはご使用のアダプターのバージョン 6.2 だけと連携します。

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。

このタスクの手順

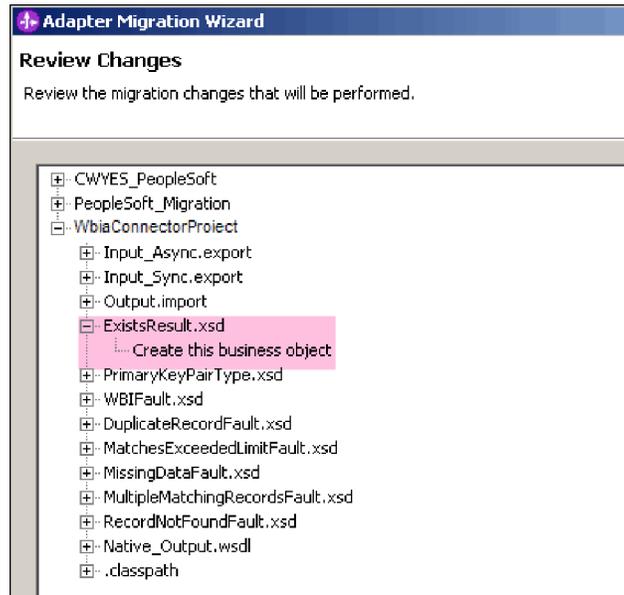
1. Java EE パースペクティブで、コネクタ・プロジェクトを右クリックし、「更新」 → 「コネクタ・プロジェクトの更新」を選択します。

注: さらに、Java EE パースペクティブでモジュール・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」を選択して、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。
次の図では、ウィザードの機能領域について説明します。



Java EE パースペクティブでコネクタ・プロジェクト・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と、デフォルトですべての依存関係のある成果物プロジェクトが選択されます。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。

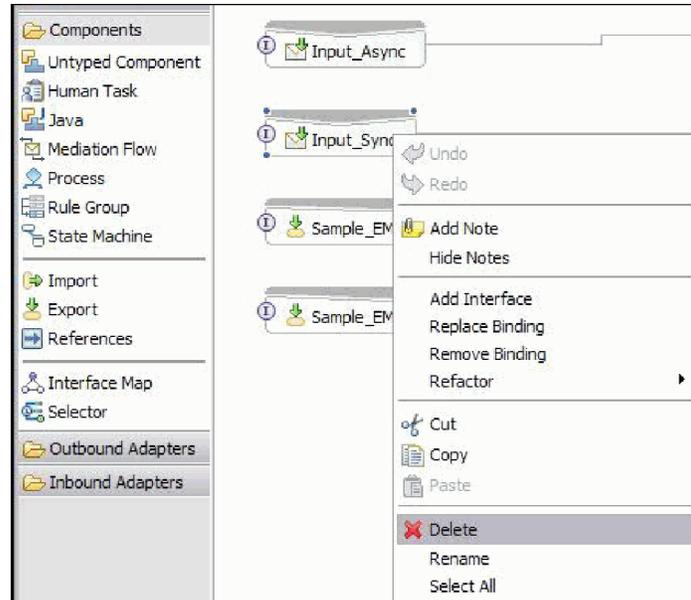
2. 「変更内容の確認」ウィンドウで、マイグレーションするそれぞれの成果物に対して行われるマイグレーションによる変更を、+ 記号をクリックして確認できます。



3. 「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、ウィザードにより、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がバックアップされます。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。何らかの理由でマイグレーションが失敗したり、マイグレーション完了前に取り消した場合には、ウィザードが変更したプロジェクトを削除し、一時フォルダー内に保管したプロジェクトで置換します。

4. 「プロジェクト」>「クリーン (Clean)」を選択して、ワークスペースを最新表示にし再ビルドして、変更内容を有効にします。
5. マイグレーションが成功した際、すべてのバックアップ・プロジェクトは削除されます。同期 Inbound フローは、アダプターで使用されないもので、手動で削除してください。マイグレーション済みのプロジェクトから、Input_Sync Inbound フローを選択し、「削除」を右クリックして選択します。



6. EAR ファイルをマイグレーションする場合、マイグレーション済みのアダプターおよび成果物を使用して新規の EAR ファイルを作成し、WebSphere Process Server または WebSphere Enterprise Service Bus にデプロイします。EAR ファイルのエクスポートおよびデプロイについては、129 ページの『実動用のモジュールのデプロイ』を参照してください。

結果

プロジェクトがバージョン 6.2 にマイグレーションされます。アダプター・マイグレーション・ウィザードを終了した後で、外部サービス・ウィザードを実行する必要はありません。

マイグレーション後のインポート、エクスポート、および WSDL ファイルの変更

WebSphere InterChange Server マイグレーション・ウィザードにより、アプリケーション成果物が WebSphere Integration Developer に移動します。この作業が完了した時点では、マイグレーションされたアダプター成果物は完全には JCA に準拠していません。アダプター・マイグレーション・ウィザードを実行すると、アダプター固有の成果物 (サービス記述子、サービス定義、ビジネス・オブジェクトなど) から JCA 準拠のフォーマットへのマイグレーションを達成できます。マイグレーション・ウィザードによる変更は、サービス定義ファイル (.import、.export、および .wsdl) に反映されます。

マイグレーション後のインポート・ファイルへの変更

マイグレーション時に、影響を受けたモジュール成果物はインポート・ファイルにマイグレーションされます。マイグレーション前の既存 JMS バインディング・プロパティは、インポート・ファイル内では EIS バインディング・プロパティに変更されます。インポート・ファイルで追加されるその他のプロパティ詳細情報として、データ・バインディング構成、Managed Connection Factory プロパティ

内の接続情報に対する変更、いくつかの新規メソッド・バインディングについての情報があります。

マイグレーション後のエクスポート・ファイルへの変更

マイグレーション時に、影響を受けたモジュール成果物はエクスポート・ファイルにマイグレーションされます。マイグレーション前の既存 JMS バインディング・プロパティは、エクスポート・ファイル内では EIS バインディング・プロパティに変更されます。エクスポート・ファイルで追加されるその他のプロパティ詳細情報として、データ・バインディング構成、Managed Connection Factory プロパティ内の接続情報に対する変更、いくつかの新規メソッド・バインディングについての情報があります。

マイグレーション後の WSDL ファイルへの変更

マイグレーション時に、影響を受けたモジュール成果物是对応する WSDL ファイルにマイグレーションされます。これには、Flat Files 固有のサービス記述 WSDL 成果物が含まれます。サービス記述ファイルは、JCA 互換になります。WSDL ファイルには、それぞれの操作ごとに入力タイプと出力タイプがあります。Inbound および Outbound 操作は両方とも特定の入力タイプに機能し、操作実行後、対応する出力タイプが生成されます。次の表では、マイグレーション時に生成される Outbound 操作を示します。

操作	入力	出力
Create	CustomerWrapperBG	CreateResponse
Append	CustomerWrapperBG	AppendResponse
Overwrite	CustomerWrapperBG	OverwriteResponse

注:

- プロジェクト内の複数の Inbound トップレベル・ビジネス・オブジェクトをマイグレーションした場合、最初のトップレベル・ビジネス・オブジェクトの Inbound機能のみが正しく機能します。他のトップレベル・ビジネス・オブジェクトの Inbound を正しく機能させるには、Input_Processing.java および Input_Async_Processing.java クラスで「emit + [verb name] + afterimage + [business object name]」メソッドを手動で変更し、正しい宛先サービスを呼び出す必要があります。
- Inbound 処理中に、emit という動的メソッドが生成されます。emit にビジネス・オブジェクト名が付加されて、実際のメソッドが作成されます。emit メソッドに付加するビジネス・オブジェクト名は、関数セクターにより決定されます。例えば、ビジネス・オブジェクト名が Customer の場合、実際のメソッド名は emitCustomer になります。Emit は、Inbound 処理時に使用可能な唯一の操作です。
- 無効または WebSphere Adapter for Flat Files でサポートされない WebSphere ビジネス・インテグレーション・アダプターのプロパティ (config など) が、マイグレーション後の成果物から除去されます。

第 3 章 サンプルとチュートリアル

ユーザーが、WebSphere Adapters を円滑に使用できるようにするため、「Business Process Management Samples and Tutorials」Web サイトからサンプルおよびチュートリアルが利用できます。

サンプルおよびチュートリアルには、以下のいずれかの方法でアクセスできます。

- WebSphere Integration Developer を始動したときに開くウェルカム・ページから。WebSphere Adapter for Flat Files のサンプルとチュートリアルを表示するには、「取得 (Retrieve)」をクリックします。次に、表示されたカテゴリーを参照して選択します。
- Web 上の以下のロケーション: Business Process Management Samples and Tutorials <http://publib.boulder.ibm.com/bpcsamp/index.html>

第 4 章 デプロイメントのためのモジュールの構成

アダプターを WebSphere Process Server または WebSphere Enterprise Service Bus 上にデプロイできるように構成するには、WebSphere Integration Developer を使用して、アダプターをデプロイするときに 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。その後、作成するビジネス・オブジェクトと、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 Flat Files 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 Flat Files 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するトピックを参照してください。

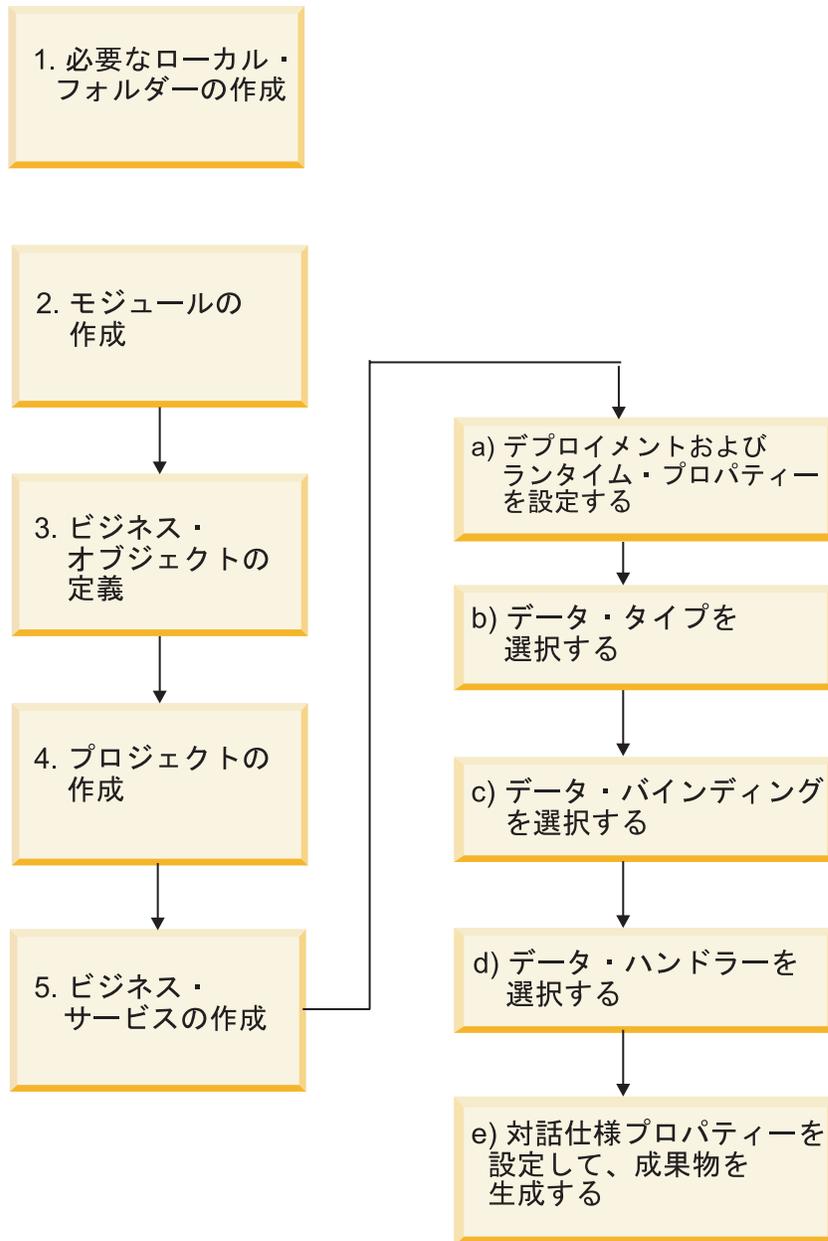


図8. モジュールの構成のためのロードマップ

モジュールの構成

この作業は、次の概略的なステップから成ります。

注: これらのステップでは、データ変換が必要なユーザー定義のビジネス・オブジェクトを使用していることを前提としています。データ変換が不要な汎用ビジネス・オブジェクトを使用している場合は、以下のステップの一部は無視されます。例えば、データ・バインディングとデータ・ハンドラーを選択する必要はありません。

1. WebSphere Integration Developer 内にモジュールを作成します。次に、モジュールにビジネス・オブジェクトを作成します。
2. プロジェクトで使用されるビジネス・オブジェクトを定義します。

3. プロジェクトを作成します。これは、WebSphere Integration Developer で外部サービス・ウィザードを使用していて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します。
4. 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・サービスを作成した後に、以下の手順を実行します。
 - a. 以下のデプロイメントおよびランタイム・プロパティを指定します。
 - 接続プロパティ
 - セキュリティ・プロパティ
 - デプロイメント・オプション
 - 関数セレクター - Inbound のみ
 - b. データ・タイプを選択し、そのデータ・タイプに関連付けられる操作を指定します。操作ごとに、以下を指定します。
 - 操作の種類。例えば、Create、Append、Exists です。
 - 操作をパススルーまたはユーザー定義に指定します。
 - 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同等のデータ・バインディングがありません。
 - d. ビジネス・オブジェクトとネイティブ形式の間の変換を実行するデータ・ハンドラーを選択します。
 - e. 対話仕様プロパティ値を指定して、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せん。

必須のローカル・フォルダーの作成

Inbound モジュールまたは Outbound モジュールを作成する前に、イベントおよび出力用のフォルダーをローカル・ファイル・システムに作成する必要があります。またオプションで、ステージングおよびアーカイブ用のフォルダーを作成することができます。

Inbound モジュールまたは Outbound モジュールを作成する前に、イベント・ディレクトリーおよび出力ディレクトリーを、外部サービス・ウィザードの「サービス構成プロパティ」画面で指定する必要があります。また、ステージング・ディレクトリーおよびアーカイブ・ディレクトリーも作成することができますが、これらは必須ではありません。

- イベント・ディレクトリーは Inbound 処理のイベントを保管します。アダプターは、一定の間隔でこのフォルダーをポーリングし、イベントが検出されると、それをビジネス・オブジェクトの形式でサーバーに送信します。
- 出力ディレクトリーは、Outbound 処理中の Create、Append、および Overwrite 操作の最終出力ファイルを書き込むために、アダプターが使用します。

- ステージング・ディレクトリーは、アダプターが書き込みの競合を回避するために、Create および Overwrite 操作中に初期出力ファイルを書き込む一時ディレクトリーです。出力ファイルはその後名前変更され、出力ディレクトリーにコピーされます。
- アーカイブ・ディレクトリーは、アダプターが処理済みイベント・ファイルを保管するディレクトリーです。

外部サービス・ウィザード を実行する際にこれらのディレクトリーの名前を指定する代わりに、WebSphere Application Server 環境変数を使用できます。

関連タスク

58 ページの『WebSphere Application Server 環境変数の定義』

WebSphere Process Server または WebSphere Enterprise Service Bus 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。

関連資料

177 ページの『Managed Connection Factory プロパティー』

Managed Connection Factory プロパティーでは、アダプターがローカル・ファイル・システムとの Outbound 通信の実行時に必要とする情報を指定します。

198 ページの『活動化仕様プロパティー』

活動化仕様プロパティーは、エクスポート用の Inbound イベント処理の構成情報を保持しています。活動化仕様プロパティーは、外部サービス・ウィザードまたは管理コンソールのいずれかを使用して設定します。

モジュールの作成

WebSphere Integration Developer 内に、モジュールを作成します。このモジュールにより、プロジェクトで使用されるビジネス・オブジェクトを定義できます。

このタスクを実行する理由および時期

外部サービス・ウィザードを起動し、この手順に従って新規モジュールを作成してください。

このタスクの手順

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

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

- c. オプション: 「WebSphere Integration Developer」ウィンドウが表示されたら、「ビジネス・インテグレーション・パースペクティブに移動 (Go to the Business Integration perspective)」をクリックします。

2. 「WebSphere Integration Developer」ウィンドウの「ビジネス・インテグレーション」セクション内を右クリックします。「新規」 → 「モジュール」をクリックします。

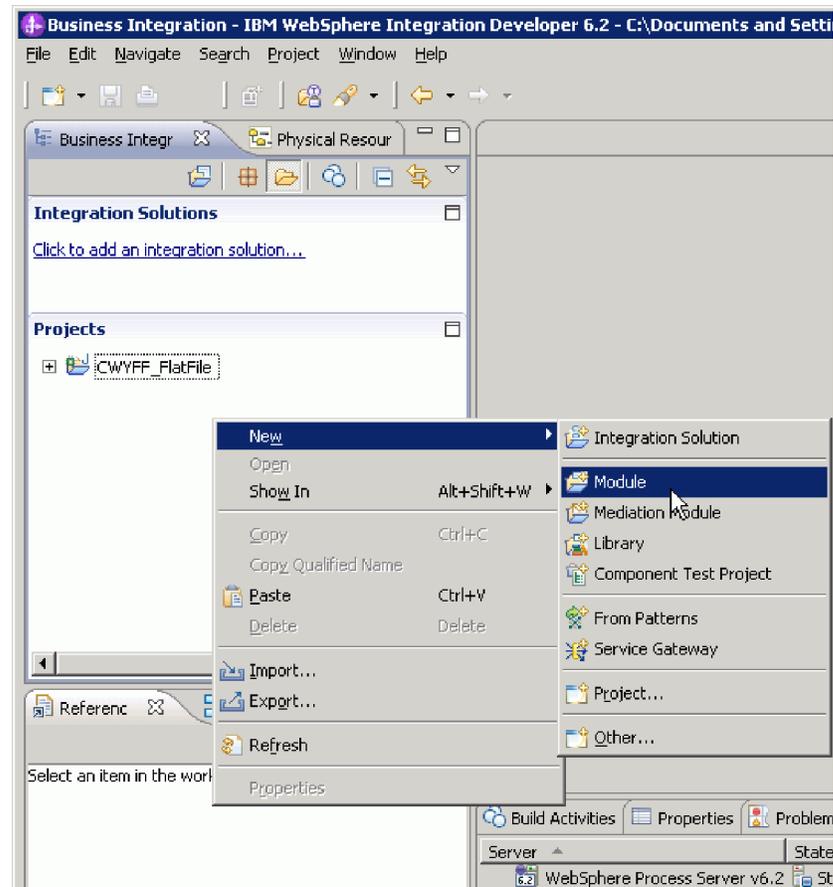


図9. ウィンドウの「ビジネス・インテグレーション」セクション

3. 「新規モジュール」ウィンドウで、「モジュール名」に新しいモジュールの名前を入力します。その他のオプション（「デフォルト・ロケーションの使用 (Use default location)」および「モジュール・アセンブリー・ダイアグラムを開く (Open module assembly diagram)」）はチェック・マークを付けたままにします。

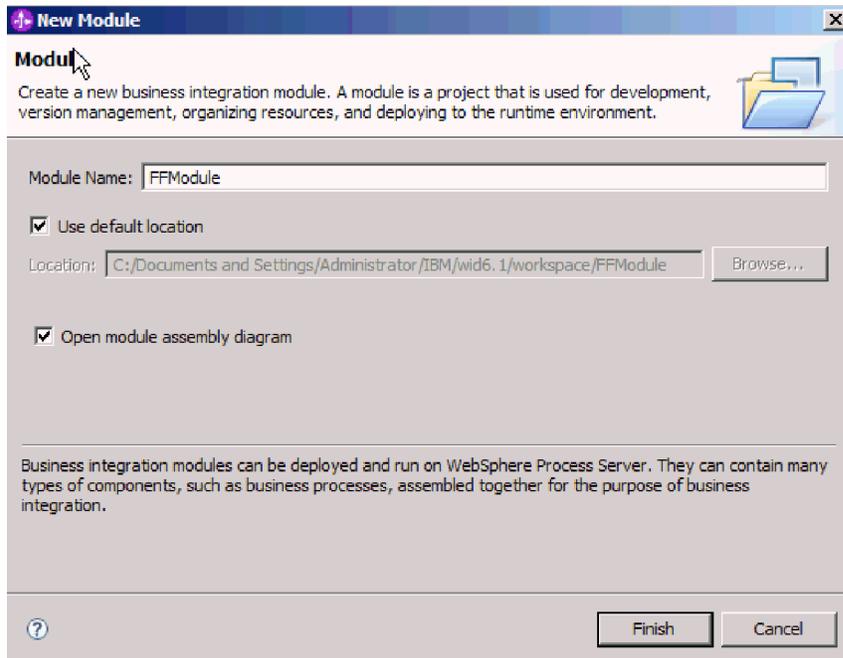


図 10. 「新規モジュール」ウィンドウ

4. 「終了」をクリックします。

結果

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

次のタスク

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

WebSphere Application Server 環境変数の定義

WebSphere Process Server または WebSphere Enterprise Service Bus 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、次の手順を使用してください。

このタスクの手順

1. 管理コンソールを開始します。
2. 「環境」 → 「WebSphere 変数 (WebSphere Variables)」を選択します。

- 環境変数の有効範囲を選択します。有効範囲は、リソース定義が管理コンソール・パネルで表示されるレベルを指定します。指定可能な値は、サーバー、ノード、セルです。この例では、「Cell=widCell」を選択します。

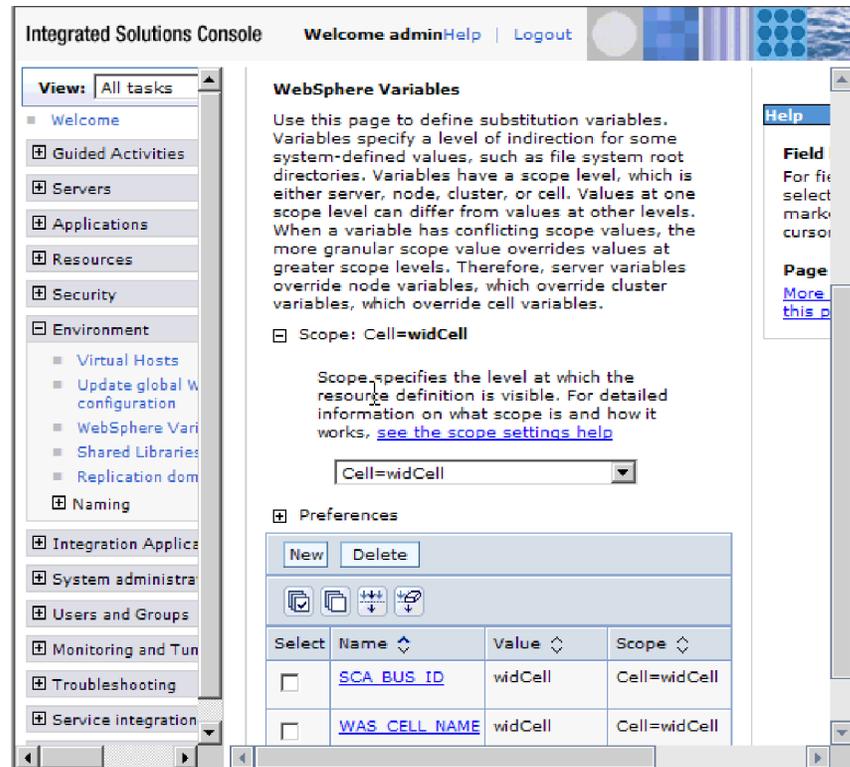


図 11. 環境変数の有効範囲の設定

- 「新規」をクリックし、環境変数の名前および値を設定します。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、名前が EVENT_DIRECTORY で、値は C:/flatfile/event です。オプションで「説明」フィールドを使用して、変数の目的を記述できます。

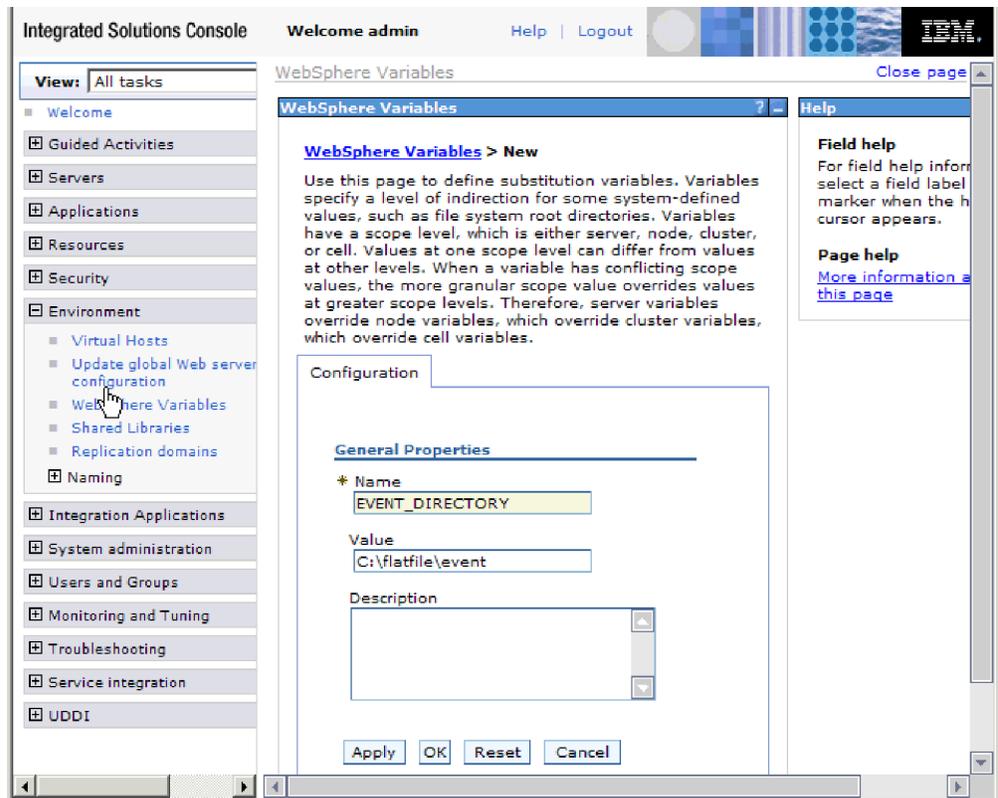


図 12. 環境変数の名前および値の設定

5. 「OK」をクリックして、変更を保存します。

結果

EVENT_DIRECTORY という環境変数が、値 C:\flatfile\event、有効範囲 Cell=widCell で設定されます。イベント・ディレクトリーを指定するときは常に、外部サービス・ウィザードでこれを使用できます。

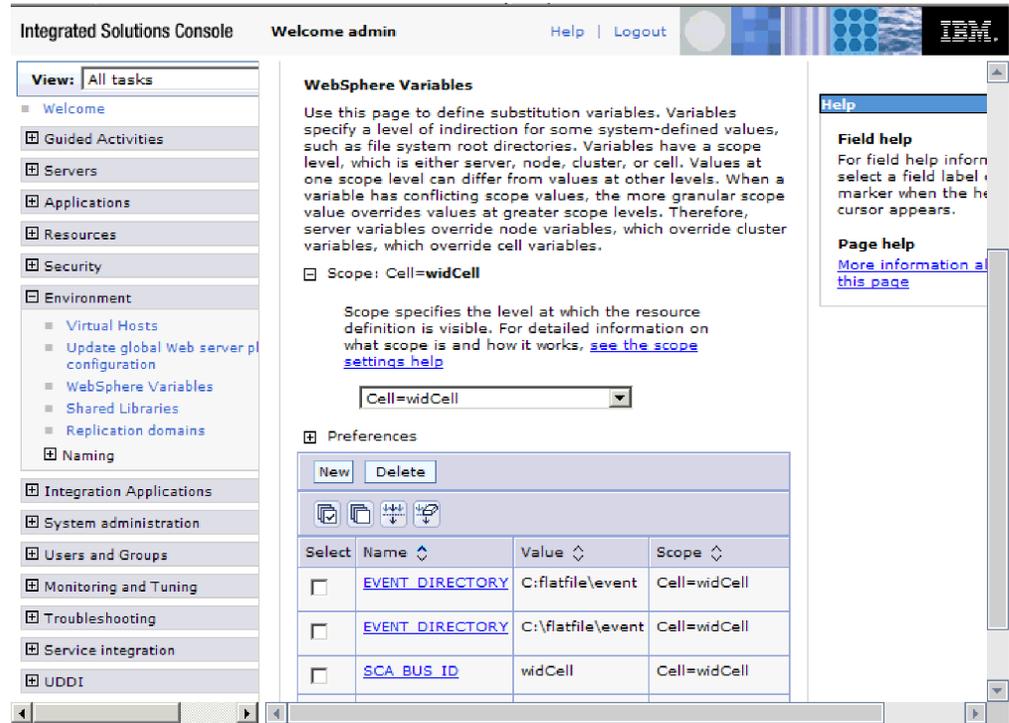


図 13. 「WebSphere 変数 (WebSphere Variables)」 ウィンドウに表示された新規環境変数 `EVENT_DIRECTORY`

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。

関連概念

28 ページの『WebSphere Application Server 環境変数』

外部サービス・ウィザードで WebSphere Application Server 環境変数を使用して、ディレクトリー値を指定できます。

55 ページの『必須のローカル・フォルダーの作成』

Inbound モジュールまたは Outbound モジュールを作成する前に、イベントおよび出力用のフォルダーをローカル・ファイル・システムに作成する必要があります。またオプションで、ステージングおよびアーカイブ用のフォルダーを作成することができます。

関連資料

177 ページの『Managed Connection Factory プロパティー』

Managed Connection Factory プロパティーでは、アダプターがローカル・ファイル・システムとの Outbound 通信の実行時に必要とする情報を指定します。

198 ページの『活動化仕様プロパティー』

活動化仕様プロパティーは、エクスポート用の Inbound イベント処理の構成情報を保持しています。活動化仕様プロパティーは、外部サービス・ウィザードまたは管理コンソールのいずれかを使用して設定します。

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で使用されるビジネス・オブジェクトを WebSphere Integration Developer で事前定義します。

このタスクを実行する理由および時期

ビジネス・オブジェクト・エディターを使用して、新規ビジネス・オブジェクトの事前定義を行うには、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。

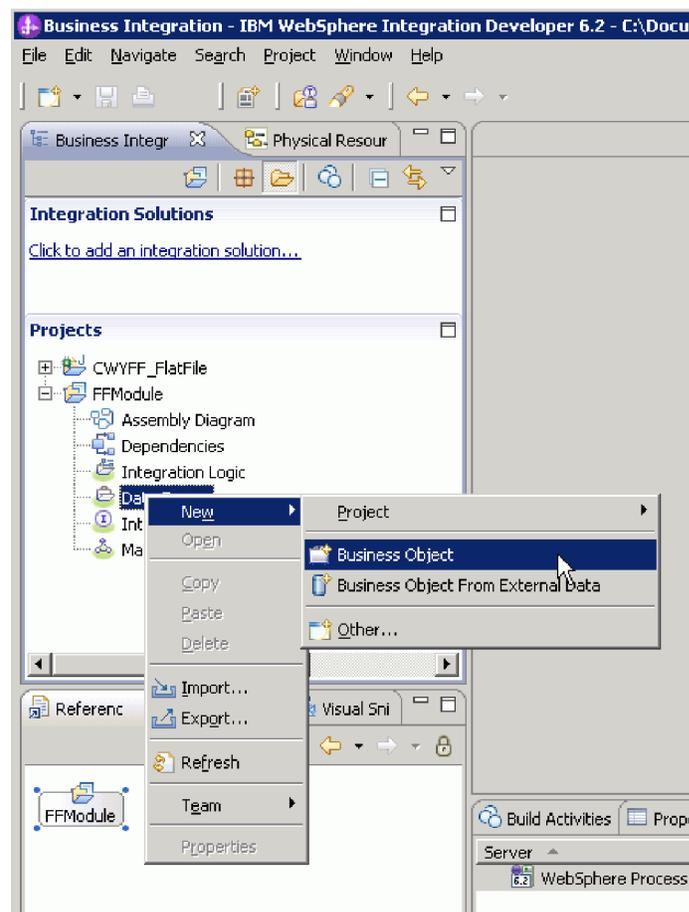


図 14. 新規ビジネス・オブジェクトの選択ビュー

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

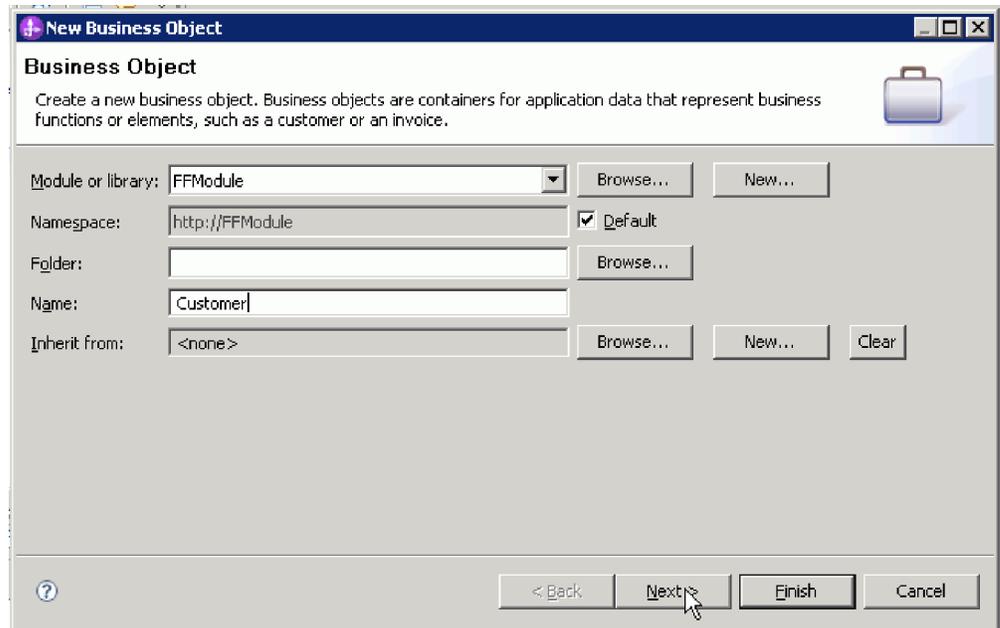


図 15. 「ビジネス・オブジェクト」ウィンドウ

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。

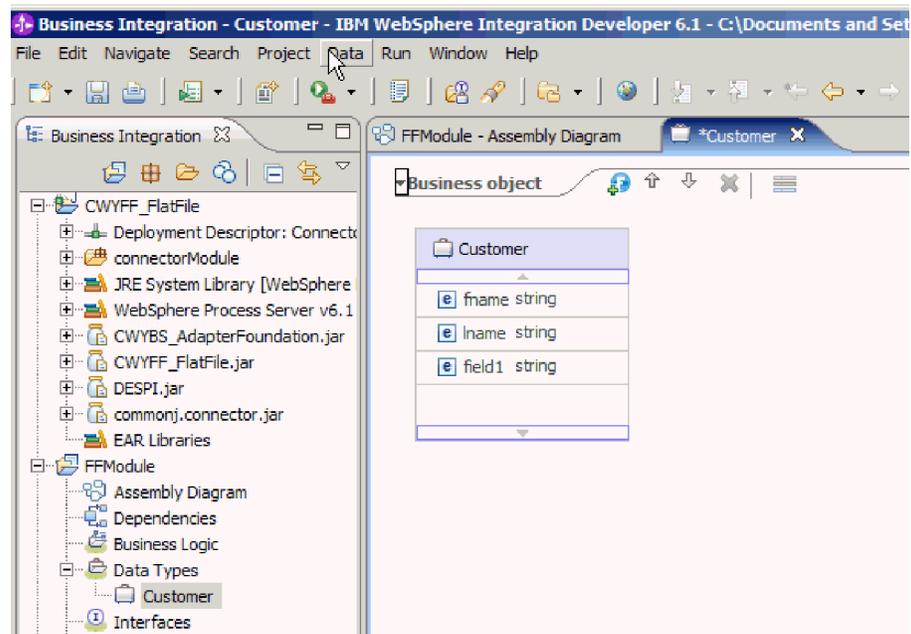


図 16. 「ビジネス・オブジェクト・フィールドの追加 (Add Business object fields)」アイコンを追加します。

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、上記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。

関連概念

27 ページの『ビジネス・オブジェクト』

ビジネス・オブジェクトは、アダプターで処理されるデータを表す論理データ・コンテナです。データは、ビジネス・エンティティ (送り状または従業員レコードなど) または非構造化テキスト (Eメールの本文またはワープロ文書) のいずれかを表すことができます。アダプターはビジネス・オブジェクトを使用してローカル・ファイル・システムにデータを送信するか、またはデータを取得します。

関連資料

165 ページの『ビジネス・オブジェクト情報』

ビジネス・オブジェクトの目的は、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と、ビジネス・オブジェクトの名前の両方を調べることによって判断できます。アプリケーション固有情報は、ローカル・ファイル・システムで実行可能な操作を示します。名前には通常、実行される操作とビジネス・オブジェクトの構造が反映されます。

Outbound 処理時のビジネス・オブジェクトから 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 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 処理中に使用します。

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 内にモジュールが作成済みである。
2. COBOL プログラム・ソース・ファイル (.ccp file) が、ワークステーション上のローカル・ディレクトリー内にある。
3. wrapper ビジネス・オブジェクト定義を生成する場合は、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ある。

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、オプションで外部データ・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・オブジェクト定義を生成します。
 - 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. 「ビジネス・オブジェクト」ウィンドウで、「ビジネス・オブジェクトの作成 (Create business objects)」を選択します。

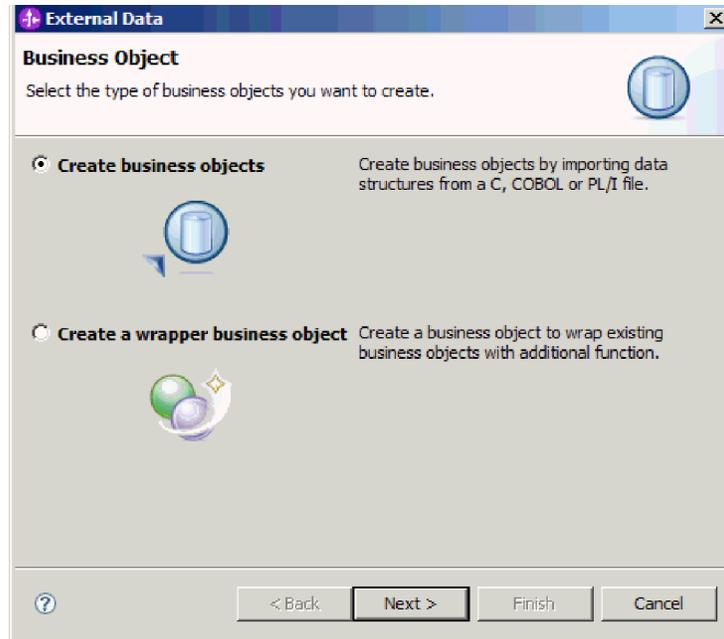


図 17. 「ビジネス・オブジェクト」ウィンドウ

- c. 「次へ」をクリックします。
- d. 「ビジネス・オブジェクト・マッピングの詳細 (Business Object Mapping Details)」ウィンドウで、「マッピングの選択 (Choose mapping)」の値が「COBOL からビジネス・オブジェクト (COBOL to Business Object)」であることを確認します。「参照」をクリックし、.ccp ファイル (この例では taderc99.ccp) を選択します。

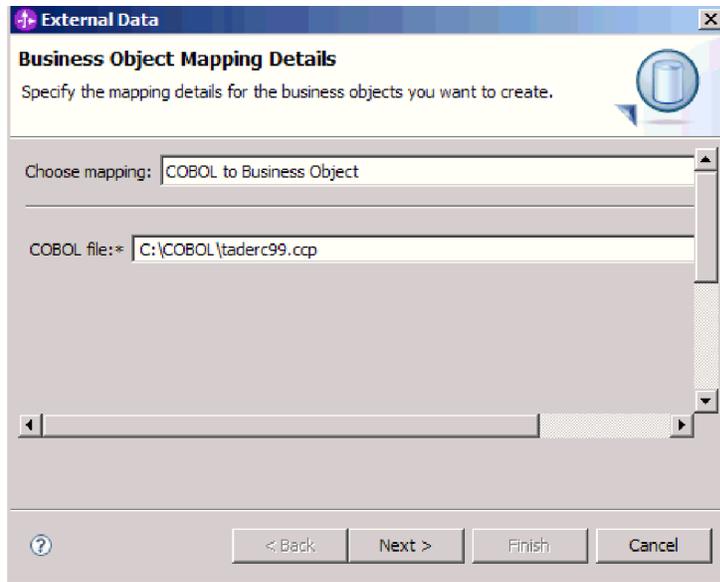


図 18. 「ビジネス・オブジェクト・マッピングの詳細 (Business Object Mapping Details)」 ウィンドウ

- 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」をクリックします。新規ビジネス・オブジェクト (図の「DFHCOMMAREA」) が表示されます。

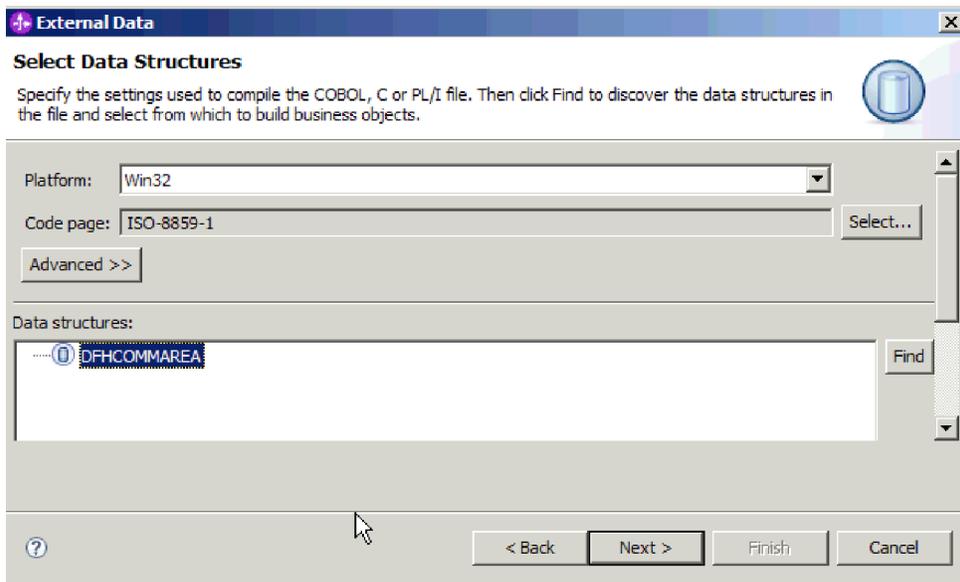


図 19. 「データ構造の選択 (Select Data Structures)」 ウィンドウ

- g. ビジネス・オブジェクト 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。

ビジネス・オブジェクト (図の「DFHCOMMAREA」) がモジュール内に作成されます。

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、既存のビジネス・オブジェクト定義を追加機能でラップします。 wrapper ビジネス・オブジェクト定義の生成オプションは、アダプター RAR ファイルがワークスペースにインポート済みである場合に限り表示されます。
 - 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 (External Data)」を選択します。
 - b. 「ビジネス・オブジェクト」ウィンドウで、「wrapper ビジネス・オブジェクトの作成 (Create a wrapper business object)」を選択します。

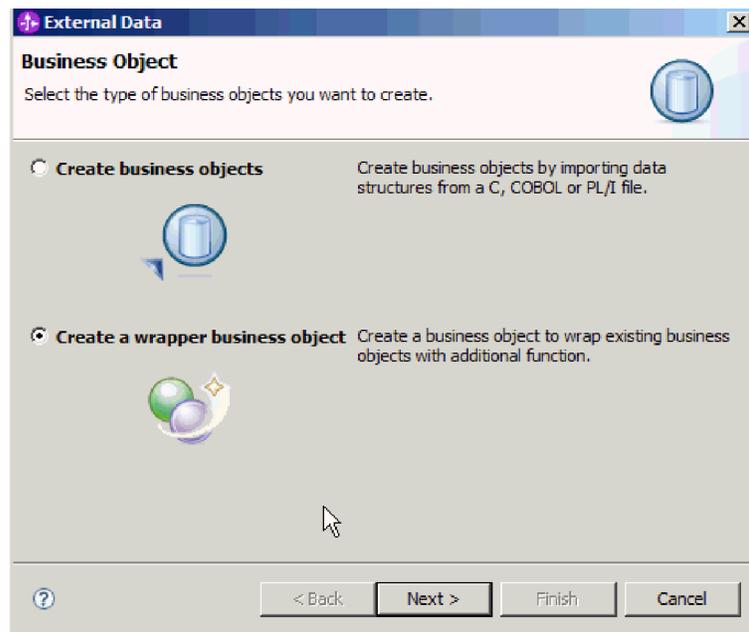


図 20. 「ビジネス・オブジェクト」ウィンドウ

- c. 「次へ」をクリックします。
- d. 「アダプターの選択 (Select an Adapter)」ウィンドウで、wrapper ビジネス・オブジェクトを生成するコネクタ・プロジェクトを選択し、「次へ」をクリックします。
- e. 「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、ステップ 1 で作成したビジネス・オブジェクト (例: DFHCOMMAREA) をデータ・タイプとして選択します。
- f. 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。Retrieve 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 (Generate retrieve container to retrieve multiple business objects)」チェック・ボックスを選択します。

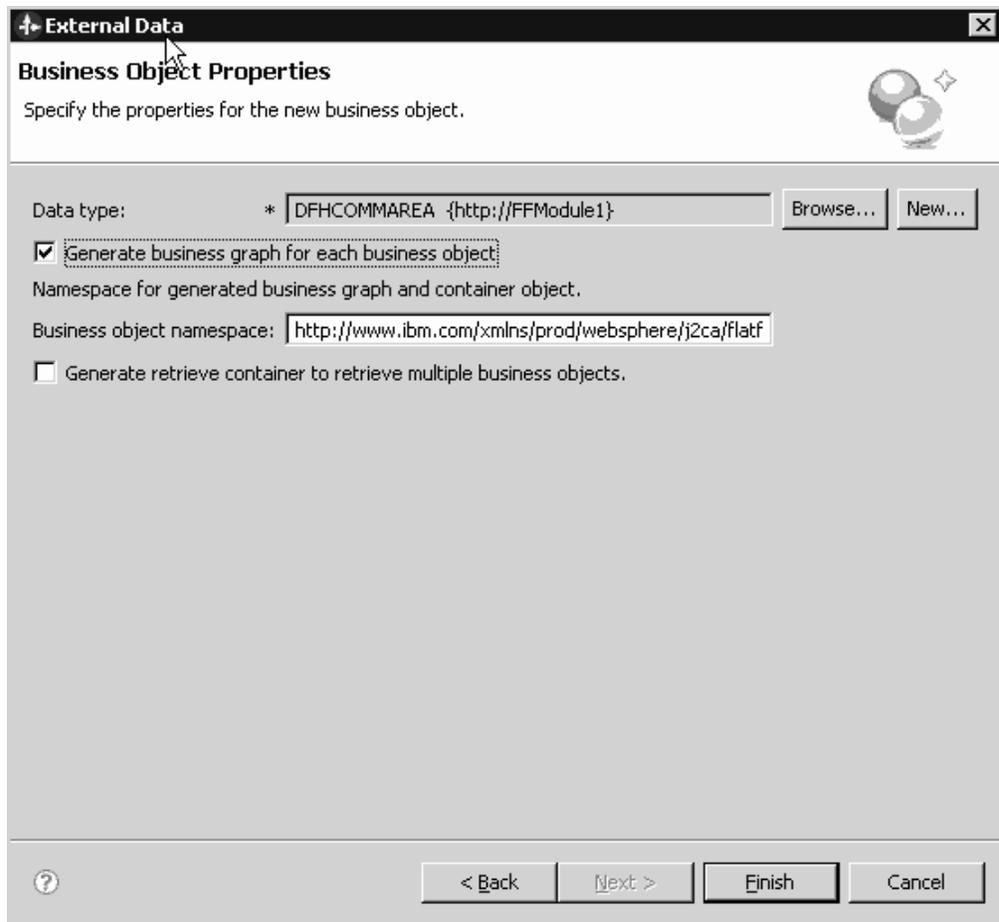


図 21. 「ビジネス・オブジェクト・プロパティ」ウィンドウ

g. 「終了」をクリックします。

wrapper ビジネス・オブジェクトとビジネス・グラフ (それぞれ、図の DFHCOMMAREAWrapper と DFHCOMMAREAWrapperBG) が、「ビジネス・インテグレーション」ウィンドウ内の現行モジュールにリストされます。取得のためのラッパーが選択されている場合は、DFHCOMMAREARetrieveWrapper という名前のビジネス・オブジェクトと DFHCOMMAREARetrieveWrapperBG という名前のビジネス・グラフも、「ビジネス・インテグレーション」ウィンドウ内の現行モジュールにリストされます。

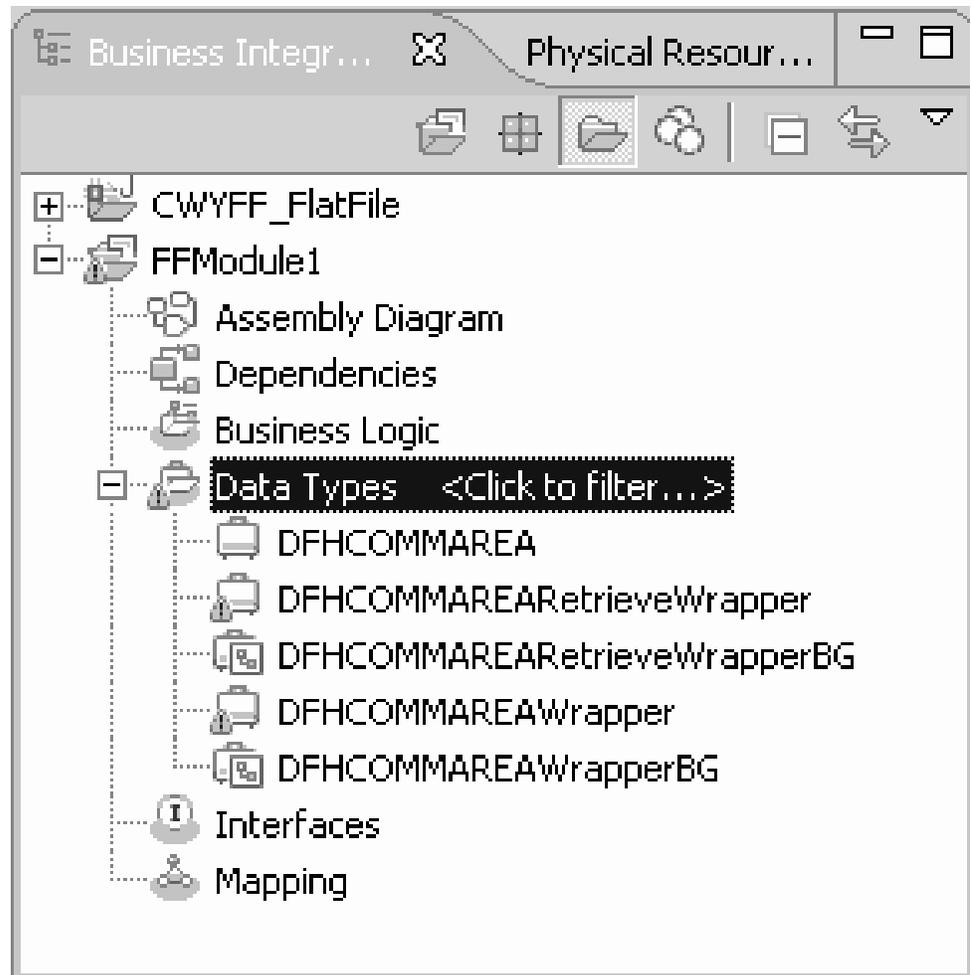


図 22. 「ビジネス・インテグレーション」ウィンドウにリストされた wrapper ビジネス・オブジェクトとビジネス・グラフ

3. COBOL コピーブックの Outbound モジュールに必要な成果物を生成します。
この例では、Create 操作の構成を示します。
 - 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 (External Service)」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 (Select an Adapter)」ウィンドウで、Flat Files コネクター・プロジェクトを選択し、「次へ」をクリックします。

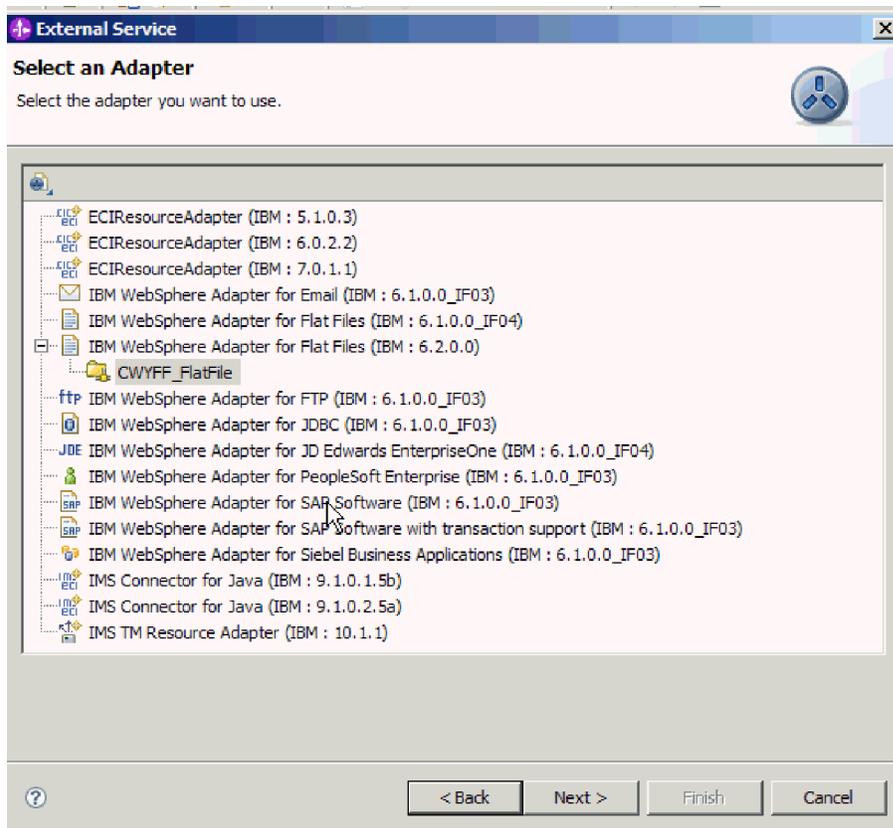


図 23. 「アダプターの選択 (Select an Adapter)」 ウィンドウ

- d. 「処理指示」 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・バインディング**」リストで、「**COBOL、 C、または PL/I データ・バインディング・オプションを使用 (Use COBOL, C or PL/I data binding option)**」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。このツールを使用して、現行モジュールで適切なデータ・バインディング・コードを生成します。

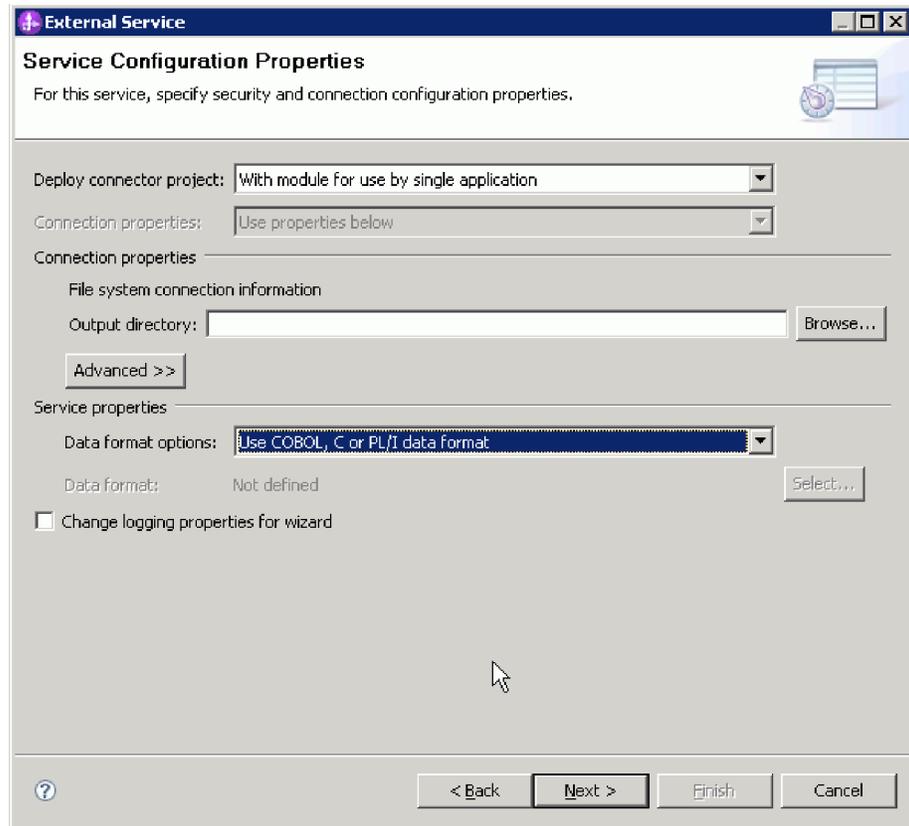


図 24. 「サービス構成プロパティ」ウィンドウ

- g. Outbound 操作に必要な他のプロパティを指定し、「次へ」をクリックします。
- h. 「操作」ウィンドウで、「追加」、「作成」の順にクリックします。Retrieve 操作の場合は、「Retrieve」を選択します。「データ・タイプ」リストから「ユーザー定義タイプ」を選択し、「次へ」をクリックします。
- i. 入力タイプ (DFHCOMMAREA、DFHCOMMAREAWrapper、または DFHCOMMAREAWrapperBG) を参照し、「OK」をクリックします。「Retrieve」操作の場合は、該当入力タイプ (DFHCOMMAREA、DFHCOMMAREARetrieveWrapper、または DFHCOMMAREARetrieveWrapperBG) を参照します。

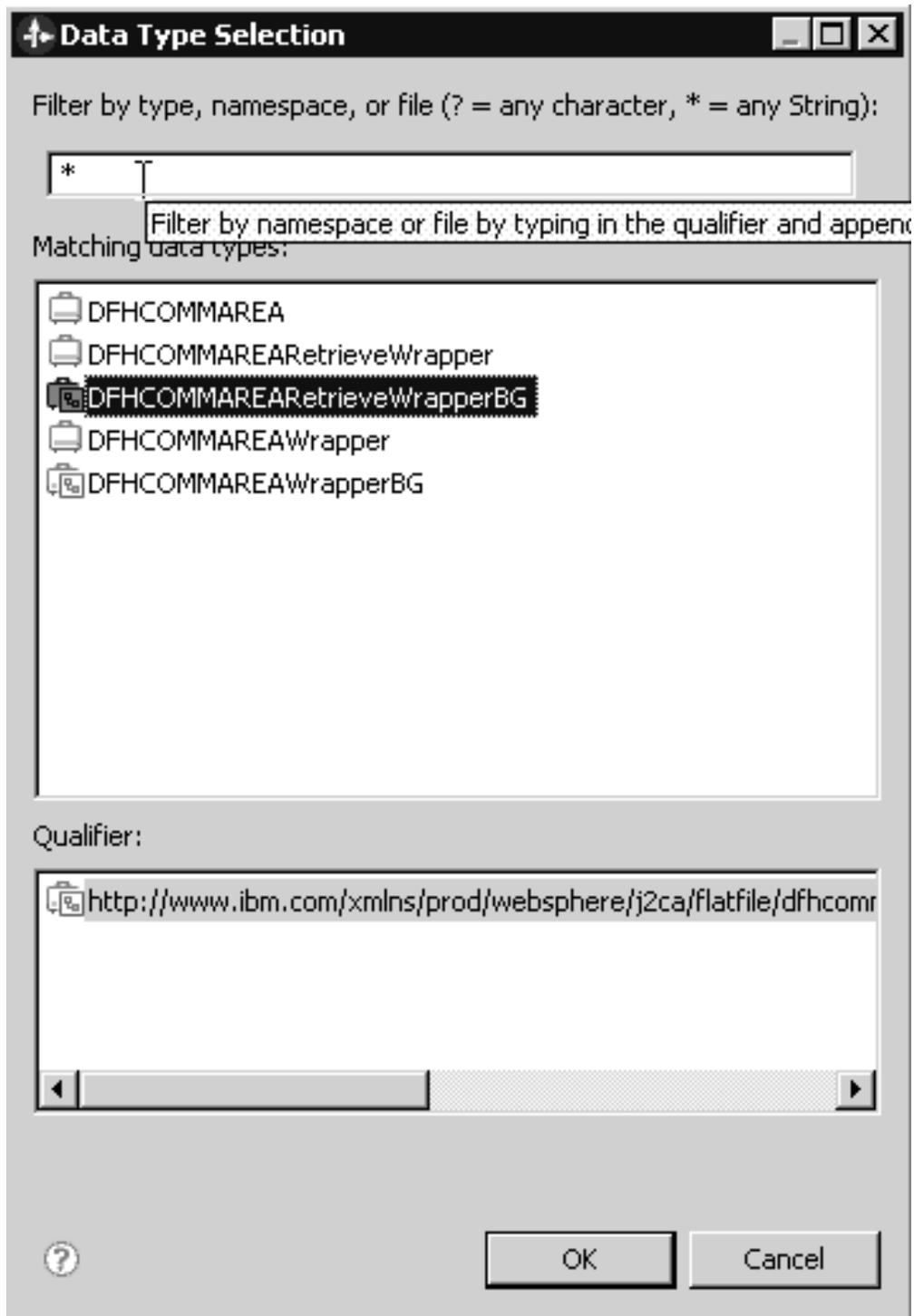


図 25. 「データ・タイプの選択 (Data Type Selection)」 ウィンドウ

j. 「次へ」をクリックし、外部サービス・ウィザードを完了します。

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。生成済みデータ・バインディング・クラスについては、「プロジェクト・エクスプローラー (Project Explorer)」ウィンドウを参照してください。

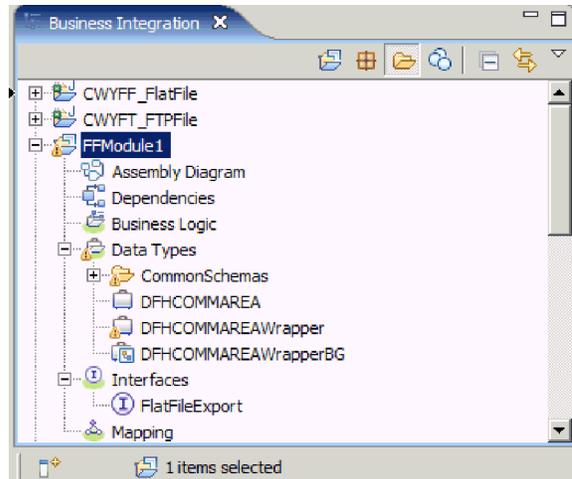


図 26. COBOL コピーブック、WSDL ファイル、インポート・ファイル、およびその他の成果物によって使用されるデータ・バインディング

結果

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、Outbound モジュールの COBOL プログラム・ソース・ファイル用に作成されます。COBOL コピーブック・データ・バインディングを使用する Outbound Create 操作に成果物が生成されます。このモジュールを、WebSphere Process Server または WebSphere Enterprise Service Bus にデプロイして、Create 操作をテストできます。

注: その他のサポートされる操作 (Append および Overwrite) 用に成果物を生成するには、ステップ 3h から同じ手順を実行してください。

次のタスク

モジュールをデプロイします。

Inbound 処理時の COBOL コピーブック・ファイルからビジネス・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 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 処理中に使用します。

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 内にモジュールが作成済みである。
2. COBOL プログラム・ソース・ファイル (.ccp file) が、ワークステーション上のローカル・ディレクトリー内にある。
3. ローカル・イベント・ディレクトリーを作成済みである。
4. wrapper ビジネス・オブジェクト定義を生成する場合は、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ある。

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、オプションで外部データ・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・オブジェクト定義を生成します。
 - 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 (External Data)」を選択します。
 - b. 「ビジネス・オブジェクト」ウィンドウで、「ビジネス・オブジェクトの作成 (Create business objects)」を選択します。

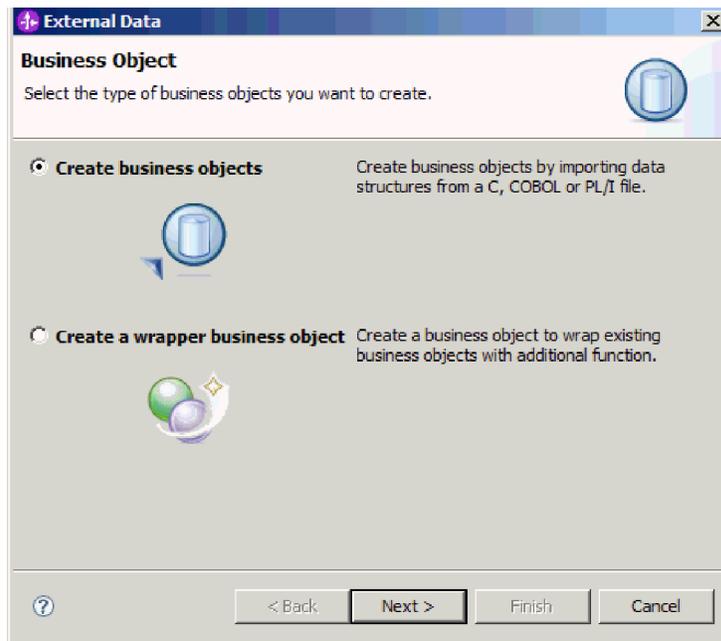


図 27. 「ビジネス・オブジェクト」ウィンドウ

- c. 「次へ」をクリックします。
- d. 「ビジネス・オブジェクト・マッピングの詳細 (Business Object Mapping Details)」ウィンドウで、「マッピングの選択 (Choose mapping)」の値が「COBOL からビジネス・オブジェクト (COBOL to Business Object)」であることを確認します。「参照」をクリックし、.ccp ファイル (この例では taderc99.ccp) を選択します。

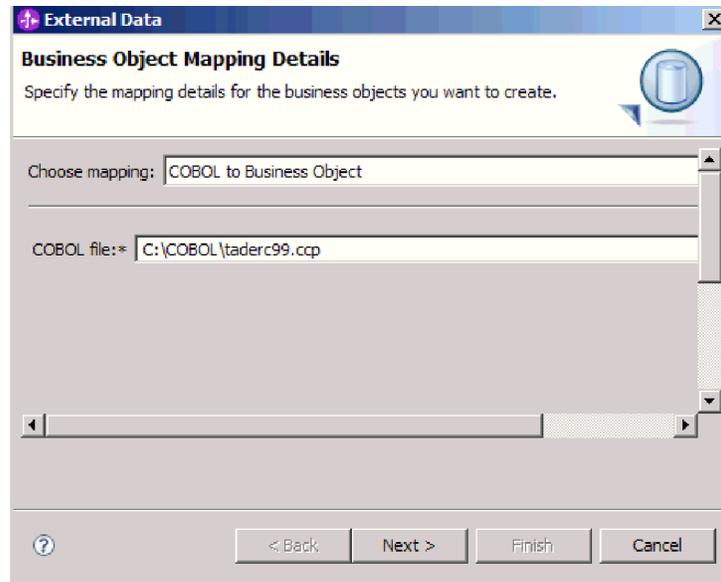


図 28. 「ビジネス・オブジェクト・マッピングの詳細 (Business Object Mapping Details)」 ウィンドウ

- 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」をクリックします。新規ビジネス・オブジェクト (図の「DFHCOMMAREA」) が表示されます。

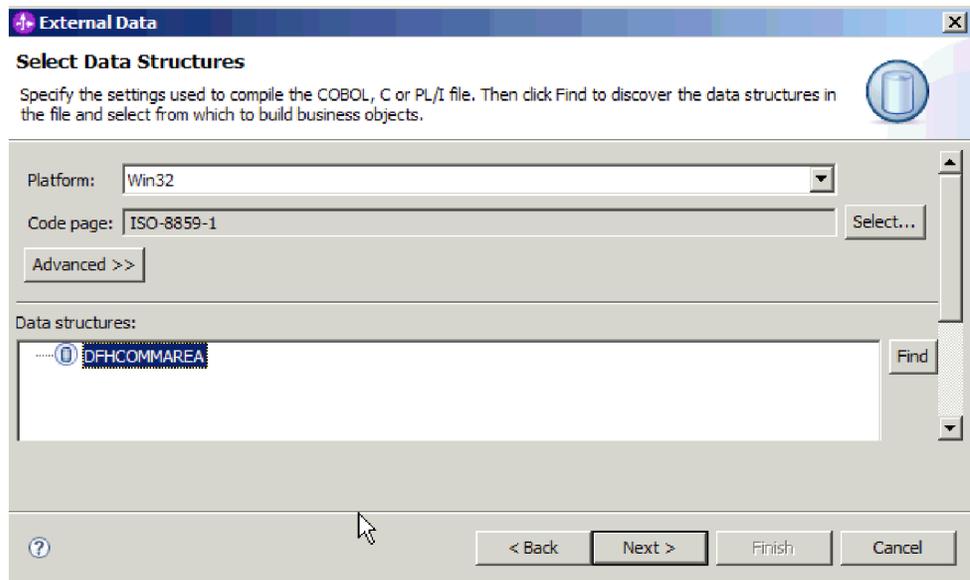


図 29. 「データ構造の選択 (Select Data Structures)」 ウィンドウ

- 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。

ビジネス・オブジェクト (図の「DFHCOMMAREA」) がモジュール内に作成されます。

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、既存のビジネス・オブジェクト定義を追加機能でラップします。 wrapper ビジネス・オブジェクト定義の生成オプションは、アダプター RAR ファイルがワークスペースにインポート済みである場合に限り表示されます。
 - 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 (External Data)」を選択します。
 - b. 「ビジネス・オブジェクト」ウィンドウで、「wrapper ビジネス・オブジェクトの作成 (Create a wrapper business object)」を選択します。

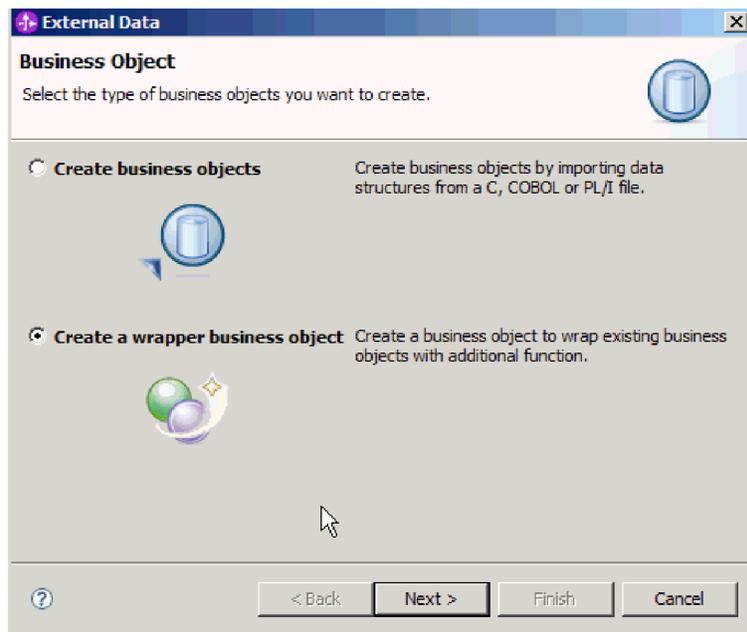


図 30. 「ビジネス・オブジェクト」ウィンドウ

- c. 「次へ」をクリックします。
- d. 「アダプターの選択 (Select an Adapter)」ウィンドウで、新規ビジネス・オブジェクトを保存したコネクタ・プロジェクトを選択し、「次へ」をクリックします。
- e. 「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、ステップ 1 で作成したビジネス・オブジェクト (例: DFHCOMMAREA) をデータ・タイプとして選択します。
- f. 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。

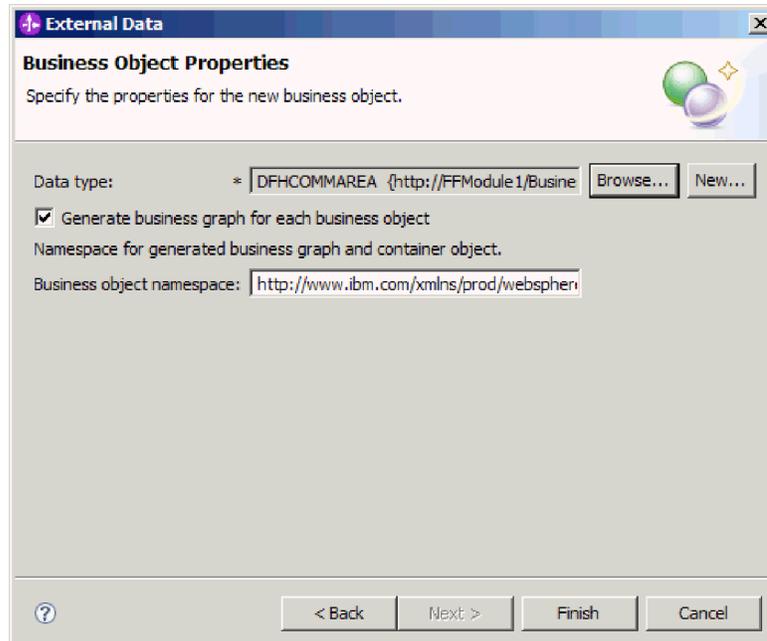


図 31. 「ビジネス・オブジェクト・プロパティ」ウィンドウ

g. 「終了」をクリックします。

wrapper ビジネス・オブジェクトとビジネス・グラフ (それぞれ、図の DFHCOMMAREAWrapper と DFHCOMMAREAWrapperBG) が、「ビジネス・インテグレーション」ウィンドウ内の現行モジュールにリストされます。

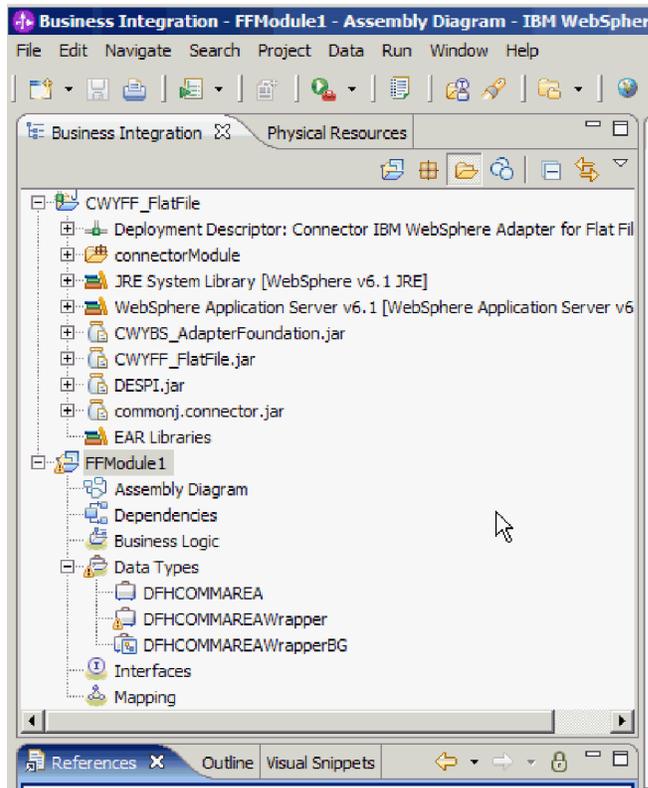


図 32. 「ビジネス・インテグレーション」ウィンドウにリストされた *wrapper* ビジネス・オブジェクトとビジネス・グラフ

3. COBOL コピーブックの Inbound モジュールに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 (External Service)」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 (Select an Adapter)」ウィンドウで、今作成した新規ビジネス・オブジェクトを保存したコネクタ・プロジェクトを選択し、「次へ」をクリックします。

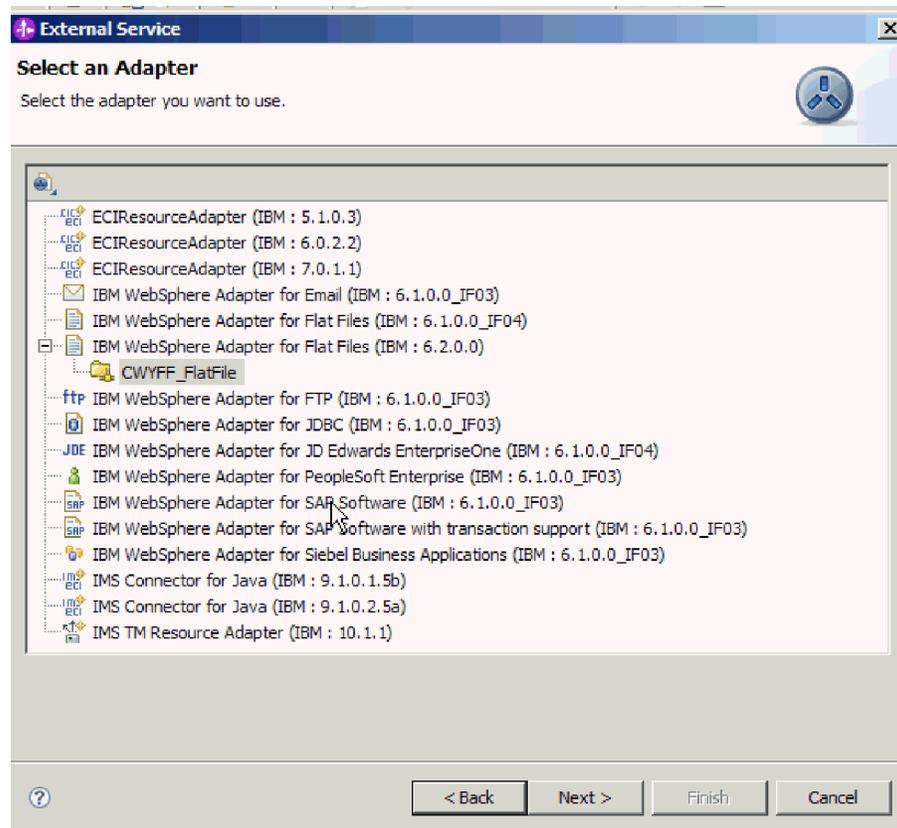


図 33. 「アダプターの選択 (Select an Adapter)」 ウィンドウ

- d. 「処理指示 (Processing Direction)」 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、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」で、デフォルト値を選択します。
- g. 「データ・バインディング」で、「**COBOL、C、または PL/I データ・バインディング・オプションを使用 (Use COBOL, C or PL/I data binding option)**」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。このツールを使用して、現行モジュールで適切なデータ・バインディング・コードを生成します。

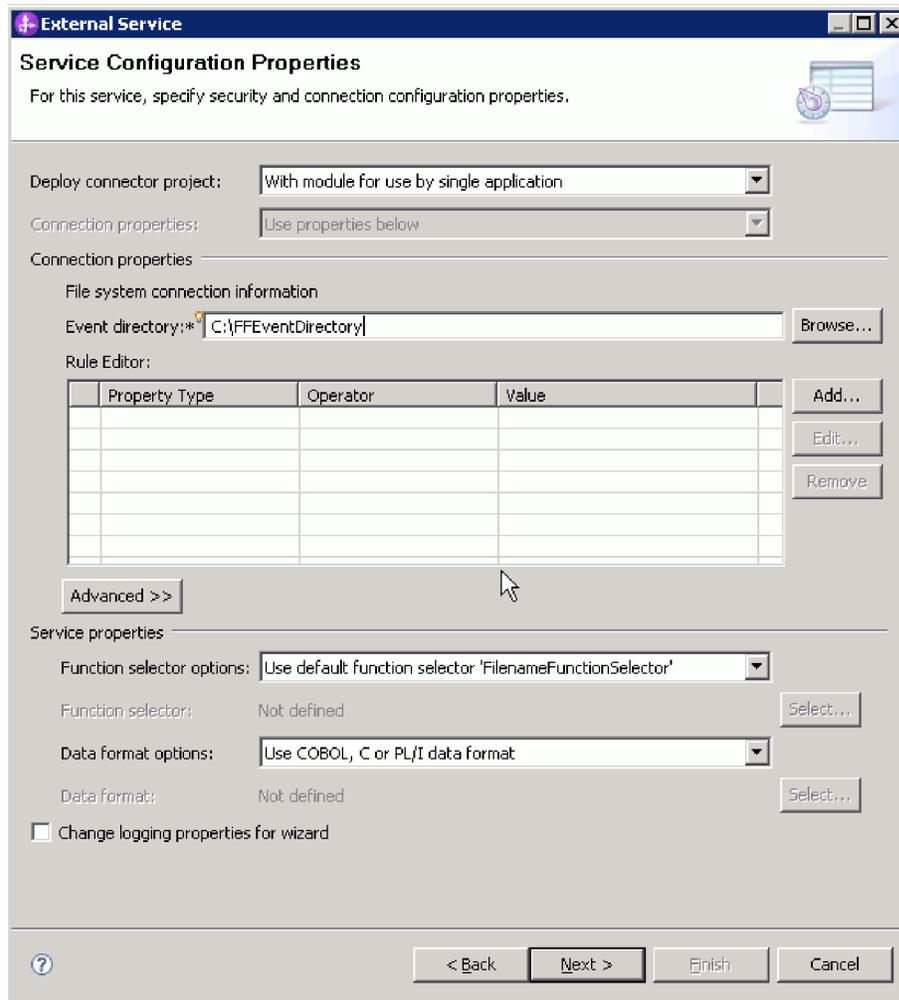


図 34. 「サービス構成プロパティ」 ウィンドウ

- 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・ファイルが含まれている場合は、サイズまたは区切り文字でのファイル分割を有効にすることができます。ファイル分割を有効にするには、「**拡張**」をクリックし、「**拡張プロパティ**」をクリックします。サイズでのファイル分割を有効にするには、それぞれの 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。ビジネス・オブジェクトをテキスト・エディターで開き、最大長を追加することも、ファイル上部で DFHCOMMAREA のコンテンツ・サイズを探すこともできます。212 ページの『ファイル内容を分割するための基準の指定』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、日付タイプとして「**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. 入力タイプとして、「**参照**」をクリックして、生成済みビジネス・オブジェクト (DFHCOMMAREA) を選択します。「**OK**」をクリックします。

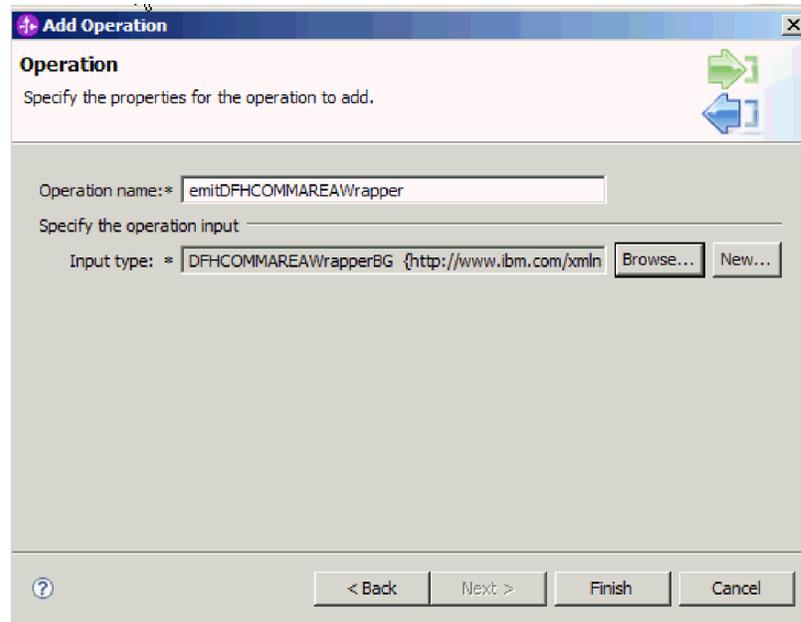


図 35. 「データ・タイプの選択 (Data Type Selection)」 ウィンドウでの入力タイプの選択

- m. 「終了」をクリックします。
- n. 「次へ」、「終了」の順にクリックします。

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。生成済みデータ・バインディング・クラスについては、「プロジェクト・エクスプローラー (Project Explorer)」 ウィンドウを参照してください。

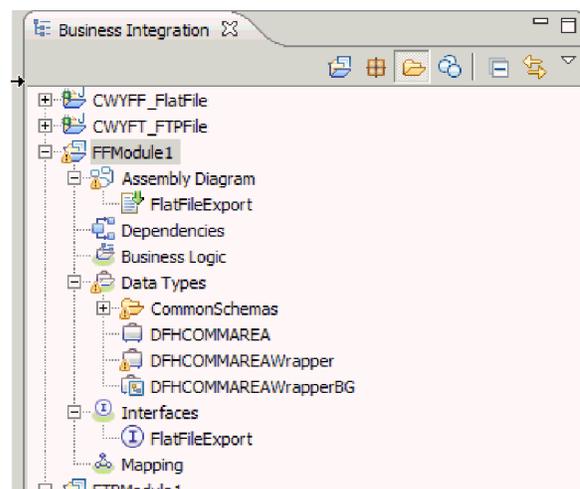


図 36. COBOL コピーブック、WSDL ファイル、エクスポート・ファイル、およびその他の成果物によって使用されるデータ・バインディング

結果

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、Inbound モジュールの COBOL プログラム・ソース・ファイル用に作成されます。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

に成果物が生成されます。このモジュールを、WebSphere Process Server または WebSphere Enterprise Service Bus にデプロイして、Inbound 操作をテストできます。

次のタスク

モジュールをデプロイします。

関連資料

198 ページの『活動化仕様プロパティ』

活動化仕様プロパティは、エクスポート用の Inbound イベント処理の構成情報を保持しています。活動化仕様プロパティは、外部サービス・ウィザードまたは管理コンソールのいずれかを使用して設定します。

アダプター・パターン・ウィザードを使用した単純なサービスの作成

アダプター・パターンは、アダプターで単純なサービスを素早く簡単に作成する方法を提供します。

始める前に

RetrieveAFileModule という名前のモジュールと、Customer という名前のビジネス・オブジェクトが既に作成されています。WebSphere Application Server 環境変数を使用してローカル・ファイルおよびディレクトリーを指定する場合、それらは既に WebSphere Process Server 管理コンソールを使用して定義済みです。

このタスクを実行する理由および時期

以下のアダプター・パターンが、Adapter for Flat Files で使用可能です。

表 10.

アダプター・パターン	説明
Inbound Flat File パターン	Flat File Inbound パターンは、ローカル・ファイル・システム上の特定のディレクトリーにあるファイルを取り出すサービスを作成します。ファイルの形式が XML 以外の場合は、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データ・ハンドラーを指定できます。ファイル内容に処理対象のデータ構造のコピーが複数含まれている場合は、その内容を分割することができます。
Outbound Flat File パターン	Flat File Outbound パターンは、ローカル・ファイル・システム上の特定のディレクトリー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。必要な出力形式が XML 以外の場合は、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データ・ハンドラーを指定できます。

ここでは、処理対象のファイル・システムからファイルを受け取る Flat File Inbound サービスの作成例を示します。この例の完全なサービスでは、ファイルを読み取り、区切り文字に基づいて内容を個別のファイルに分割します。

以下のステップを実行して、アダプター・パターン・ウィザードでサービスを作成します。

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「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. 「ローカル・ファイルから読み取る Inbound Flat File サービスを作成 (Create an inbound Flat File service to read from a local file)」を選択して、「次へ」をクリックします。

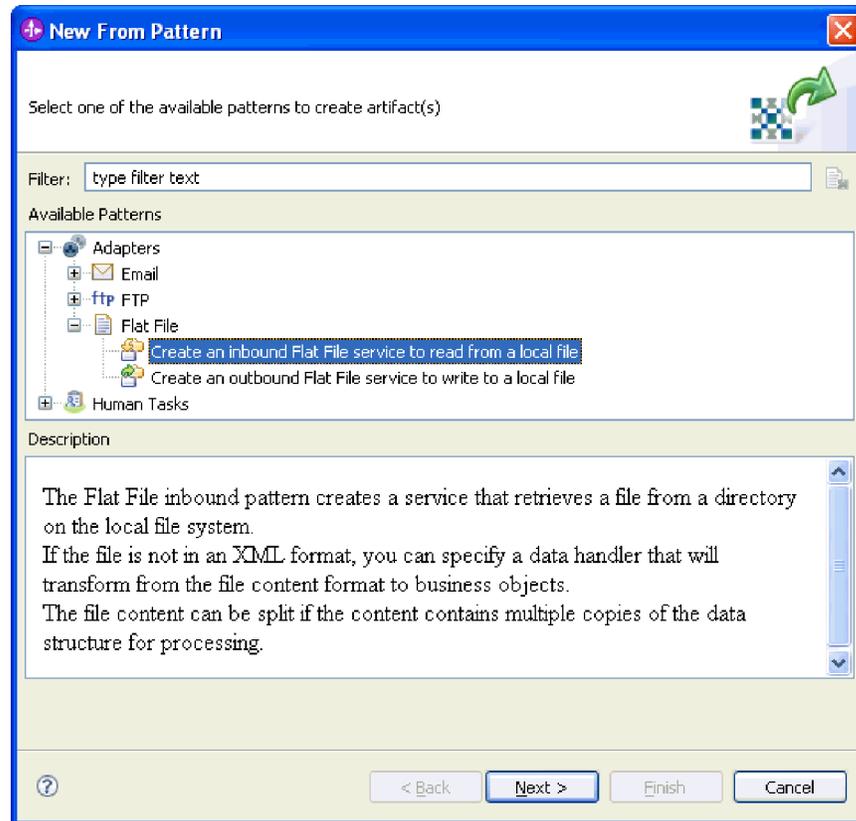


図 37. 「パターンから新規作成」ウィンドウ

3. 「新規 Inbound Flat File サービス (New Inbound Flat File Service)」ウィンドウで、名前を「FlatFileInboundInterface」などの有効な名前に変更してから「次へ」をクリックします。

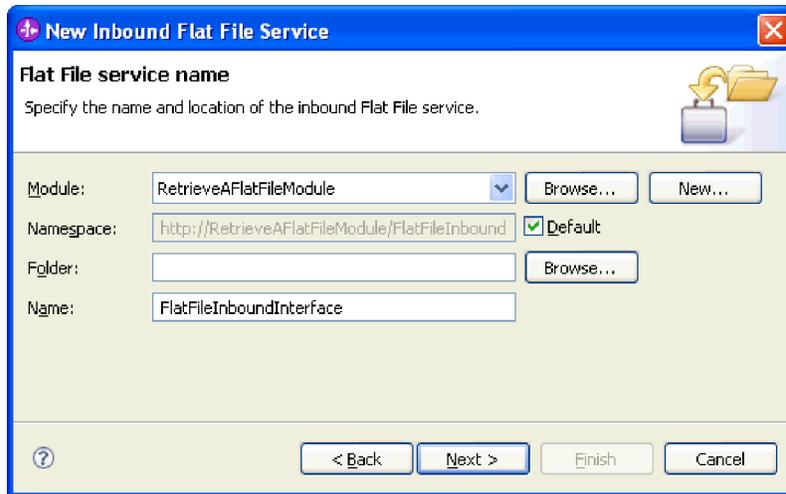


図 38. 「Flat File サービス名 (Flat File service name)」 ウィンドウ

4. 「ビジネス・オブジェクトおよびディレクトリー (Business object and directory)」 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. 入力ファイルを配置したディレクトリー (この場合は、「FFInboundEvents」ディレクトリー) を指定して、「次へ」をクリックします。この値として WebSphere Application Server 環境変数を使用するには、環境変数の名前を中括弧で囲み、前に \$ 記号を付けて指定します。例えば、\${FFINBOUNDEVENTS} のようになります。

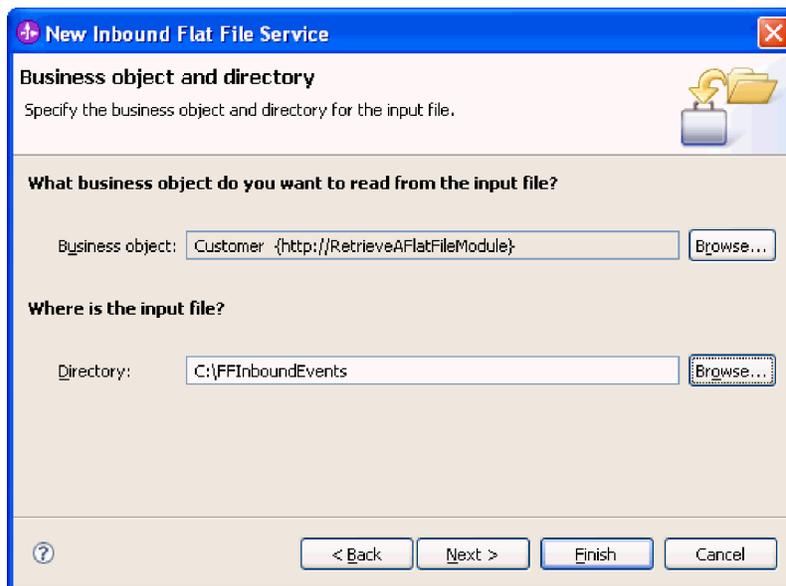


図 39. 「ビジネス・オブジェクトおよびディレクトリー (Business object and directory)」 ウィンドウ

6. 「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」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、ネイティブ形式からビジネス・オブジェクト形式にデータを変換するデータ・ハンドラーを指定します。
7. 「区切り文字でファイル内容を分割する」を選択し、使用する区切り文字（この場合は、「###;¥r¥n」）を入力します。「次へ」をクリックします。

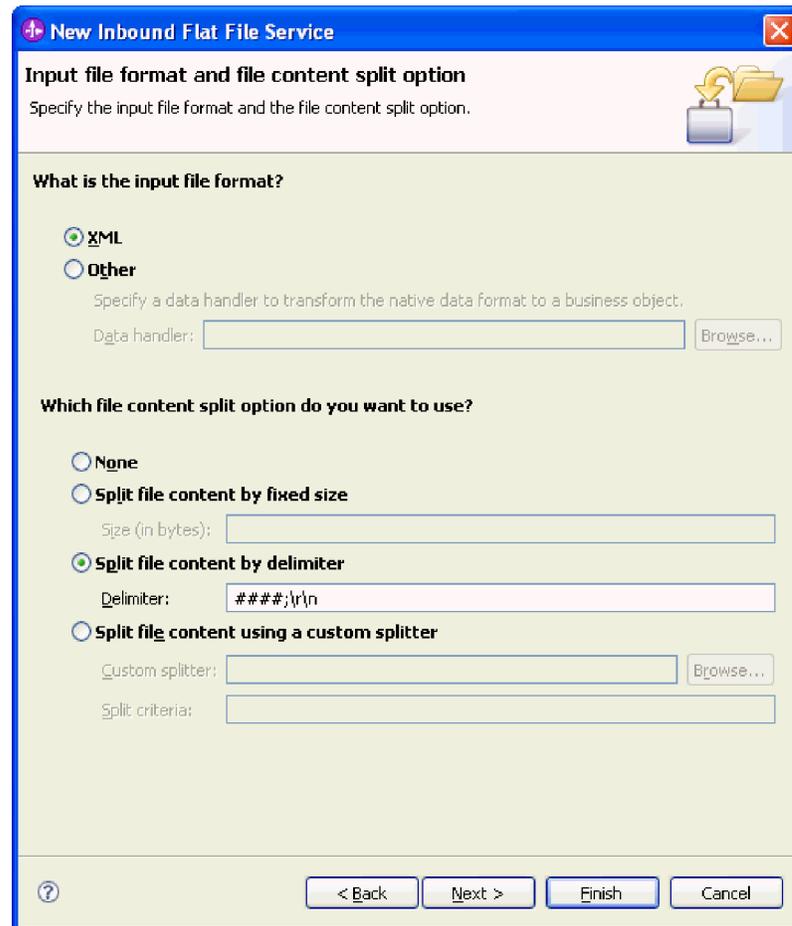


図 40. 「入力ファイル・フォーマットとファイル内容の分割オプション」ウィンドウ

8. 「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」（この場合は、「FFInboundArchive」）を指定します。この値として WebSphere Application Server 環境変数を使用するには、環境変数の名前を中括弧で囲み、前に \$ 記号を付けて指定します。例えば、\${FFINBOUNDARCHIVE} のようになります。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。

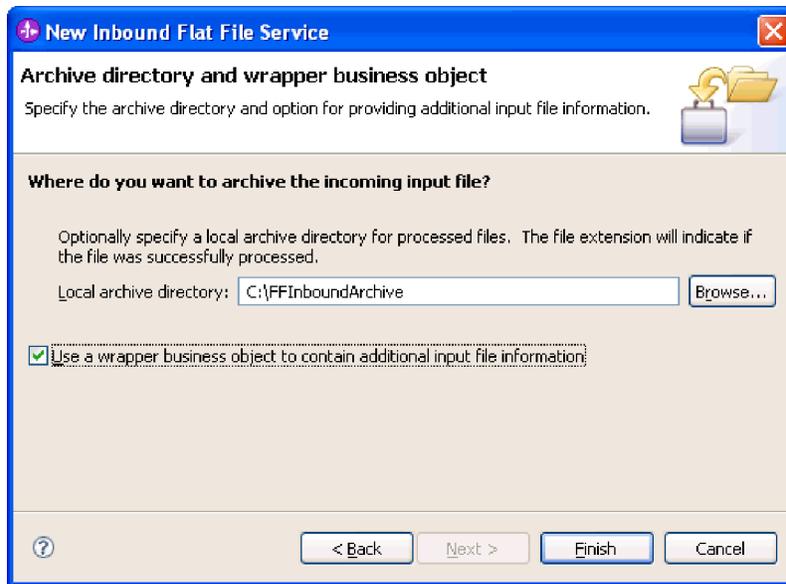


図 41. 「アーカイブ・ディレクトリーおよびラッパー・ビジネス・オブジェクト」ウィンドウ

結果

以下の成果物を含む Inbound サービスが作成されます。

表 11.

成果物	名前	説明
エクスポート	FlatFileInboundInterface	エクスポートは、モジュールを外部に公開します。ここでは、WebSphere Adapter for Flat Files が対象になります。
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、名前、住所、都道府県などのお客様データを入力するフィールドが含まれます。CustomerWrapper ビジネス・オブジェクトには、アダプター固有の情報を入力する追加フィールドが含まれます。
インターフェース	FlatFileInboundInterface	このインターフェースには、起動可能な操作が含まれます。
操作	emitCustomerInput	emitCustomerInput は、インターフェースにある唯一の操作です。

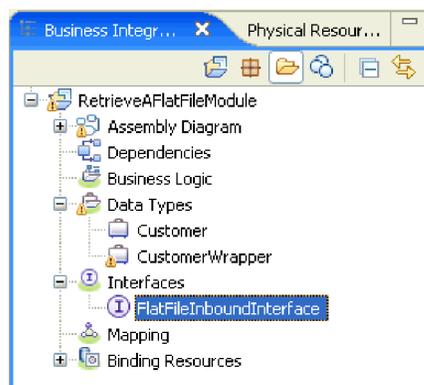


図 42. 新規成果物を表示する「WebSphere Integration Developer」ウィンドウの「ビジネス・インテグレーション」セクション

プロジェクトの作成

モジュールの作成およびデプロイを行うには、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、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するプロジェクトを作成します。

このタスクを実行する理由および時期

外部サービス・ウィザードを開始して、WebSphere Integration Developer のアダプター用のプロジェクトを作成します。既存プロジェクトがある場合は、ウィザードで新規作成する代わりに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. 「新規の外部サービス (New external service)」ウィンドウで、「アダプター (Adapters)」が選択済みであることを確認して、「次へ」をクリックします。

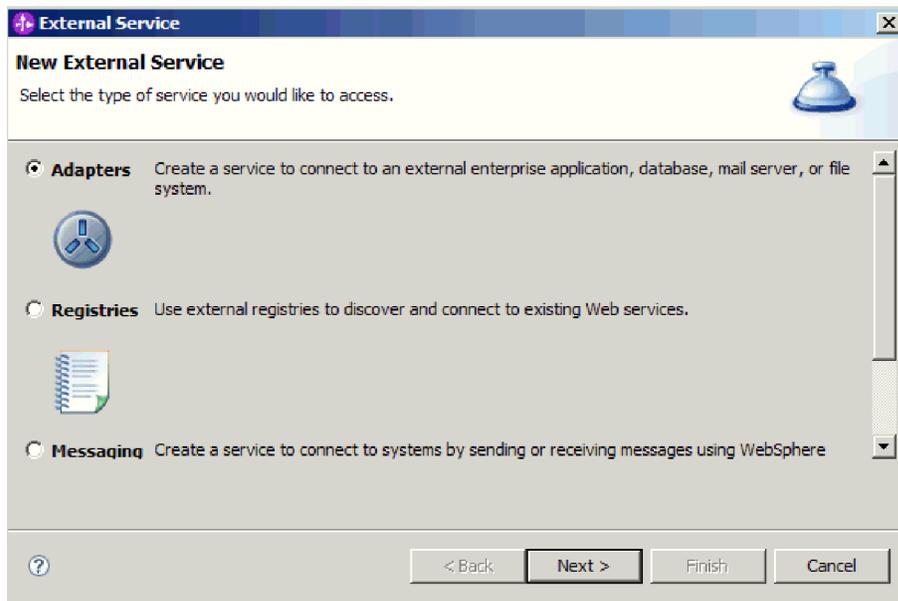


図 43. 「新規の外部サービス (New external service)」 ウィンドウ

3. 「エンタープライズ・サービス・リソース・アダプターを選択 (Select an Enterprise Service Resource Adapter)」 ウィンドウから、プロジェクトを作成または既存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 Flat Files**」を選択し、「次へ」をクリックします。
 - b. 「コネクター・インポート」ウィンドウで、プロジェクトの別名を指定し (「**CWYFF_FlatFile**」 以外の名前を使用するため)、サーバーを選択して (例えば、「**WebSphere Process Server v6.1**」)、 「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、以下の手順を実行します。
 - a. 「**IBM WebSphere Adapter for Flat Files**」を展開します。
 - b. プロジェクトを選択します。

例えば、CWYFF_FlatFiles という名前の既存プロジェクトがある場合、以下の図に示すように、「**IBM WebSphere Adapter for Flat Files**」を展開し、「**CWYFF_FlatFile**」を選択できます。

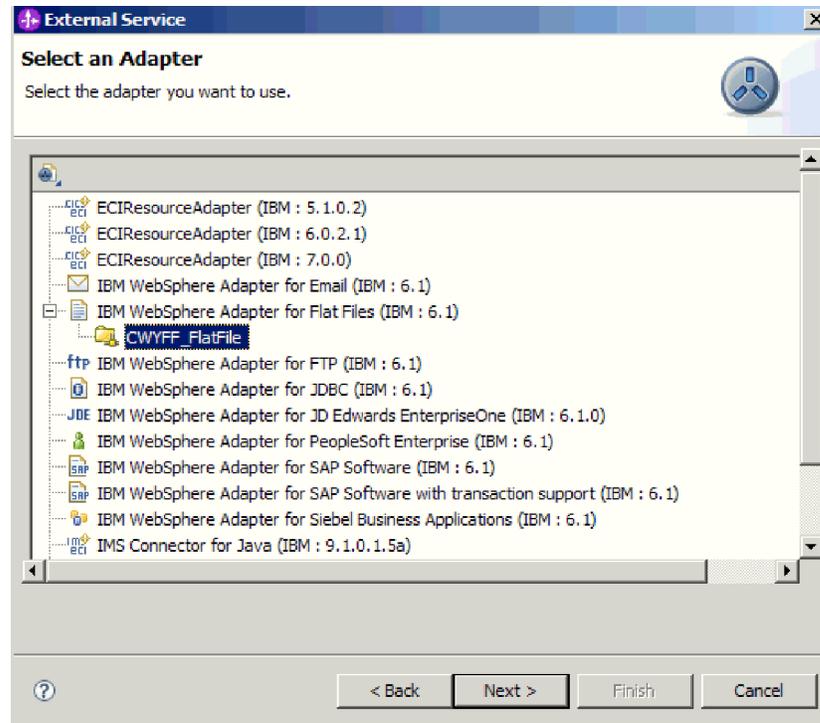


図 44. 「エンタープライズ・サービス・リソース・アダプターの選択」ウィンドウ

- c. 「次へ」をクリックします。

結果

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

次のタスク

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。

関連概念

3 ページの『Outbound 処理』

アダプターは、Outbound 処理中に、ローカル・ファイル・システム 内のファイル上で処理を実行するために、ビジネス・オブジェクトの形式でモジュールから要求を受け取ります。アダプターは要求された操作を実行して、該当する場合は、操作の結果を表すビジネス・オブジェクトをコンポーネントに返します。

デプロイメントおよびランタイム・プロパティの設定

ローカル・ファイル・システムとの Outbound または Inbound 通信でモジュールを使用することを決定した後、モジュールと ローカル・ファイル・システム間の接続をアダプターが確立できるように、Managed Connection Factory プロパティを構成する必要があります。

始める前に

このセクションでプロパティを設定するには、その前にアダプター・モジュールを作成しておく必要があります。これは、WebSphere Integration Developer ではアダプター・プロジェクトの下に表示されます。アダプター・プロジェクトの作成について詳しくは、87 ページの『プロジェクトの作成』を参照してください。

このタスクを実行する理由および時期

デプロイメントおよびランタイム・プロパティを設定するには、次の手順に従います。このトピックに記載されているプロパティについて詳しくは、177 ページの『Managed Connection Factory プロパティ』を参照してください。

このタスクの手順

1. 「処理指示 (Processing Direction)」 ウィンドウで「**Outbound**」を選択し、「次へ」をクリックします。

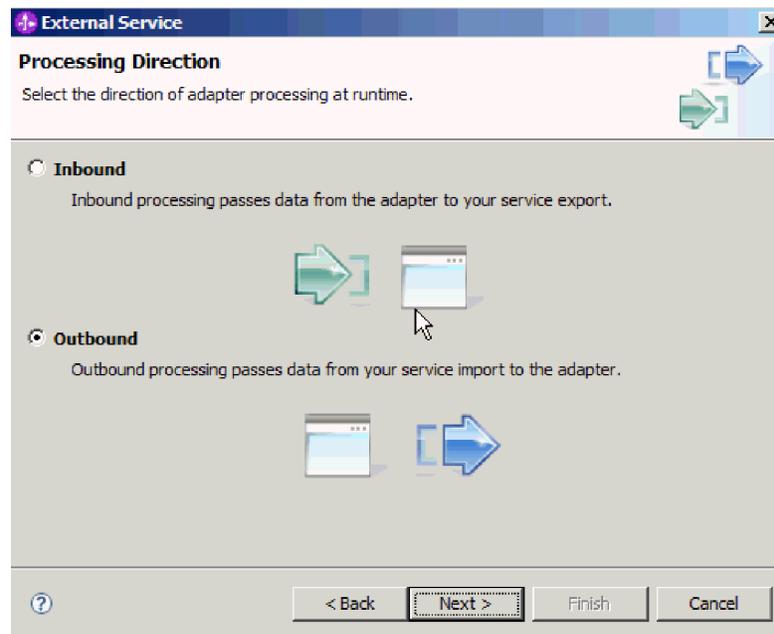


図 45. 外部サービス・ウィザードでの Inbound または Outbound 処理の選択

2. 「サービス構成プロパティ」ウィンドウの「コネクター・プロジェクトのデプロイ」フィールドで、「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)」を選択します。
3. モジュールの接続プロパティを定義します。このウィンドウに表示されるプロパティについて詳しくは、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。

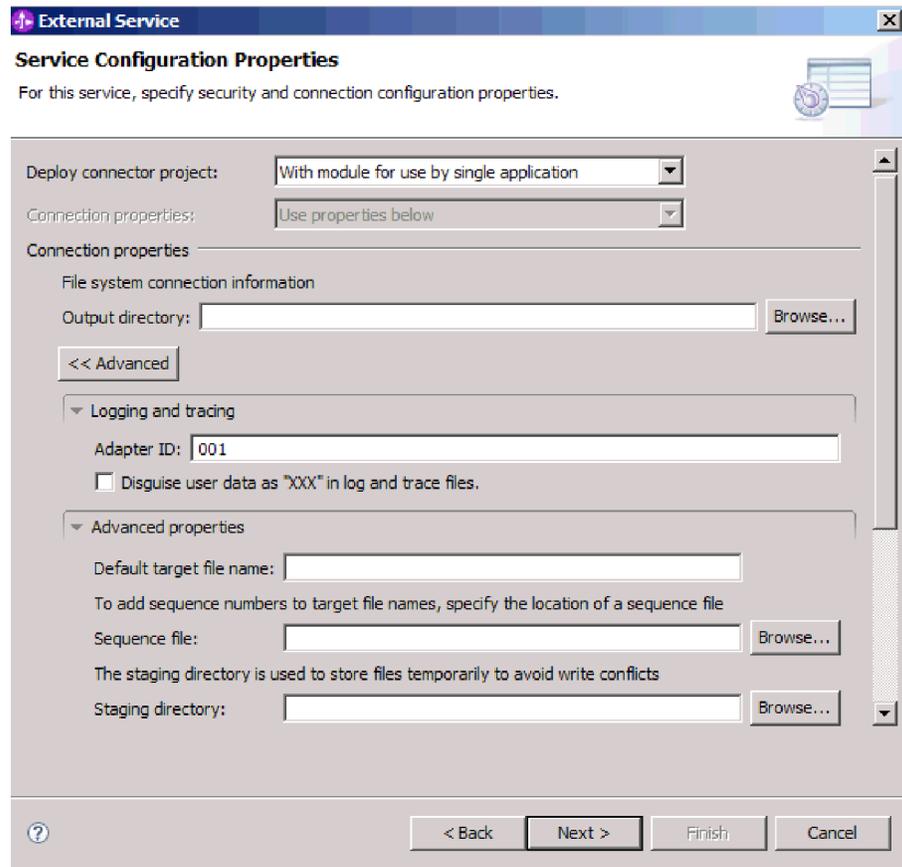


図 46. 接続プロパティの設定

- オプション: アダプターの複数インスタンスがある場合は、「**ロギングおよびトレース**」を展開し、アダプター ID をこのインスタンスの固有な値に設定します。このプロパティについて詳しくは、Managed Connection Factory プロパティに関する参照トピックを参照してください。
- 一定の情報を隠して、その情報がログまたはトレースに表示されないようにするには、「**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。
- オプション: ログ・ファイルの出力先を指定したり、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、「**ウィザードのロギング・プロパティを変更します (Change logging properties for wizard)**」チェック・ボックスを選択します。ロギング・レベルの設定については、『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』のセクションを参照してください。
- 「次へ」をクリックします。

結果

アダプターにより接続プロパティが保存されます。

次のタスク

モジュールのデータ・タイプと、選択したデータ・タイプに関連付ける操作の名前を指定します。

関連資料

173 ページの『ウィザードの接続プロパティ』

接続プロパティはサービス記述を作成して、組み込みの成果物を保存するために使用されます。これらのプロパティは、外部サービス・ウィザードで構成されます。

操作およびデータ・タイプの選択

外部サービス・ウィザードを使用して、ローカル・ファイル・システムの機能にアクセスするために使用される Outbound 操作と、その操作で使用されるデータ・タイプを選択します。サポートされる操作は、

Create、Append、Overwrite、Delete、Exists、List、および Retrieve です。外部サービス・ウィザードでは、3 種類 (汎用 FlatFile ビジネス・オブジェクト、ビジネス・グラフ付きの汎用 FlatFile ビジネス・オブジェクト、およびユーザー定義タイプ) の中からデータ・タイプを選択できます。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、ローカル・ファイル・システム 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

Outbound 操作およびその操作で使用されるデータ・タイプを選択するには、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、「追加」をクリックします。

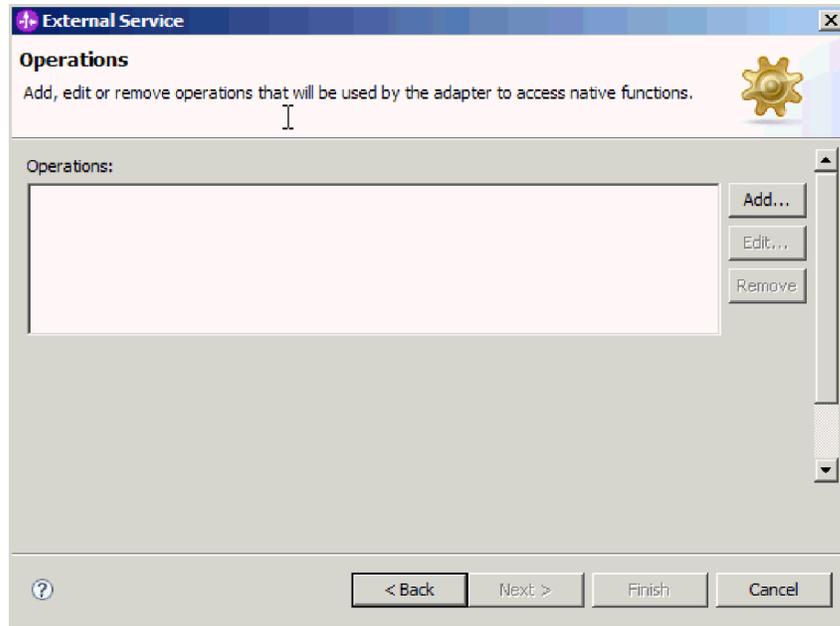


図 47. 操作の追加

2. 「操作の追加 (Add Operation)」ウィンドウで、「**操作の種類**」リストを開いて操作を選択します。この例では、「**Create**」を選択します。
3. 「操作の追加 (Add Operations)」ウィンドウで、データ・タイプを選択し、「次へ」をクリックします。この例では、「**ユーザー定義 (User defined)**」タイプを選択します。

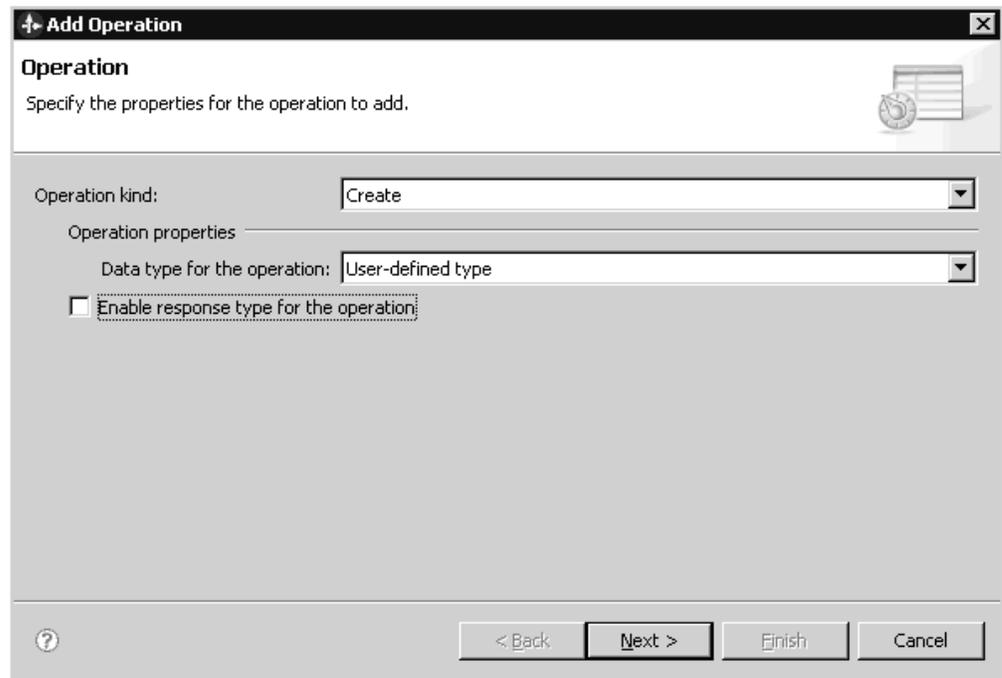


図 48. 操作のデータ・タイプの選択

Delete、Retrieve、Exists、および List 操作の場合には、入力として汎用データ・タイプ (汎用 FlatFile ビジネス・オブジェクト、またはビジネス・グラフ付きの汎用 FlatFile ビジネス・オブジェクト) のみがサポートされます。これらの操作のいずれかで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。

Create、Append、および Overwrite 操作の場合には、ユーザー定義タイプ、汎用 FlatFile 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 FlatFile ビジネス・オブジェクトから選択します。データ・タイプについては、ビジネス・オブジェクト構造について説明している本書のトピックを参照してください。

4. オプション: Create、Append、および Overwrite 操作の場合には、ファイル名を戻す場合や固有ファイル名を生成する場合、またはファイル・シーケンスを有効にしている場合は、「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 操作の場合は出力が必要であるため、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。Delete 操作の場合、操作が成功した時に true の値が返されるようにしたい場合は、「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。

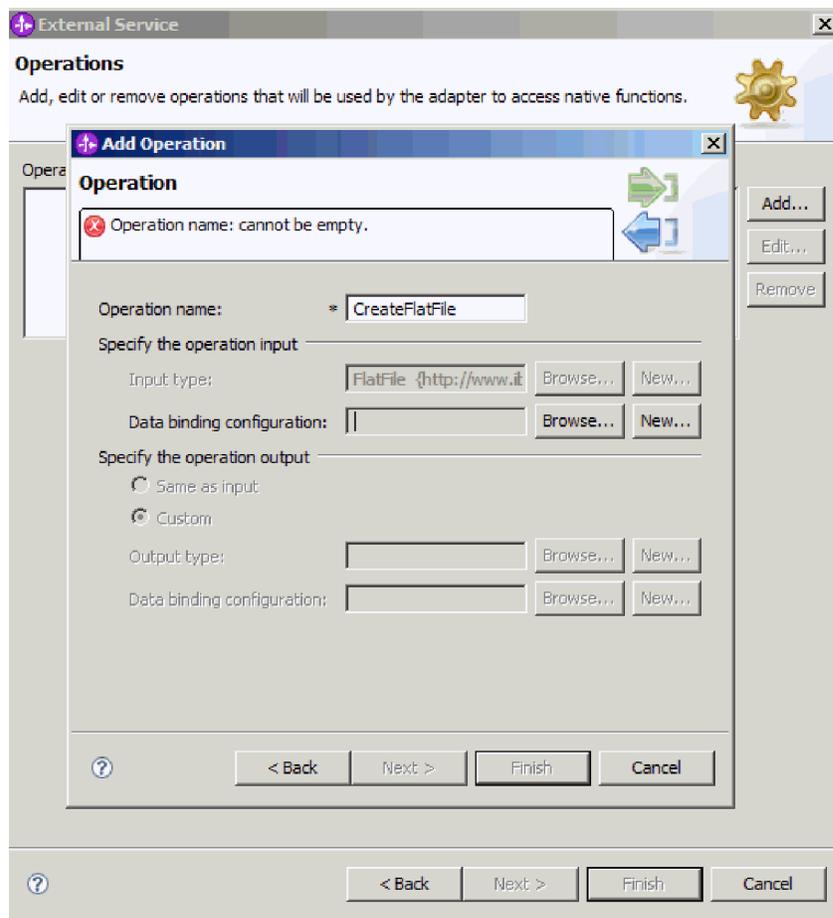


図 49. 操作の命名と入力タイプの指定

5. 「次へ」をクリックします。

6. 「操作の追加 (Add Operation)」画面で、「**操作名 (Operation name)**」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。アダプターが実行可能な操作のタイプについて詳しくは、本書のサポートされる操作に関する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。

デフォルトでは、出力のデータ・タイプが `CreateResponse` または `CreateResponseBG` に設定されています。

7. 入力タイプを選択します。「**参照**」をクリックし、以前に作成したビジネス・オブジェクトを選択します。汎用データ・タイプ (汎用 `FlatFile` ビジネス・オブジェクト、またはビジネス・グラフ付きの汎用 `FlatFile` ビジネス・オブジェクト) を指定した場合、入力タイプはデフォルトで `FlatFile` または `FlatFileBG` に設定されます。

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。

次のタスク

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。

データ・バインディングの構成

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、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。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

操作とその操作で使用されるデータ・タイプ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、`WebSphere Integration Developer` を使用して外部サービス・ウィザードを実行する前に構成できます。構成を行うには、`WebSphere Integration Developer` で「**新規**」→「**リソース構成 (Resource configuration)**」を選択し、本書で説明されているデータ・バインディング画面を完了してください。

このタスクの手順

1. 「操作の追加 (Add Operation)」ウィンドウの、「**Input 操作のデータ・バインディング構成 (operation input Data binding configuration)**」フィールドで「**新規**」

を選択します。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行います。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。

2. データ・バインディングの「名前」（この例では DBConfig を使用）を入力し、「次へ」をクリックします。

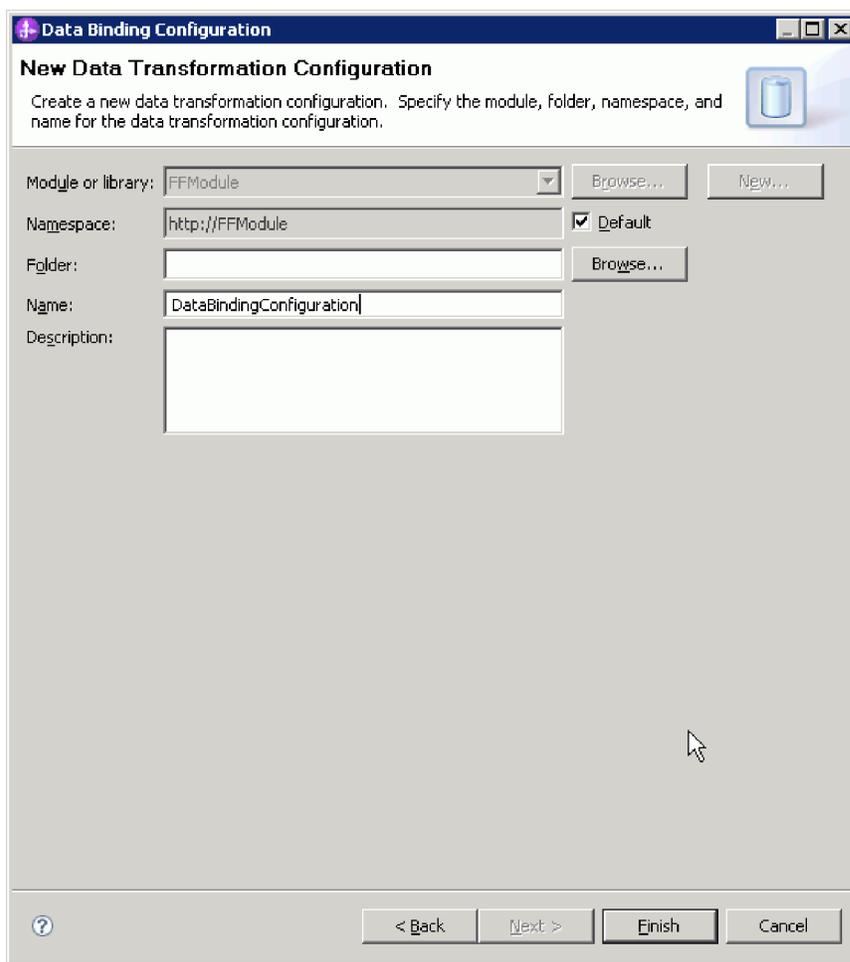


図 50. データ・バインディングの名前の指定

3. 「次へ」をクリックします。

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

データ・ハンドラー構成を選択します。

データ・ハンドラーの構成

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形式の間の変換を実行します。

始める前に

モジュールのデータ・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。また、WebSphere Integration Developer Business Object Editor を使用して、ビジネス・オブジェクトを事前に定義しておく必要があります。ここでウィザードを停止してビジネス・オブジェクトを作成する場合は、ウィザードのステップを最初から開始する必要があります。

注: データ・ハンドラーは、WebSphere Integration Developer を使用して外部サービス・ウィザードを実行する前に構成できます。構成を行うには、WebSphere Integration Developer で「新規」→「リソース構成 (Resource configuration)」を選択し、本書で説明されているデータ・ハンドラー画面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

ビジネス・オブジェクトのデータ・ハンドラーを指定するには、以下の手順を実行します。

このタスクの手順

1. 「操作の追加 (Add Operation)」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 (この例では DataBindingConfiguration を使用) を入力します。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。

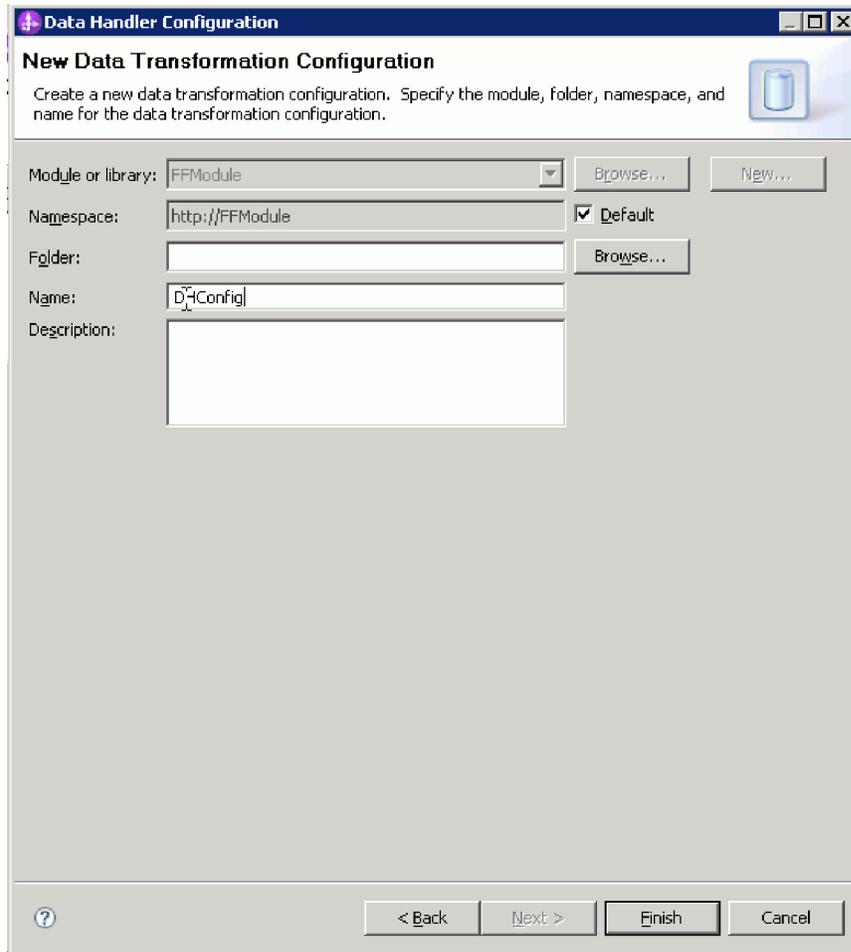
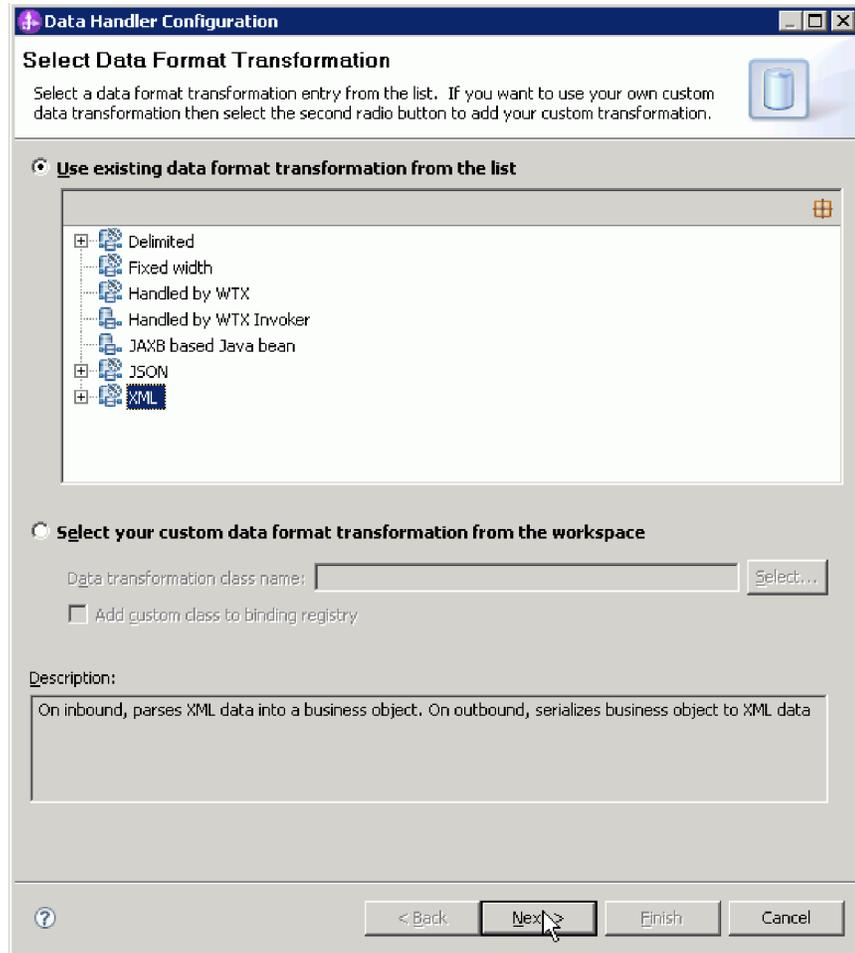


図 51. データ・ハンドラー構成の名前の指定

2. 「次へ」をクリックします。
3. 「データ・バインディング・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。「新規」をクリックして新しいデータ・ハンドラー構成を作成します。
4. 「新規」をクリックします。
5. 「新規データ・ハンドラー構成 (New Data Handler Configuration)」ウィンドウで、データ・ハンドラー構成のモジュール、ネーム・スペース、フォルダーおよび名前を指定します。
6. 「次へ」をクリックします。
7. データ・ハンドラーのクラス名を選択します。「データ形式変換の選択」ウィンドウで、「リストから既存のデータ形式変換を使用する」オプションをクリックします。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 52. 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。

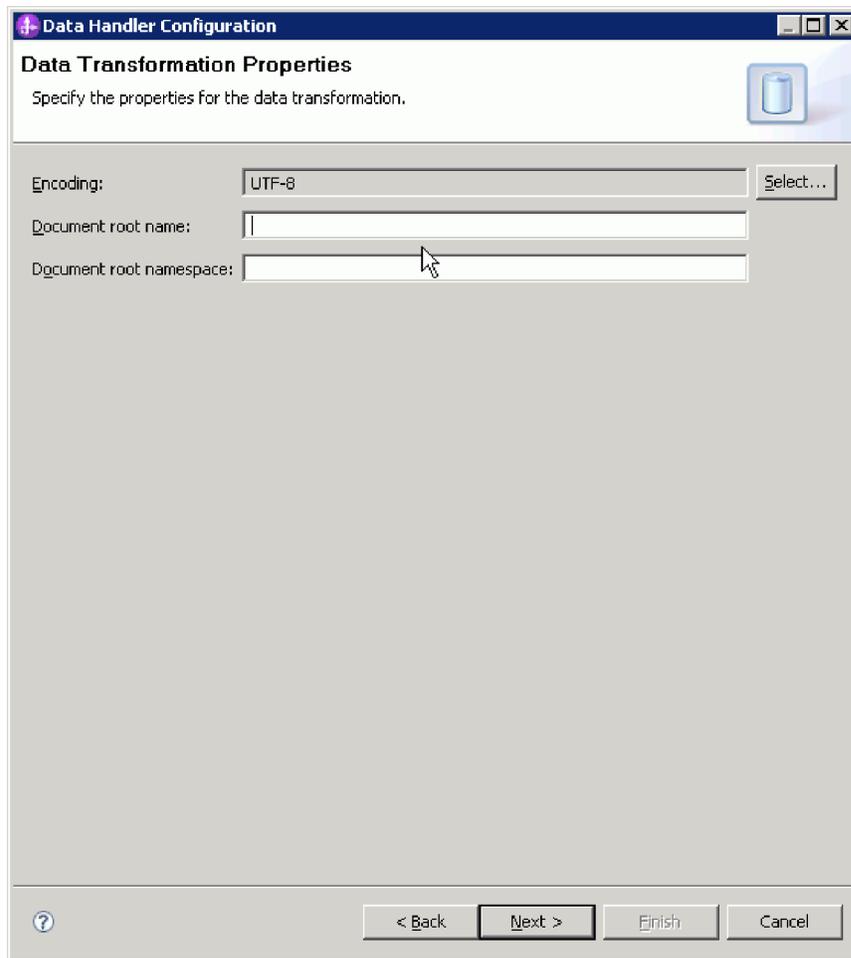


図 53. データ・ハンドラー構成のエンコード方式の指定

9. 「終了」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。「操作の追加 (Add operation) 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。

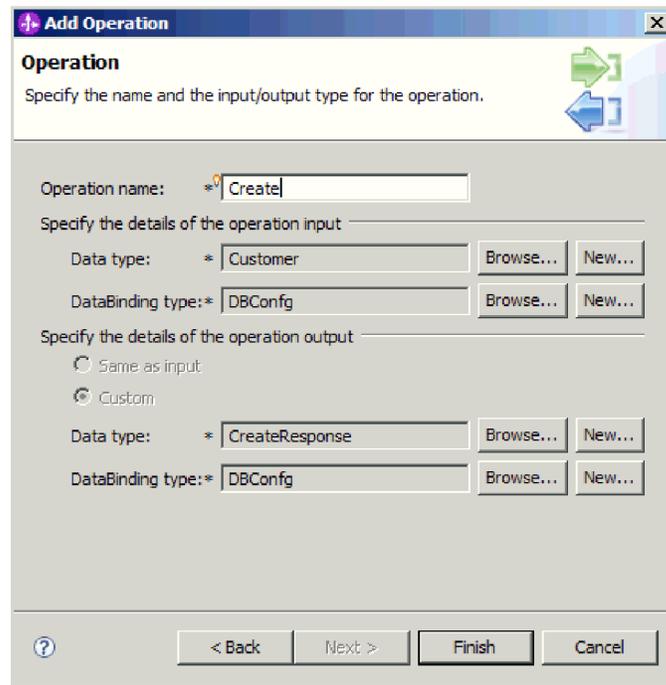


図 54. Output 操作でのデータ・バインディング構成の選択

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ータともに表示されます。

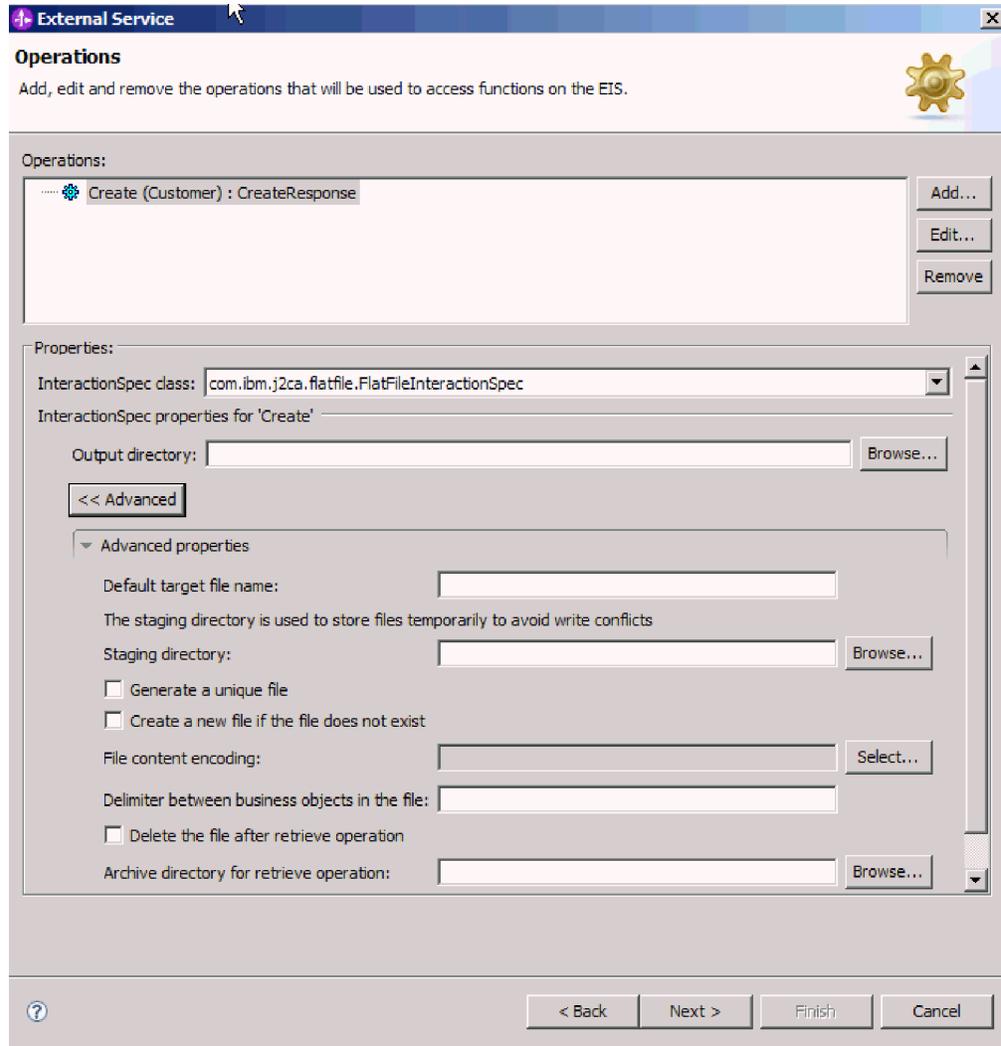


図 55. Create 操作と InteractionSpec プロパティ

12. 「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

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。

対話プロパティの設定およびサービスの生成

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

始める前に

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

このタスクを実行する理由および時期

対話仕様プロパティを設定してモジュールの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、本書の該当する参照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、以下のステップを実行します。
 - a. 「操作」ウィンドウで、「拡張」をクリックします。
 - b. デフォルトとして設定するすべてのフィールドに値を入力します。
 - c. 「次へ」をクリックします。

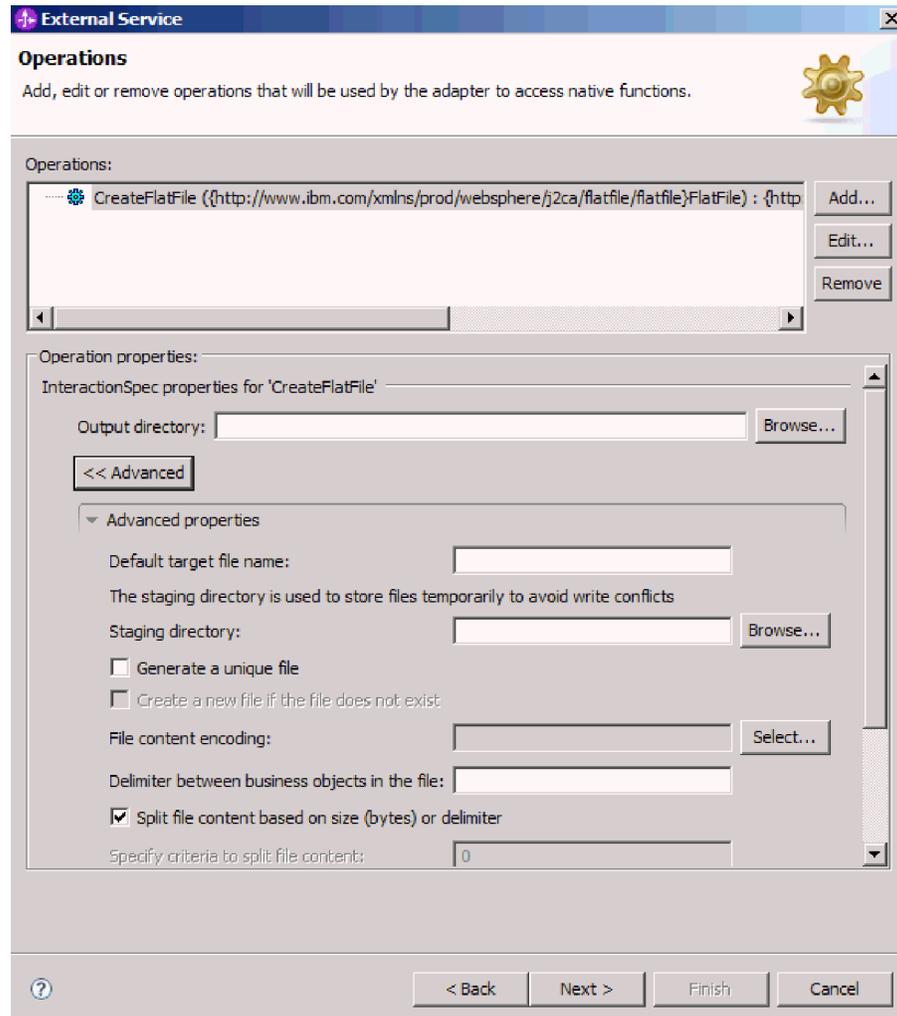


図 56. 対話仕様プロパティの設定

2. 「操作」ウィンドウで、「次へ」をクリックします。「サービスの生成 (Generate Service)」画面で、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せん。

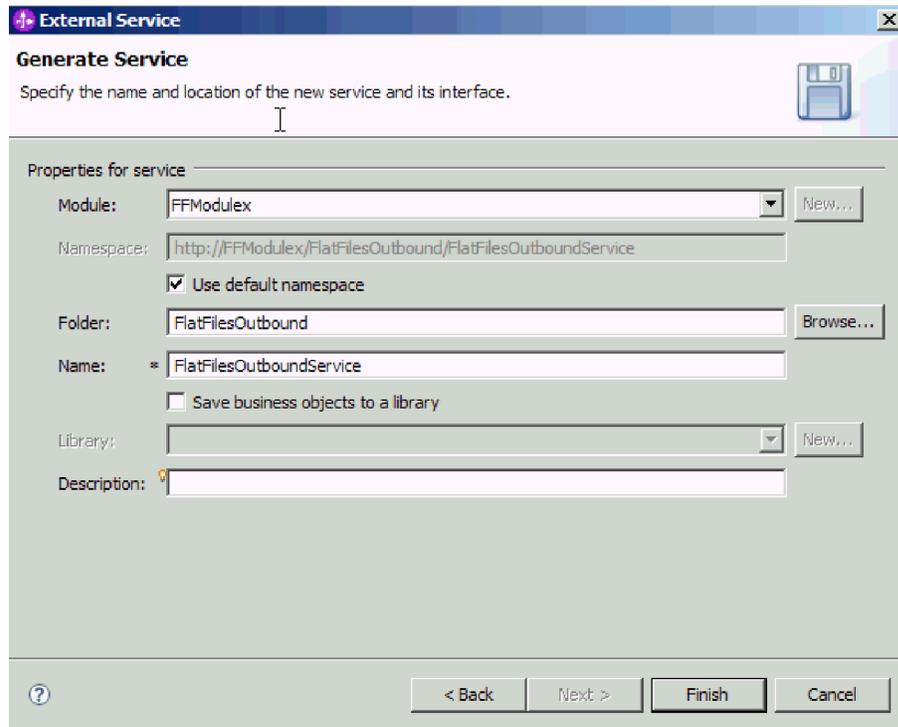


図 57. サービスの名前指定

3. 「終了」をクリックします。

結果

WebSphere Integration Developer がサービスおよびインポートを生成します。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

モジュールをデプロイします。

関連資料

171 ページの『Outbound 構成プロパティ』

WebSphere Adapter for Flat Files 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 Outbound 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用してモジュールを WebSphere Process Server または WebSphere Enterprise Service Bus にデプロイした後に、リソース・アダプターおよび Managed Connection Factory のプロパティを変更することができますが、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。

192 ページの『Inbound 構成プロパティー』

WebSphere Adapter for Flat Files 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ーがあります。WebSphere Integration Developer または管理コンソールを使用してモジュールをデプロイした後に、リソース・アダプターおよび活動化仕様のプロパティーを変更することができますが、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。

218 ページの『グローバル化』

WebSphere Adapter for Flat Files 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。

Inbound 処理のモジュールの構成

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

関連概念

15 ページの『Inbound 処理』

Adapter for Flat Files は Inbound イベント処理をサポートします。これは、イベント (ファイルの作成や変更など) に指定された間隔で、ローカル・ファイル・システムのポーリングを行います。イベントを検出すると、イベント・データをビジネス・オブジェクトに変換し、これをモジュールに送信して処理します。

関連資料

170 ページの『カスタム・ファイル分割』

分割論理を含むカスタム・クラスを実装することができます。アダプターは、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。

デプロイメントおよびランタイム・プロパティーの設定

エンタープライズ情報システム (ローカル・ファイル・システム) とのOutbound または Inbound 通信でご使用するモジュールを使用することを決定した後、エクスポート用の Inbound イベント処理構成情報を保持する活動化仕様プロパティーを構成する必要があります。

始める前に

このセクションでプロパティーを設定するには、その前にアダプター・モジュールを作成しておく必要があります。これは、WebSphere Integration Developer ではアダプター・プロジェクトの下に表示されます。アダプター・プロジェクトの作成について詳しくは、87 ページの『プロジェクトの作成』を参照してください。

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を設定するには、以下の手順に従います。このトピックに記載されているプロパティについては、198 ページの『活動化仕様プロパティ』を参照してください。

このタスクの手順

1. 「処理指示 (Processing Direction)」 ウィンドウで「**Inbound**」を選択し、「次へ」をクリックします。

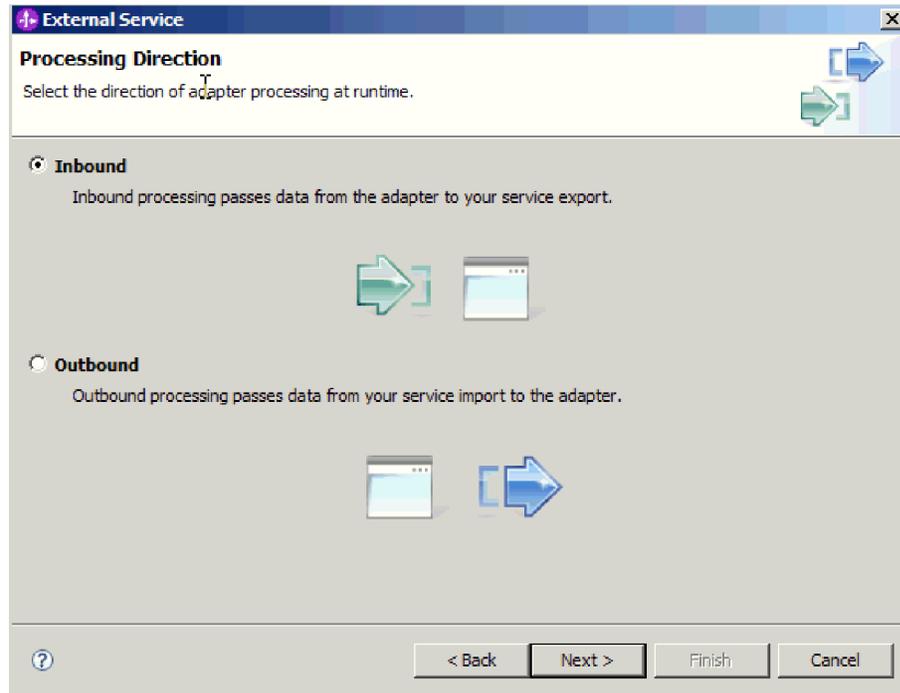


図 58. 外部サービス・ウィザードでの *Inbound* または *Outbound* の選択

2. 「サービス構成プロパティ」ウィンドウの「コネクター・プロジェクトのデプロイ」フィールドで、「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)」を選択します。
3. 「サービス構成プロパティ」ウィンドウで、モジュールの活動化仕様プロパティを定義します。このウィンドウに表示されるプロパティについては、198 ページの『活動化仕様プロパティ』を参照してください。

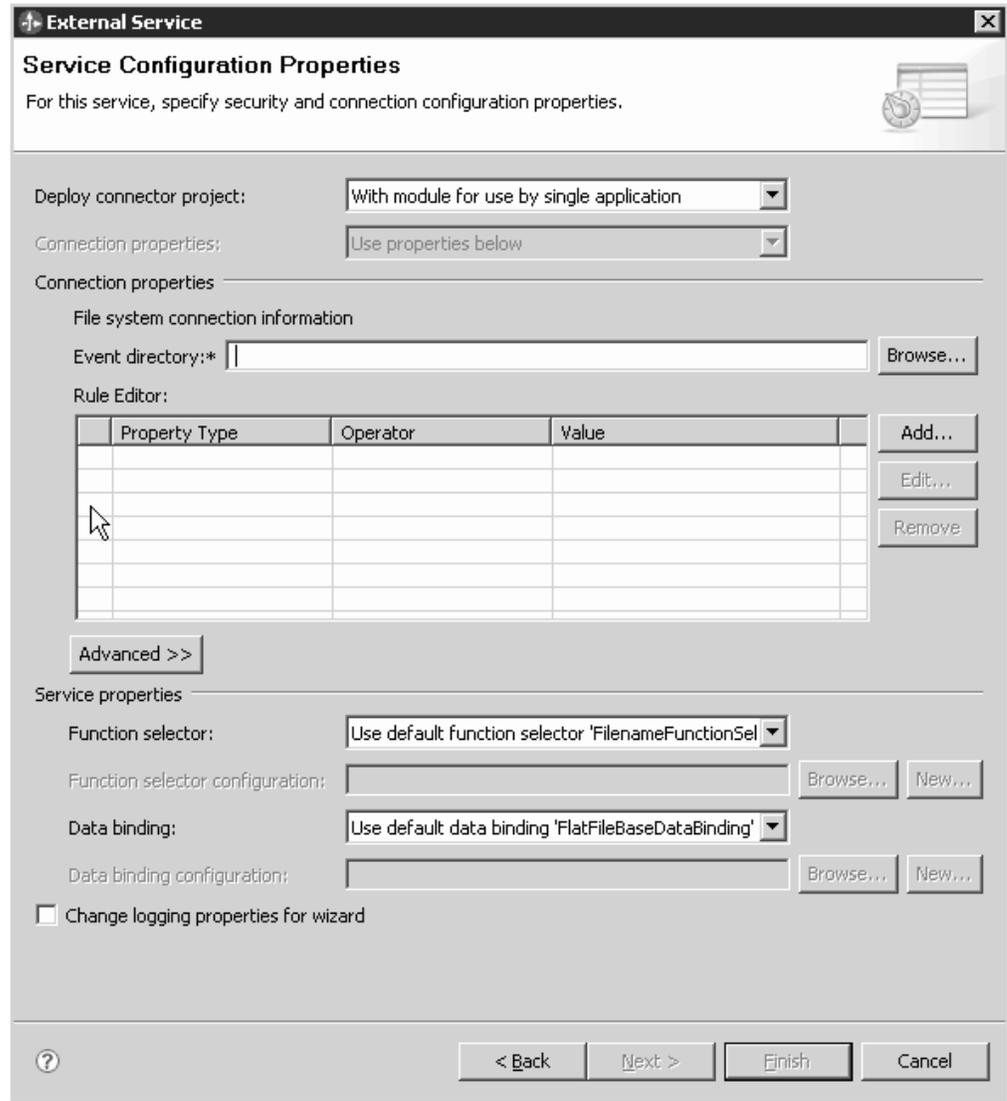


図 59. 接続プロパティの設定

4. 「イベント・ディレクトリー」プロパティでは、イベント・ファイルの保管先ローカル・ファイル・システムのディレクトリーを指定します。
5. 「拡張」をクリックして、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、拡張プロパティ、ファイル・アーカイブ構成、BIDI プロパティ、および追加プロパティを指定するロギングおよびトレースのセクションを展開します。
 - a. オプション: イベント・ポーリング構成セクションで、「開始時に EIS 接続を再試行する」を選択します。このプロパティを選択すると、アダプターは、始動時に接続できなかったシステムへの接続を継続して試行します。詳しくは、211 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。

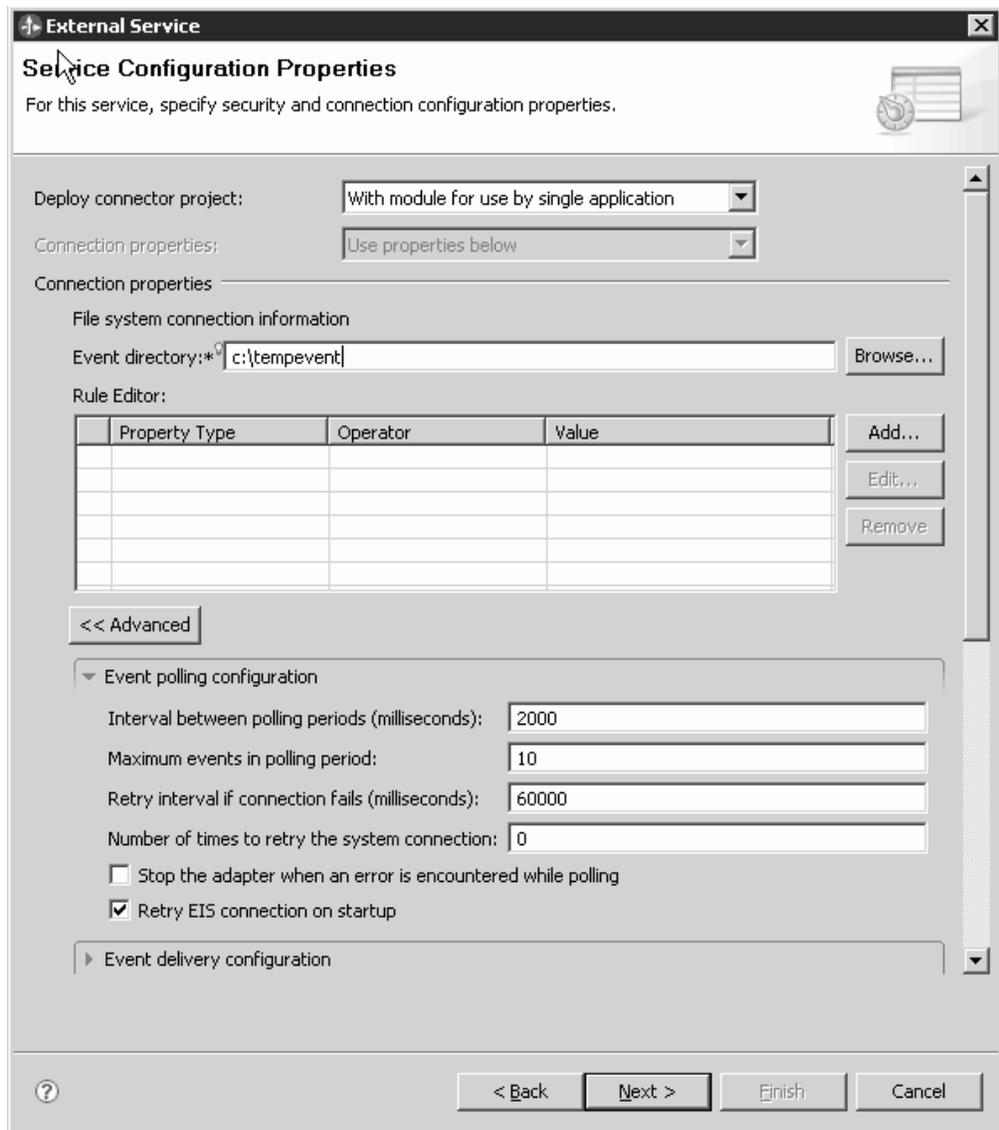


図 60. 「開始時に EIS 接続を再試行する」チェック・ボックスの選択

- b. オプション: 「拡張プロパティ」セクションで、「ファイル内容のエンコード」フィールドの値を選択します。バイナリー・イベント・データを処理する場合は、「BINARY」を選択します。テキストや XML などのバイナリー以外のイベント・データを処理する場合は、UTF-8 (デフォルト値) などの有効なファイル・エンコード方式の値に設定します。

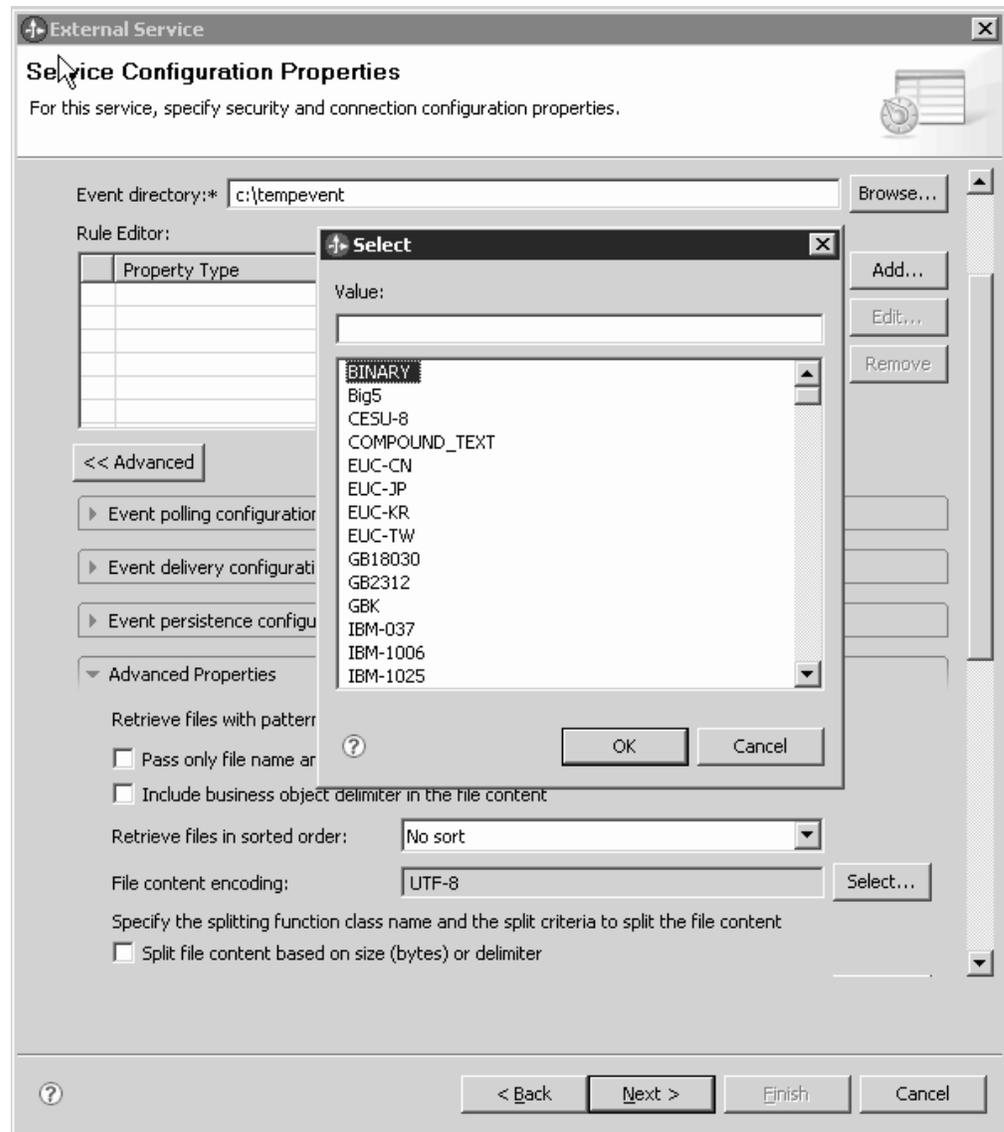


図 61. 「ファイル内容のエンコード」フィールドでの *BINARY* の選択

- c. オプション: アダプターの複数インスタンスがある場合は、「**ロギングおよびトレース**」を展開し、「**アダプター ID**」の値をこのインスタンスの固有な値に設定します。このプロパティについて詳しくは、183 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。
- d. 一定の情報を隠して、その情報がログまたはトレースに表示されないようにするには、「**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。
- e. オプション: ログ・ファイルの出力先を指定したり、このモジュールのロギング・レベルを定義したりするには、「**ウィザードのロギング・プロパティを変更します (Change logging properties for wizard)**」チェック・ボックスを選択します。ロギング・レベルの設定については、156 ページの『ロギング・プロパティの構成』を参照してください。

6. 「関数セレクトター」フィールドでは、既存の関数セレクトター構成を使用するか、新しい構成を作成するかを選択します。関数セレクトターによって、サービスの正しい操作に対して着信メッセージまたは要求が割り当てられます。
 - 既存の関数セレクトター構成を使用するには、「参照」をクリックして関数セレクトターのリストを表示します。使用可能な関数セレクトターの説明については、外部サービス・ウィザードの、接続プロパティのトピックを参照してください。
 - 既存の関数セレクトター構成を使用するには、「参照」をクリックして関数セレクトターのリストを表示します。使用可能な関数セレクトターの説明については、外部サービス・ウィザードの、接続プロパティのトピックを参照してください。

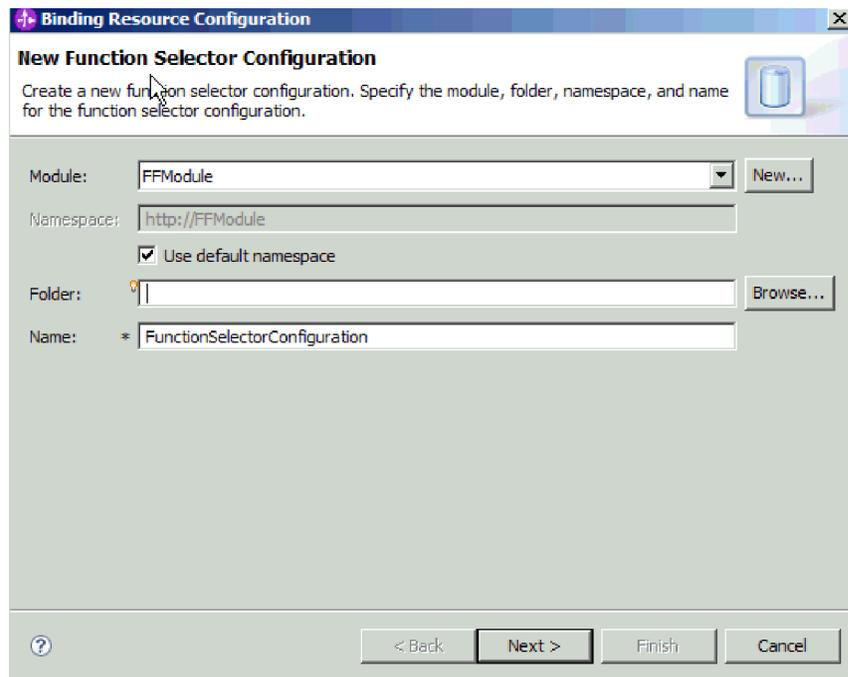


図 62. 新規関数セレクトター構成の作成

注: エンタープライズ情報システム (EIS) 関数名は、外部サービス・ウィザードでは使用できません。アダプター (基底クラス) で生成されるデフォルト以外の値を指定する場合、アセンブリー・エディターを使用してその値を編集することができます。

7. 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。

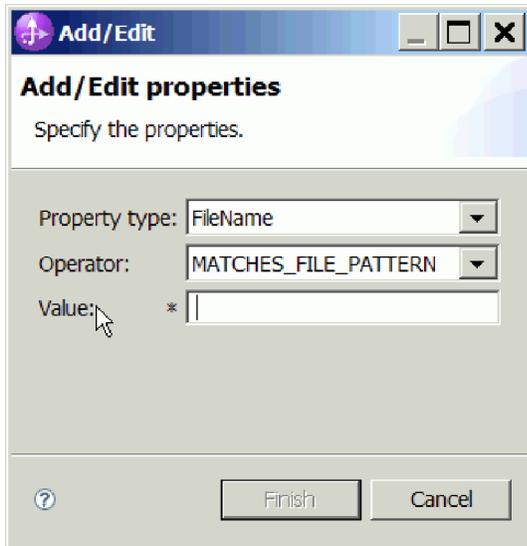


図 63. ルールの追加または編集

- 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
 - Directory
 - 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_REGEX(regex) 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。

Directory には、演算子として MATCHES_REGEX が組み込まれます。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_REGEX 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ります。

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、215 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。

8. 「終了」をクリックします。

結果

アダプターにより活動化仕様プロパティが保存されます。

次のタスク

モジュールのデータ・タイプと、選択したデータ・タイプに関連付ける操作の名前を指定します。

関連資料

173 ページの『ウィザードの接続プロパティ』

接続プロパティはサービス記述を作成して、組み込みの成果物を保存するために使用されます。これらのプロパティは、外部サービス・ウィザードで構成されます。

198 ページの『活動化仕様プロパティ』

活動化仕様プロパティは、エクスポート用の Inbound イベント処理の構成情報を保持しています。活動化仕様プロパティは、外部サービス・ウィザードまたは管理コンソールのいずれかを使用して設定します。

操作およびデータ・タイプの選択

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。外部サービス・ウィザードでは、3 種類 (汎用 FlatFile ビジネス・オブジェクト、ビジネス・グラフ付きの汎用 FlatFile ビジネス・オブジェクト、およびユーザー定義タイプ) の中からデータ・タイプを選択できます。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、ローカル・ファイル・システム 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。

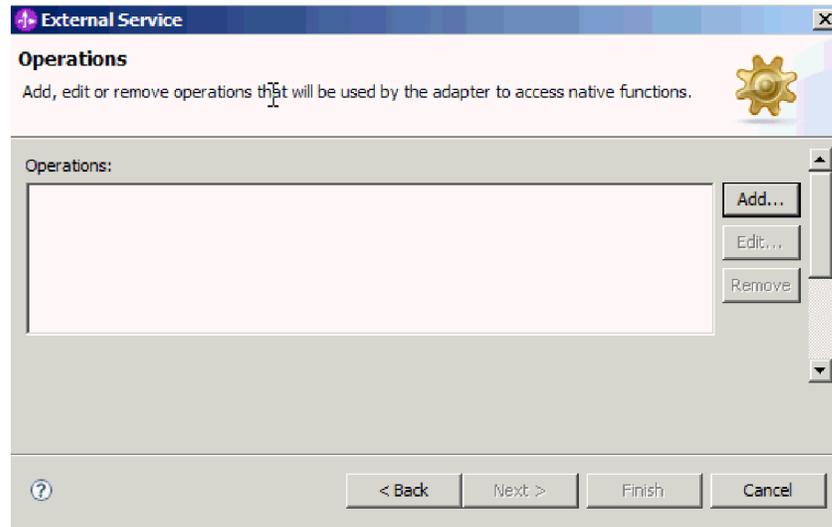


図 64. 操作の追加

2. 「操作の追加 (Add Operations)」ウィンドウで、データ・タイプを選択します。
3 つのデータ・タイプ (汎用 FlatFile ビジネス・オブジェクト、ビジネス・グラフ付きの汎用 FlatFile ビジネス・オブジェクト、およびユーザー定義タイプ) から選択できます。各データ・タイプと、どのタイプのビジネス・オブジェクトの生成に各データ・タイプが使用されるかについて詳しくは、本書のビジネス・オブジェクトの構造に関するセクションを参照してください。この例では、「汎用 FlatFile ビジネス・オブジェクト」を選択します。
3. 「次へ」をクリックします。「操作」ウィンドウに、「emitFlatFile」という操作名が表示されます。Inbound 処理で使用できる操作は、出力操作のみです。

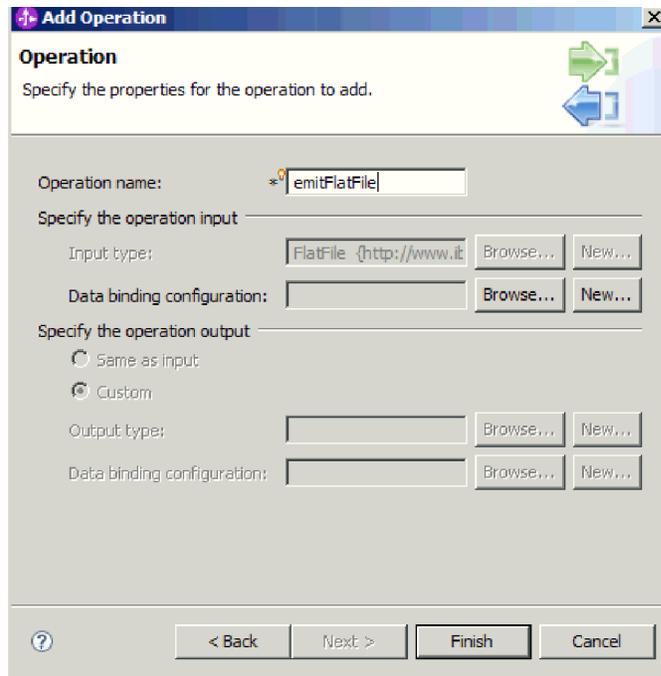


図 65. 操作の追加

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。

次のタスク

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。

データ・バインディングの構成

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、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。外部サービス・ウィザードで、モジュールにデータ・バインディングを追加し、追加したデータ・バインディングを、使用するデータ・タイプに合うように構成します。このようにして、アダプターはファイル内のフィールドに、ビジネス・オブジェクト内で受け取った情報を取り込む方法を識別します。

始める前に

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

このタスクを実行する理由および時期

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

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

Integration Developer で「新規」 → 「リソース構成 (Resource configuration)」を選択し、本書で説明されているデータ・バインディング画面を完了してください。

このタスクの手順

1. 「操作」ウィンドウの、「Input 操作のデータ・バインディング構成 (operation input Data binding configuration)」フィールドで「新規」を選択します。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。
2. オプション: 「新規データ・バインディング構成」画面には、「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・バインディング構成の「名前」(この例では DataBindingConfiguration を使用) を入力します。「次へ」をクリックします。

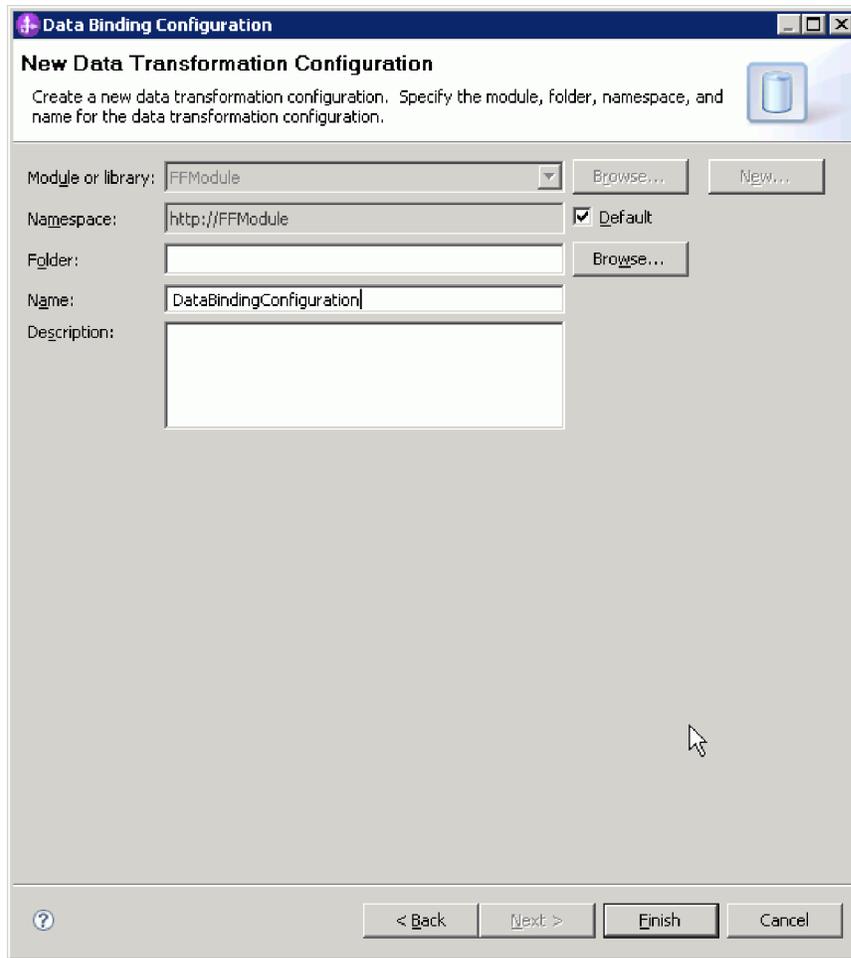


図 66. データ・バインディング構成の名前指定

5. 「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

データ・ハンドラー構成を選択します。

データ・ハンドラーの構成

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形式の間の変換を実行します。

始める前に

モジュールのデータ・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。また、WebSphere Integration Developer Business Object Editor を使用して、ビジネス・オブジェクトを事前に定義しておく必要があります。ここでウィザードを停止してビジネス・オブジェクトを作成する場合は、ウィザードのステップを最初から開始する必要があります。

注: データ・ハンドラーは、WebSphere Integration Developer を使用して外部サービス・ウィザードを実行する前に構成できます。構成を行うには、WebSphere Integration Developer で「新規」→「リソース構成 (Resource configuration)」を選択し、本書で説明されているデータ・ハンドラー画面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

ビジネス・オブジェクトのデータ・ハンドラーを指定するには、以下の手順を実行します。

このタスクの手順

1. 「データ・バインディング・プロパティ」ウィンドウで、「新規」をクリックし、データ・ハンドラー構成の名前 (この例では DHConfig を使用) を入力します。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。

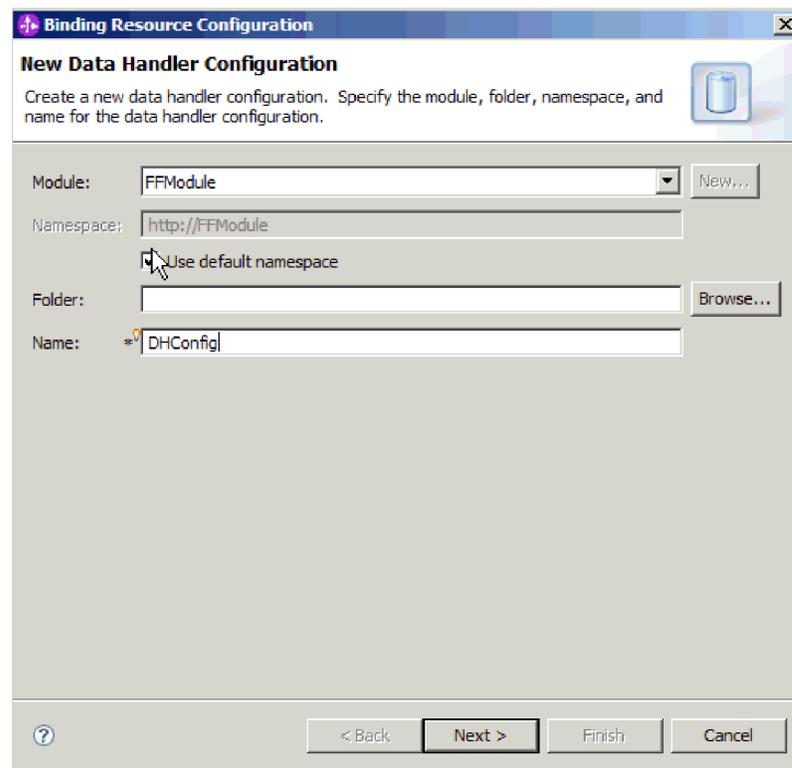


図 67. データ・ハンドラー構成の名前の指定

2. 「次へ」をクリックします。
3. データ・ハンドラーのクラス名を選択します。「構成タイプの選択 (Select a Configuration Type)」ウィンドウで、「参照」をクリックして、データ・ハンドラーのクラス名を選択します。「データ・ハンドラー・クラスの表示 (Show data handler classes)」ラジオ・ボタンを選択します。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 XMLDataHandler を使用)。「OK」をクリックします。

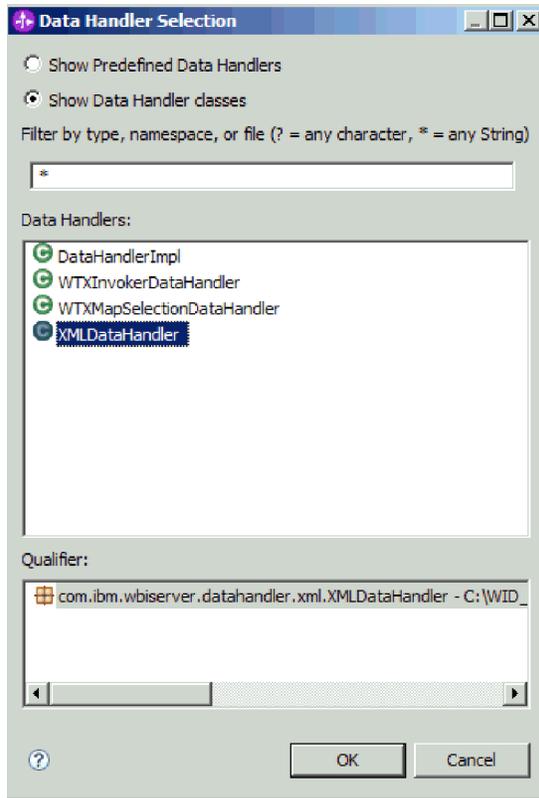


図 68. データ・ハンドラー・クラスを選択

4. 「次へ」をクリックします。
5. 「プロパティの指定 (Specify properties)」ウィンドウで、エンコード方式を指定します (この例では UTF-8 を使用)。

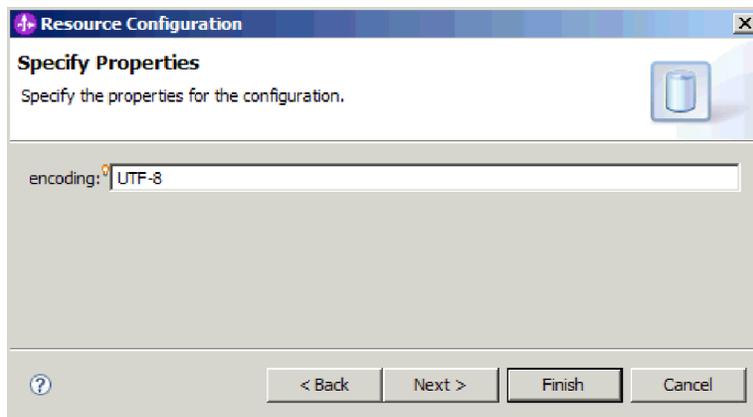


図 69. データ・ハンドラー構成のエンコード方式の指定

6. 「終了」をクリックします。次の画面に、追加された Inbound 操作が対話仕様プロパティとともに表示されます。

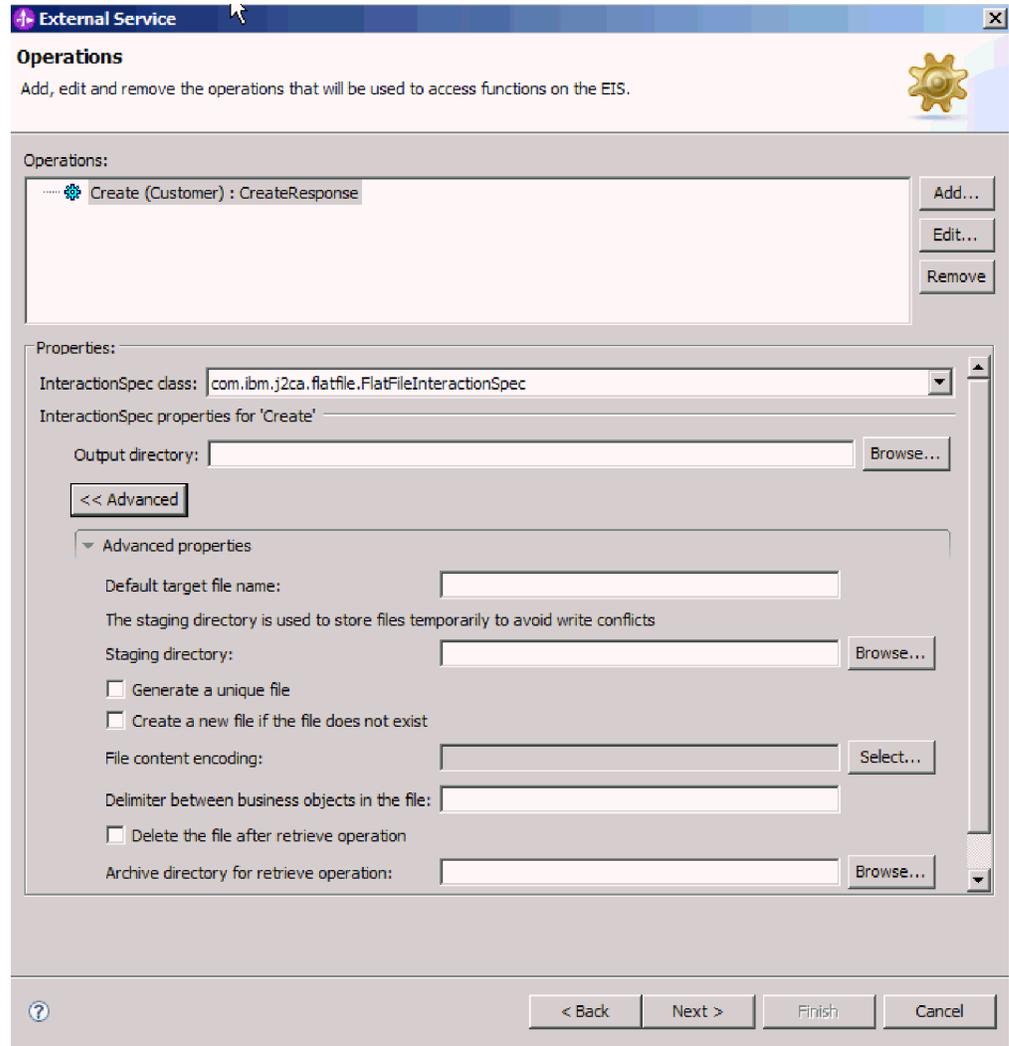


図 70. Inbound 操作と InteractionSpec プロパティ

7. 「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

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。

デプロイメント・プロパティの設定およびサービスの生成

外部サービス・ウィザードを使用して、活動化仕様プロパティを設定し、モジュールで使用する成果物を生成します。成果物は、外部サービスの一部として作成される、ビジネス・オブジェクト、WSDL ファイル、およびインポート・ファイルとエクスポート・ファイルです。アダプターは、モジュールの成果物を生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、最上位レベルのビジネス・オブジェクトの操作が含まれます。

始める前に

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

このタスクを実行する理由および時期

活動化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。活動化仕様プロパティについて詳しくは、本書の該当する参照トピックを参照してください。

このタスクの手順

1. 活動化仕様プロパティを設定して成果物を生成するには、以下のステップを完了します。
 - a. 「サービス構成プロパティ」ウィンドウで、「**拡張**」をクリックします。
 - b. デフォルトとして設定するすべてのフィールドに値を入力します。
 - c. 「**次へ**」をクリックします。
2. 「操作」ウィンドウで、「**次へ**」をクリックします。「サービスの生成 (Generate Service)」画面で、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せん。

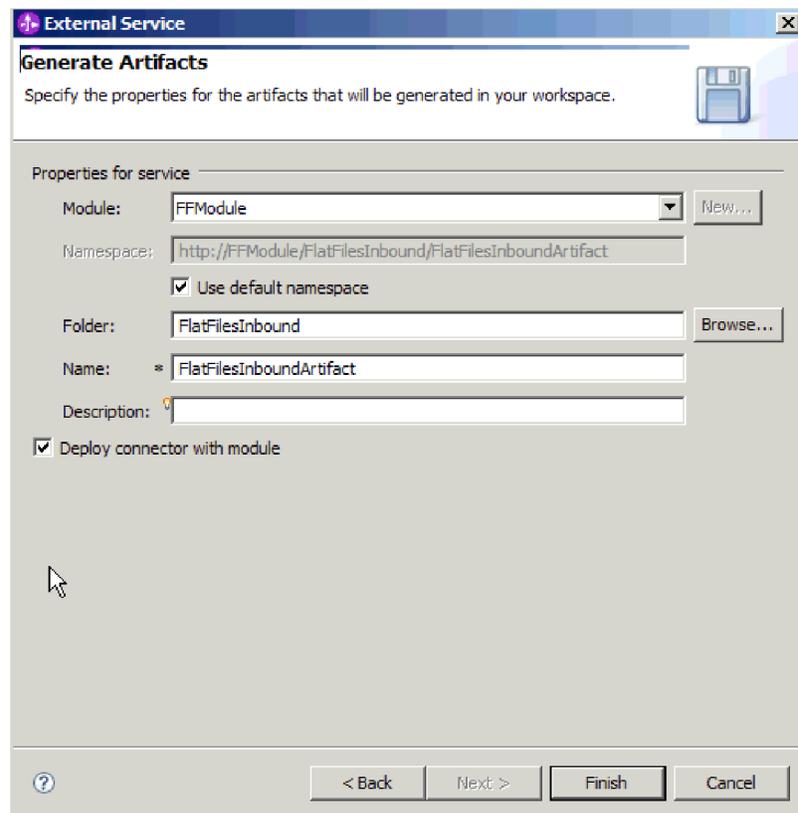


図 71. 成果物への命名

3. 「**終了**」をクリックします。

結果

WebSphere Integration Developer が成果物およびイン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

モジュールをデプロイします。

関連資料

171 ページの『Outbound 構成プロパティ』

WebSphere Adapter for Flat Files 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 Outbound 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用してモジュールを WebSphere Process Server または WebSphere Enterprise Service Bus にデプロイした後に、リソース・アダプターおよび Managed Connection Factory のプロパティを変更することができますが、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。

192 ページの『Inbound 構成プロパティ』

WebSphere Adapter for Flat Files 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用してモジュールをデプロイした後に、リソース・アダプターおよび活動化仕様のプロパティを変更することができますが、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。

218 ページの『グローバリゼーション』

WebSphere Adapter for Flat Files 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。

第 5 章 アセンブリー・エディターの使用による対話仕様プロパティの変更

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

始める前に

外部サービス・ウィザードを使用して、アダプターのサービスが生成済みである必要があります。

このタスクを実行する理由および時期

アダプターのサービスを生成した後、対話仕様プロパティを変更したい場合があります。対話仕様プロパティ (オプション) は、特定のビジネス・オブジェクトの特定の操作に対し、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。EAR ファイルをエクスポートする前に、これらのプロパティを変更することができます。アプリケーションのデプロイ後に、これらのプロパティを変更することはできません。

対話仕様プロパティを変更するには、次の手順を使用してください。

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 (Assembly Diagram)」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・エディターのインターフェースをクリックします。(それ以上クリックしなければ、モジュール・プロパティが表示されます。)
4. 「プロパティ」タブをクリックします。(図のインターフェースを右クリックして、「プロパティに表示 (Show in Properties)」をクリックすることもできます。)
5. 「バインディング」の下で、「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 つ表示されます。
6. 対話仕様プロパティを変更するメソッドを選択します。
7. 「拡張」をクリックし、「汎用 (Generic)」タブでプロパティを変更します。対話仕様プロパティを変更する各メソッドについて、この手順を繰り返します。

結果

アダプター・モジュールに関連付けられた対話仕様プロパティが変更されます。

次のタスク

モジュールをデプロイします。

関連資料

186 ページの『対話仕様プロパティ』
対話仕様プロパティには、アダプターがファイル・システムとのインターフェースとなるときに使用する Outbound 接続プロパティが含まれています。外部サービス・ウィザード を使用して、これらのプロパティを構成します。アプリケーションがデプロイされた後に対話仕様プロパティを変更するには、WebSphere Integration Developer でアSEMBリー・エディターを使用します。

第 6 章 モジュールのデプロイ

モジュールをデプロイし、モジュールとアダプターを構成しているファイルを実動またはテスト用の稼働環境に置きます。WebSphere Integration Developer では、統合テスト環境は、インストール中に選択したテスト環境プロファイルによって、WebSphere Process Server または WebSphere Enterprise Service Bus、あるいはこれら両方のランタイム・サポートが可能です。

デプロイメント環境

モジュールおよびアダプターをデプロイできるテスト環境および実稼働環境があります。

WebSphere Integration Developer では、テスト環境の 1 つ以上のサーバーにモジュールをデプロイできます。これは、通常ビジネス・インテグレーション・モジュールを実行およびテストするための最も一般的な方法です。しかし、管理コンソール・ツールまたはコマンド行ツールを使用して、サーバー・デプロイメント用のモジュールを EAR ファイルとして WebSphere Process Server または WebSphere Enterprise Service Bus にエクスポートすることもできます。

テストする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含む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、サーバーの始動と停止、およびモジュール・コードでのエラーのテストなどのタスクを実行できるサーバー・ツールで作業を行うことができます。テストは、一般にコンポーネントのインターフェース操作で実行され、これにより、コンポーネントが正しく実装されているかどうか、および参照が正しく関連付けられているかどうかを判別できます。

このタスクを実行する理由および時期

Inbound 処理のテスト用ターゲット・コンポーネントの生成および配線

テスト環境に Inbound 処理用のアダプターを含むモジュールをデプロイする前に、まずターゲット・コンポーネントを生成し、配線する必要があります。このターゲット・コンポーネントは、アダプターによるイベントの送信の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・モジュールが生成済みでなければなりません。

このタスクを実行する理由および時期

Inbound 処理のテスト用ターゲット・コンポーネントの生成および配線は、テスト環境のみにおいて必要です。実稼働環境においてアダプターをデプロイ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け取ります。WebSphere Integration Developer のアセンブリー・エディターを使用して、ターゲット・コンポーネントへのエクスポート (2 つのコンポーネントを接続) を配線します。アダプターは、配線を使用して、(ターゲット・コンポーネントへのエクスポートから) 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・コンポーネントの作成

- a. WebSphere Integration Developer の Business Integration パースペクティブから、「アセンブリー・ダイアグラム (Assembly Diagram)」を展開し、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の名前 + **InboundInterface** です。

インターフェースは、呼び出すことのできる操作および受け渡されるデータ (入力引数、戻り値、例外など) を指定します。**InboundInterface** は、Inbound 処理をサポートするのにアダプターが必要とする操作を含み、外部サービス・ウィザードを実行する際に作成されます。

- b. 「コンポーネント」を展開し、「型なしコンポーネント」を選択して、コンポーネントをアセンブリー・ダイアグラムにドラッグすることで、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. アセンブリー・ダイアグラムに表示させるにはコンポーネントをクリックします。

2. コンポーネントを配線します。

- a. エクスポート・コンポーネントをクリックして新規コンポーネントにドラッグします。これにより、以下の図に示されるように、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が引かれます。
- b. アセンブリー・ダイアグラムを保管します。「ファイル」→「保管」をクリックします。

3. 生成、新規コンポーネントの実装の

- a. 新規コンポーネントを右クリックして、「実装の生成 (Generate implementation)」を選択します。

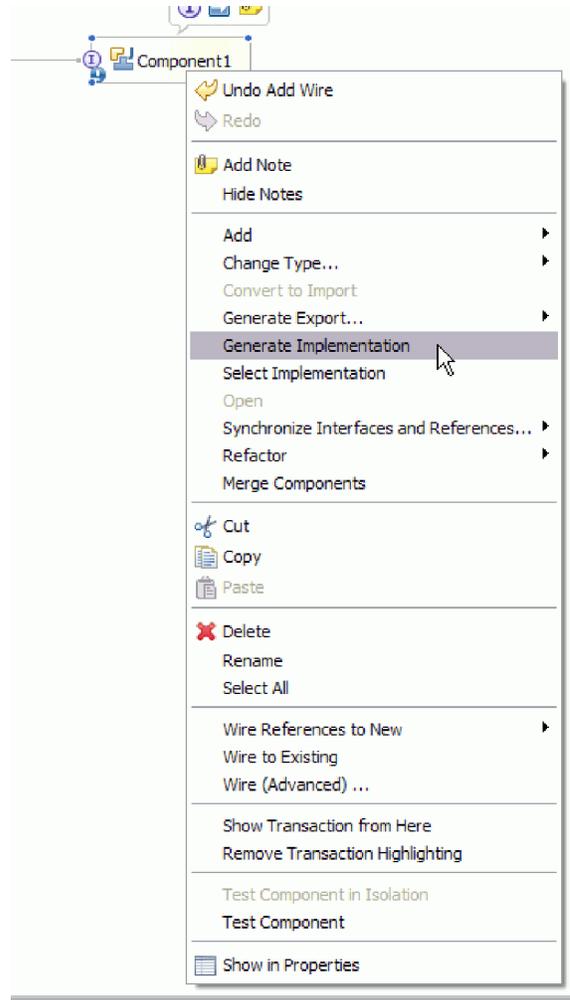


図 72. Java 実装の生成

- 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。

Java 実装は、別のタブに表示されます。

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

次のタスク

テストする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追加できます。

始める前に

テスト中のモジュールが、アダプターを使用して Inbound 処理を実行する場合、アダプターによるイベントの送信先であるターゲット・コンポーネント を生成して配線する必要があります。

このタスクを実行する理由および時期

モジュールと、モジュールによるアダプターの使用をテストするために、サーバーへモジュールを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、以下のステップを実行して、新規サーバーを追加および定義します。
 - a. 「サーバー」ビューにカーソルを置き、右クリックして「新規」 → 「サーバー」と選択します。
 - b. 「新規サーバーの定義 (Define a New Server)」ウィンドウから、サーバー・タイプを選択します。
 - c. サーバーの設定を構成します。
 - d. 「終了」をクリックすると、サーバーが公開されます。
2. モジュールをサーバーに追加します。
 - a. 「サーバー」ビューに切り替えます。 WebSphere Integration Developer で、「ウィンドウ (Windows)」 → 「ビューの表示 (Show View)」 → 「サーバー」と選択します。
 - a. サーバーを始動します。 WebSphere Integration Developer の画面の右下のペインの「サーバー」タブで、「サーバー」を右クリックして、「始動 (Start)」を選択します。
3. サーバー状況が「始動済み」である場合は、「サーバー」を右クリックし、「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」画面でプロジェクトを選択し、「追加」をクリックします。 プロジェクトは、「使用可能プロジェクト」リストから「構成プロジェクト」リストに移動します。
5. 「終了」をクリックします。 これにより、モジュールがサーバーにデプロイされます。

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

次のタスク

モジュールとアダプターの機能をテストしてください。

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

WebSphere Integration Developer 統合テスト・クライアントを使用して、アSEMBルされた、Outbound 処理用モジュールおよびアダプターをテストします。

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、一般にコンポーネントのインターフェース操作で実行され、これにより、コンポーネントが正しく実装されているかどうか、および参照が正しく関連付けられているかどうかを判別できます。

このタスクの手順

1. テストするモジュールを選択して右クリックし、「テスト」 → 「モジュールのテスト (Test Module)」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 (Testing modules and components)」トピックを参照してください。

次のタスク

モジュールとアダプターのテスト結果が満足できるものであれば、モジュールとアダプターを実稼働環境にデプロイできます。

実動用のモジュールのデプロイ

外部サービス・ウィザードで作成したモジュールの、実稼働環境の WebSphere Process Server または WebSphere Enterprise Service Bus へのデプロイは、2 つのステップからなるプロセスです。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・インスタンスのデプロイされたすべてのアプリケーションで使用可能にすることを選択する場合は、RAR ファイルの形式でアダプターをアプリケーション・サーバーにインストールする必要があります。RAR ファイルは、Java 2 Connector (J2C) アーキテクチャーのリソース・アダプターをパッケージ化するのに使用される Java アーカイブ (JAR) ファイルです。

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

RAR ファイルの形式でアダプターをインストールすると、アダプターは、サーバー・ランタイムに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」とクリックします。
3. 「リソース・アダプター」 ページで、「RAR のインストール」をクリックします。

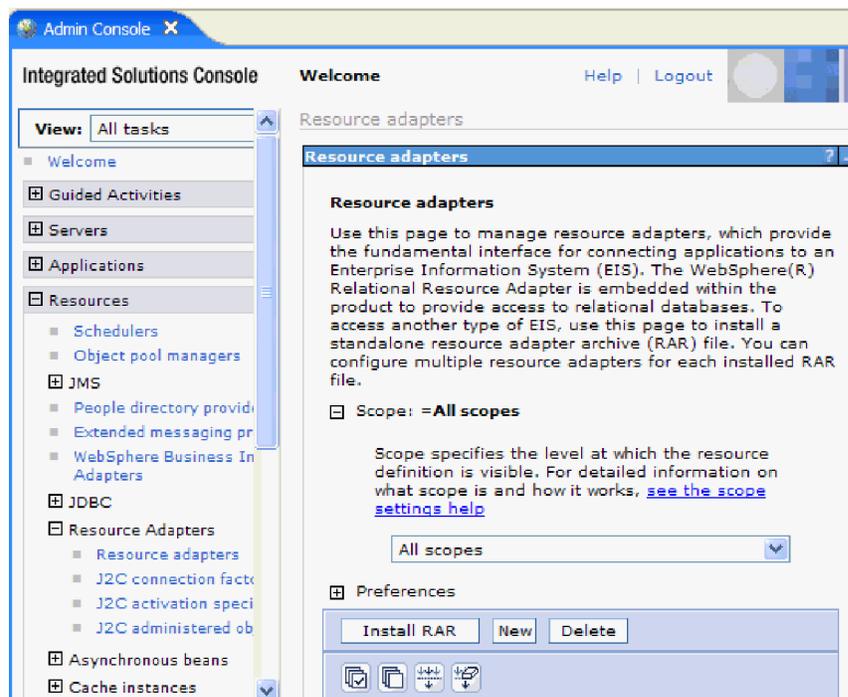


図 73. 「リソース・アダプター」 ページの「RAR のインストール」 ボタン

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

RAR ファイルは、通常、パス `WID_installation_directory/ResourceAdapters/adapter_name/deploy/adapter.rar` 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 (Resource adapters)」 ページで、オプションで、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」ボックスの「保管」をクリックします。

次のタスク

次のステップでは、サーバーにデプロイ可能なモジュールを 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てください。

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 アプリケーションは、モジュールにちなんで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. オプションで、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 (Export source files)**」を選択します。このオプションは、EAR ファイルに加えてソース・ファイルをエクスポートする場合に提供されます。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たファイルが含まれています。
7. 既存のファイルを上書きする場合は、「**既存のファイルを上書き (Overwrite an existing file)**」をクリックします。
8. 「**終了**」をクリックします。

結果

モジュールの内容が EAR ファイルとしてエクスポートされます。

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・プロセスの最後のステップです。EAR ファイルをサーバーにインストールして実行すると、EAR ファイルの一部として組み込まれているアダプターが、インストール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

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

このタスクを実行する理由および時期

EAR ファイルをインストールするには、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 (Applications)」 → 「新規アプリケーションのインストール (Install New Applications)」をクリックします。

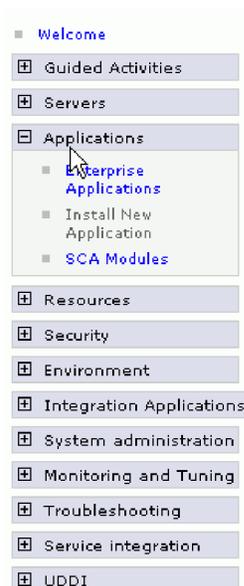


図 74. 「アプリケーション・インストールの準備」ウィンドウ

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、ランタイム環境の管理コンソールを使用します。リソース・アダプター・プロパティー (一般的なアダプター操作に使用される)、Managed Connection Factory プロパティー (Outbound 処理に使用される)、および活動化仕様プロパティー (Inbound 処理に使用される) を更新することができます。

関連資料

192 ページの『Inbound 構成プロパティー』

WebSphere Adapter for Flat Files 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ーがあります。WebSphere Integration Developer または管理コンソールを使用してモジュールをデプロイした後に、リソース・アダプターおよび活動化仕様のプロパティーを変更することができますが、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。

171 ページの『Outbound 構成プロパティー』

WebSphere Adapter for Flat Files 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 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用してモジュールを WebSphere Process Server または WebSphere Enterprise Service Bus にデプロイした後に、リソース・アダプターおよび Managed Connection Factory のプロパティーを変更することができますが、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。

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

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

このタスクを実行する理由および時期

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

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

このタスクの手順

1. 管理コンソールを開始します。
2. 「アプリケーション」の下で、「エンタープライズ・アプリケーション」を選択します。
3. 「エンタープライズ・アプリケーション」 リストから、プロパティ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。

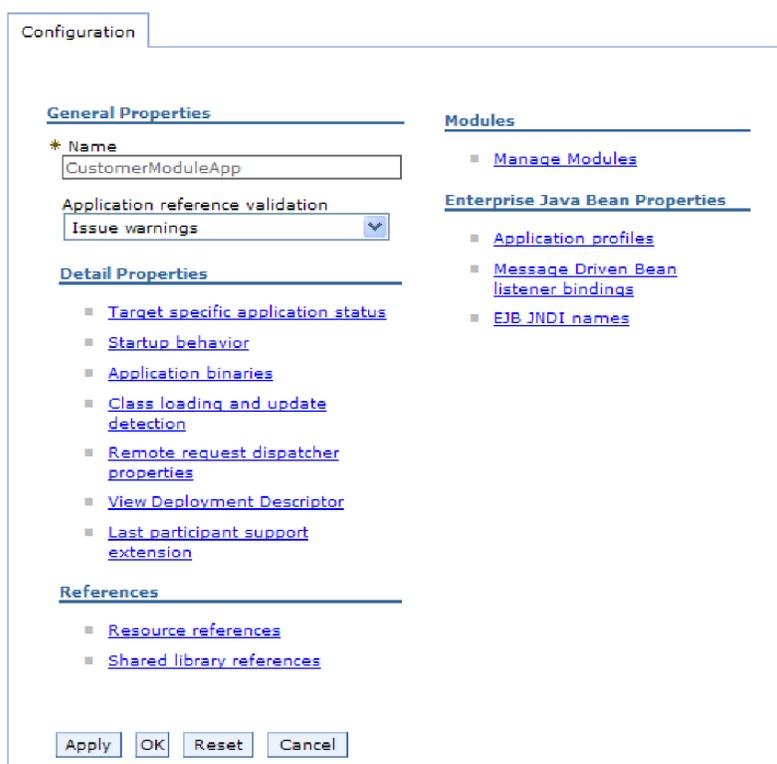


図 75. 「構成」タブでの「モジュールの管理」の選択

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 Flat Files」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、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、183 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、下図に示されたページが表示されます。

The screenshot shows a configuration window titled "Configuration". Under the "General Properties" section, there are several input fields: "Scope" with the value "widNode", "Name" with "logNumberOfFiles", and "Value" with "1". There is also a "Description" text area and a "Type" dropdown menu set to "java.lang.String". A "Required" checkbox is present and is not checked. At the bottom of the window are four buttons: "Apply", "OK", "Reset", and "Cancel".

図 76. logNumberOfFiles プロパティの「構成」タブ

- b. 「値」フィールド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、プロパティの説明を追加することができます。
 - c. 「OK」をクリックします。
9. ウィンドウの上部にある「メッセージ」ボックスの「保管」リンクをクリックします。

結果

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

関連資料

183 ページの『リソース・アダプター・プロパティ』

リソース・アダプター・プロパティでは、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・アダプターのプロパティは、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。アダプターをデプロイしたあとは、管理コンソールを使用してこれらのプロパティを変更します。

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

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

このタスクを実行する理由および時期

Managed Connection Factory プロパティは、ターゲット・ローカル・ファイル・システムのインスタンスを構成する場合に使用します。

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

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

このタスクの手順

1. 管理コンソールを開始します。
2. 「アプリケーション」の下で、「エンタープライズ・アプリケーション」を選択します。
3. 「エンタープライズ・アプリケーション」 リストで、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。

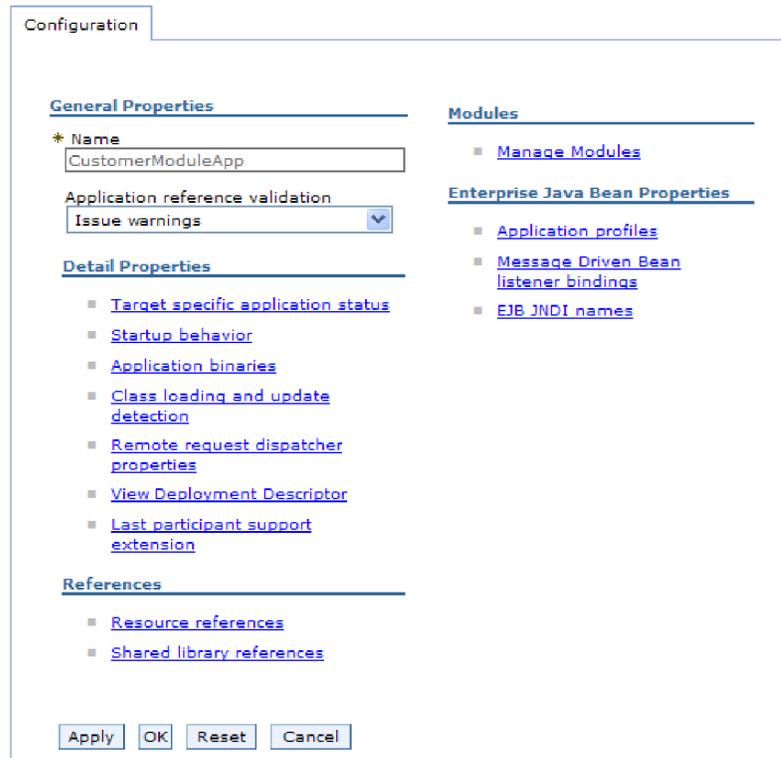


図 77. 「構成」タブでの「モジュールの管理」の選択

5. 「**IBM WebSphere Adapter for Flat Files**」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・モジュールと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 Flat Files 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、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、177 ページの『Managed 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。

関連資料

177 ページの『Managed Connection Factory プロパティ』

Managed Connection Factory プロパティでは、アダプターがローカル・ファイル・システムとの Outbound 通信の実行時に必要とする情報を指定します。

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、こ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

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

このタスクを実行する理由および時期

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、次の手順を使用してください。

このタスクの手順

1. 管理コンソールを開始します。
2. 「アプリケーション」の下で、「エンタープライズ・アプリケーション」を選択します。
3. 「エンタープライズ・アプリケーション」 リストから、プロパティ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。

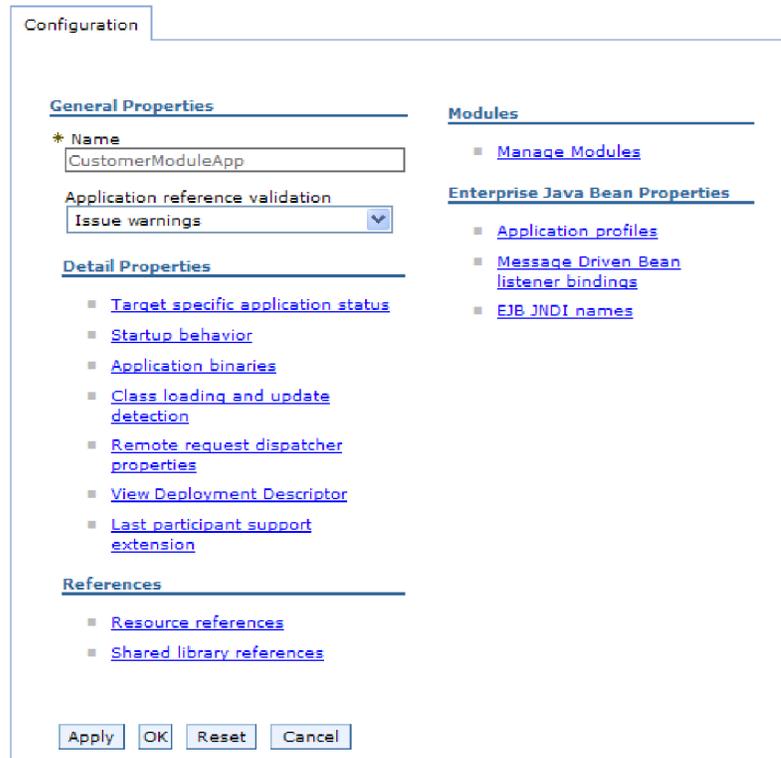


図 78. 「構成」タブでの「モジュールの管理」の選択

5. 「**IBM WebSphere Adapter for Flat Files**」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・モジュールと関連付けられた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 (J2C activation specification custom properties)**」をクリックします。
10. 変更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、198 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活動化仕様プロパティが変更されます。

関連資料

198 ページの『活動化仕様プロパティ』

活動化仕様プロパティは、エクスポート用の Inbound イベント処理の構成情報を保持しています。活動化仕様プロパティは、外部サービス・ウィザードまたは管理コンソールのいずれかを使用して設定します。

スタンドアロン・アダプターの構成プロパティの変更

スタンドアロン・アダプターをインストールした後に構成プロパティを設定するには、ランタイム環境の管理コンソールを使用します。アダプターに関する一般情報を提供してから、リソース・アダプター・プロパティ (一般的なアダプター操作に使用される) を設定します。アダプターが、Outbound 操作に使用される場合は、接続ファクトリーを作成し、その接続ファクトリーのプロパティを設定します。アダプターが、Inbound 操作に使用される場合は、活動化仕様を作成し、その活動化仕様のプロパティを設定します。

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、このアダプターのリソース・アダプター・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

ご使用のアダプターは、WebSphere Process Server または WebSphere Enterprise Service Bus にインストール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・プロパティは、すべての WebSphere アダプターが共用するデフォルトの構成プロパティです。

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

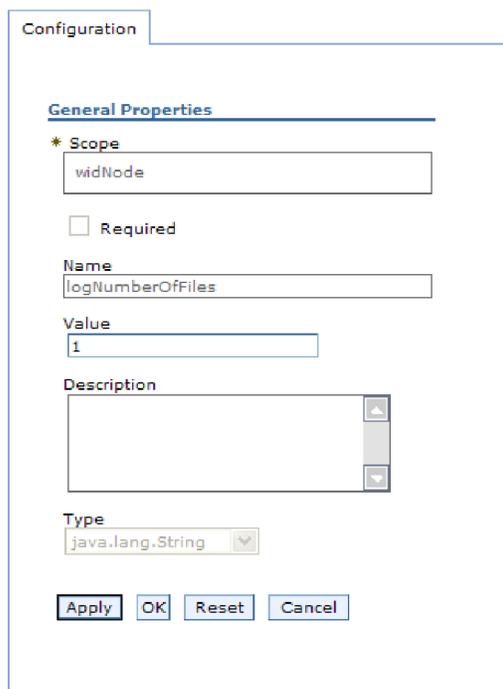
このタスクの手順

1. 管理コンソールを開始します。
2. 「リソース」 → 「リソース・アダプター」 → 「リソース・アダプター」とクリックします。
3. 「リソース・アダプター」ページで、「**IBM WebSphere Adapter for Flat Files**」をクリックします。
4. 「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。
5. 変更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は、183 ページの『リソース・アダプター・プロパティ』を参照してください。

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

例えば、「logNumberOfFiles」をクリックした場合、以下のページが表示されます。



The screenshot shows a configuration window with the following fields and controls:

- Configuration** (Tab)
- General Properties** (Section Header)
- * Scope**: widNode
- Required
- Name**: logNumberOfFiles
- Value**: 1
- Description**: (Empty text area)
- Type**: java.lang.String
- Buttons: Apply, OK, Reset, Cancel

図 79. logNumberOfFiles プロパティの「構成」タブ

「値」フィールドの数値を変更し、プロパティの説明を追加することができます。

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

結果

アダプターに関連付けられたリソース・アダプター・プロパティが変更されません。

関連資料

183 ページの『リソース・アダプター・プロパティ』

リソース・アダプター・プロパティでは、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・アダプターのプロパティは、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。アダプターをデプロイしたあとは、管理コンソールを使用してこれらのプロパティを変更します。

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、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

ご使用のアダプターは、WebSphere Process Server または WebSphere Enterprise Service Bus にインストール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・ローカル・ファイル・システムのインスタンスを構成する場合に使用します。

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

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

このタスクの手順

1. 管理コンソールを開始します。
2. 「リソース」 → 「リソース・アダプター」 → 「リソース・アダプター」とクリックします。
3. 「リソース・アダプター」 ページで、「**IBM WebSphere Adapter for Flat Files**」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、既存接続ファクトリーのリストからの選択にスキップします。

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 (General Properties)」セクションで、接続ファクトリーの名前を入力します。例えば、AdapterCF と入力します。
- c. 「JNDI 名」に値を入力します。例えば、com/eis/AdapterCF と入力します。
- d. 「コンポーネント管理認証別名」リストから認証別名を選択します。
- e. 「OK」をクリックします。
- f. ページの上部にある「メッセージ」ボックスの「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 80. リソース・アダプターと併用するための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、使用したい接続ファクトリーをクリックします。
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 Flat Files 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、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、177 ページの『Managed 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完了したら、「適用」をクリックします。
 10. ウィンドウの上部にある「メッセージ」ボックスの「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 Managed Connection Factory プロパティが設定されます。

関連資料

177 ページの『Managed Connection Factory プロパティ』

Managed Connection Factory プロパティでは、アダプターがローカル・ファイル・システムとの Outbound 通信の実行時に必要とする情報を指定します。

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、こ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

ご使用のアダプターは、WebSphere Process Server または WebSphere Enterprise Service Bus にインストール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、次の手順を使用してください。

このタスクの手順

1. 管理コンソールを開始します。
2. 「リソース」 → 「リソース・アダプター」 → 「リソース・アダプター」とクリックします。
3. 「リソース・アダプター」ページで、「**IBM WebSphere Adapter for Flat Files**」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、活動化仕様の既存リストからの選択にスキップします。

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 (General Properties)**」セクションで、活動化仕様の名前を入力します。例えば、AdapterAS と入力します。
- c. 「**JNDI 名**」に値を入力します。例えば、com/eis/AdapterAS と入力します。
- d. 「**認証別名**」リストから認証別名を選択します。
- e. メッセージ・リスナー・タイプを選択します。
- f. 「**OK**」をクリックします。
- g. ページの上部にある「**メッセージ**」ボックスの「**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、使用したいスペックをクリックします。
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 (J2C activation specification custom properties)**」をクリックします。
8. 設定するプロパティごとに、以下の手順を実行します。

注: ここで示すプロパティについて詳しくは、198 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完了したら、「適用」をクリックします。
 10. ページの上部にある「メッセージ」ボックスの「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。

関連資料

198 ページの『活動化仕様プロパティ』

活動化仕様プロパティは、エクスポート用の Inbound イベント処理の構成情報を保持しています。活動化仕様プロパティは、外部サービス・ウィザードまたは管理コンソールのいずれかを使用して設定します。

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、アプリケーションは、サーバーの始動時に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

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

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」というラベルが付けられます。

2. 開始したい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 ファイル名から .EAR ファイル 拡張子を除いたものです。
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 (Started)」に変化し、アプリケーションが開始したことを示すメッセージが管理コンソールの上部に表示されます。

アダプターを使用するアプリケーションの停止

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

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

このタスクの手順

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

注: 管理コンソールには、「Integrated Solutions Console」というラベルが付けられます。

2. 停止したい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 ファイル名から .EAR ファイル 拡張子を除いたものです。
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「停止中 (Stopped)」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 Flat Files 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組み込ま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。

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、向上させるために、PMI がどう役立つかについての詳細な説明は、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。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、152 ページの『Common Event Infrastructure (CEI) によるトレースの使用可能化』を参照してください。

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

このタスクの手順

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

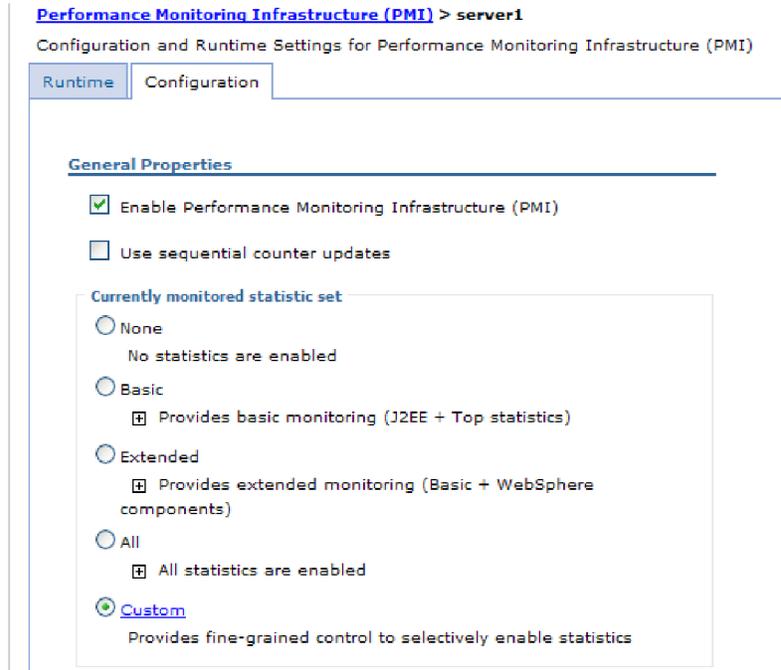


図 81. Performance Monitoring Infrastructure の使用可能化

- 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。

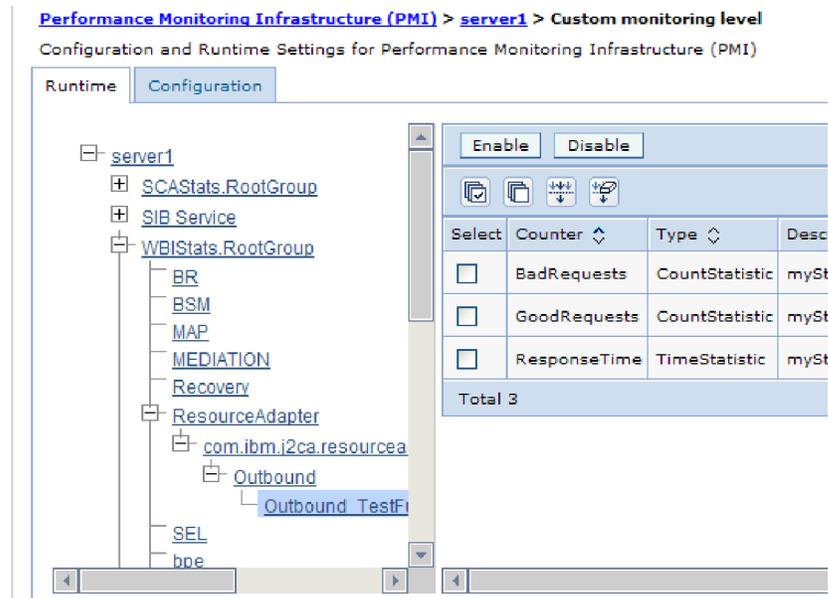


図 82. PMI の構成に使用される「ランタイム」タブ

- 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。

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. サーバーのリストで、ご使用のサーバー名をクリックします。
3. サーバー名の下で、「パフォーマンス・モジュール」を展開します。
4. 「WBISStatsRootGroup」をクリックします。
5. 「ResourceAdapter」およびアダプター・モジュールの名前をクリックします。
6. 複数のプロセスがある場合は、統計を表示させるプロセスのチェック・ボックスを選択します。

結果

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

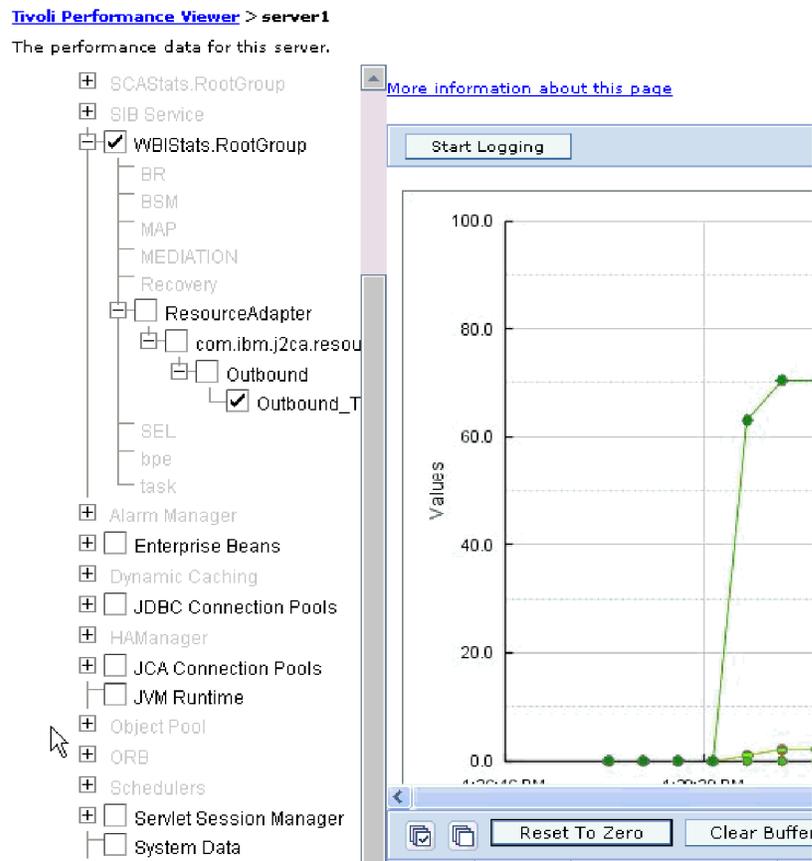


図 83. グラフ表示によるアダプターのパフォーマンス統計

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

アダプターは、サーバー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と停止などの重要なビジネス・イベントに関するデータについて報告を行います。イベント・データは、構成設定に応じて、データベースまたはトレース・ログ・ファイルに書き込むことができます。

このタスクを実行する理由および時期

このタスクの手順

1. 管理コンソールで「トラブルシューティング」をクリックします。
2. 「ログおよびトレース (Logs and Trace)」をクリックします。
3. サーバーのリストで、ご使用のサーバー名をクリックします。
4. 「ログ詳細レベルの変更」 ボックスで、アダプターがイベント・データを書き込む CEI データベース (例えば、WBIEventMonitor.CEI.ResourceAdapter.*)、またはトレース・ログ・ファイル (例えば、WBIEventMonitor.LOG.ResourceAdapter.*) の名前をクリックします。
5. アダプターが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、(オプションで) メッセージとトレースに関連付けられた詳細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 (Messages Only)」。アダプターは、イベントを報告します。
 - 「すべてのメッセージとトレース (All Messages and Traces)」。アダプターは、イベントについての詳細を報告します。
 - 「メッセージ・レベルとトレース・レベル (Message and Trace Levels)」。イベントに関連付けられたビジネス・オブジェクト・ペイロードについてアダプターが報告する詳細度の制御の設定。詳細レベルを調整する場合は、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報告しますが、ビジネス・オブジェクト・ペイロードは報告しません。
 - 「詳細 - 高」。アダプターはイベントとビジネス・オブジェクト・ペイロードの説明を報告します。
 - 「詳細 - 必須」。アダプターは、イベントおよび全ビジネス・オブジェクト・ペイロードを通知します。
6. 「OK」をクリックします。

結果

イベント・ロギングは使用可能です。管理コンソール内の Common Base Event ブラウザーを使用して、トレース・ログ・ファイルの CEI 項目を表示させることができます。

第 8 章 トラブルシューティングおよびサポート

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

関連資料

223 ページの『アダプター・メッセージ』

以下のロケーションで WebSphere Adapter for Flat Files により発行されたメッセージを表示します。

Log and Trace Analyzer のサポート

アダプターでは、Log and Trace Analyzer で表示可能なログおよびトレース・ファイルが作成されます。

Log and Trace Analyzer を使用すると、ログおよびトレース・ファイルをフィルターに掛けて、アダプターのメッセージおよびトレース情報を分離できます。また、ログ・ビューアー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。

フィルターおよび強調表示に使用されるアダプターのコンポーネント ID は、文字 FFRA とアダプター ID プロパティの値で構成されたストリングです。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、FFRA001 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 9 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 Flat Files 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、FFRA001 および FFRA002 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 FFRAInstance0 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。184 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。

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、他のログ・ファイルおよびトレース・ファイルから区別します。

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

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

このタスクを実行する理由および時期

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

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

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

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

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

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

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

このタスクの手順

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

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

ロギング・レベル	説明
詳細	サブタスクが実行中です。このロギング・レベルには、サブタスクの進行状況を詳しく示す一般情報が含まれます。

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

結果

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

ログ・ファイル名およびトレース・ファイル名の変更

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

始める前に

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

このタスクを実行する理由および時期

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

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

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

このタスクの手順

1. 管理コンソールのナビゲーション・ペインで、「アプリケーション」>「エンタープライズ・アプリケーション」を選択します。
2. 「エンタープライズ・アプリケーション」リストから、アダプター・アプリケーションの名前をクリックします。これは、アダプターの EAR ファイルの名前から .ear ファイル拡張子を除いたものです。例えば、EAR ファイルの名前が Accounting_OutboundApp.ear である場合は、Accounting_OutboundApp をクリックします。
3. 「構成」タブの「モジュール」リストから、「モジュールの管理」をクリックします。

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

First Failure Data Capture (FFDC) のサポート

アダプターは、First Failure Data Capture (FFDC) をサポートします。これにより、WebSphere Process Server または WebSphere Enterprise Service Bus のランタイム中に発生する障害および重大なソフトウェアの問題が永続的に記録されます。

FFDC 機能は、バックグラウンドで実行され、ランタイムに発生するイベントおよびエラーを収集します。この機能により、互いに失敗を関連付ける手段が提供され、ソフトウェアは障害の結果をその原因に結び付けることができ、障害の根本原因を迅速に突き止めることが容易になります。収集されたデータを使用して、アダプターのランタイム中に発生する例外処理を識別することができます。

問題が発生した場合、アダプターは例外メッセージおよびコンテキスト・データを *install_root/profiles/profile/logs/ffdc* ディレクトリーにあるログ・ファイルに書き込みます。

First Failure Data Capture (FFDC) について詳しくは、WebSphere Process Server または WebSphere Enterprise Service Bus の文書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 および WebSphere Enterprise Service Bus は、他のタイプのフォールトに対応しますが、アダプターはビジネス・フォールト（この文書中では単にフォールトと呼びます）のみを生成します。すべての例外がフォールトになるわけではありません。フォールトは、アクション可能なエラー、つまり、アプリケーションの終了を必要としないリカバリー・アクションが可能なエラーに対して生成されます。例えば、アダプターは、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ったときか、Outbound 処理中に特定のエラーが発生した場合に、フォールトを生成します。

フォールト・ビジネス・オブジェクト

外部サービス・ウィザードで、アダプターが生成できるフォールトごとにビジネス・オブジェクトを作成します。さらに、ウィザードでは WBIFault スーパーセット・オブジェクトを作成します。このオブジェクトには、図 84 に示される message、errorCode、および primaryKeySet の属性など、すべてのフォールトに共通する情報が含まれます。

WBIFault	
message	string
errorCode	string
primaryKeySet	PrimaryKeyPairType []

図 84. WBIFault ビジネス・オブジェクトの構造

一部のフォールトには matchCount 属性が含まれ、エラーに関する追加情報を提供します。それ以外の場合は、WBIFault に、フォールトの処理に必要なすべての情報が含まれます。

WebSphere Adapter for Flat Filesにより、フォールトが使用可能になります。フォールトを手動で構成する必要はありません。アダプターには、ウィザードが作成する次のフォールト・ビジネス・オブジェクトが用意されています。

- DuplicateRecordFault

このフォールトは、指定したディレクトリーにファイルが既に存在する場合の Outbound Create 操作時に生成されます。

- RecordNotFoundFault

このフォールトは、指定したディレクトリーにファイルが存在しない場合の Append、Delete、Overwrite、および Retrieve 操作時に生成されます。

- MissingDataFault

Outbound 操作に渡されたビジネス・オブジェクトに必要なすべての属性がない場合、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、Create、Delete、Update、Retrieve、ApplyChanges および Exists の各操作に対して発生する可能性があります。

例えば、指定したファイルの内容が空白であるか、ファイル名やディレクトリ・パスが空の場合に、アダプターはこのフォールトをスローします。

XAResourceNotAvailableException

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

症状:

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

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

問題:

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

解決策:

この問題を訂正するには、以下の手順を実行します。

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

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

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

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

3. プロセス・サーバーを開始します。

org.xml.sax.SAXParseException

アダプターが XML データ・ハンドラーで構成されているときに、内容が指定されたビジネス・オブジェクト形式ではない場合は、org.xml.sax.SAXParseException 例外が生成されます。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ことを確認してください。ファイルに複数のビジネス・オブジェクトが含まれている場合、区切り文字が正しく指定されていることを確認してください。

症状:

アダプターが XML データ・ハンドラーで構成されているとき、以下の例外がスローされます。

org.xml.sax.SAXParseException: 内容が後続のセクションで許可されていません
(org.xml.sax.SAXParseException: Content is not allowed in trailing section)

問題:

ファイルの内容が、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ことを確認してください。
2. 内容ファイルに複数のビジネス・オブジェクトが含まれている場合、区切り文字が正しく指定されていることを確認してください。

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手したり、サポート・ツールやフィックスをダウンロードしたり、WebSphere Adapters の問題を回避したりする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したり、IBM ソフトウェア・サポートへの連絡方法を指定するのにも役立ちます。

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・ペーパーを含む技術情報
- 教育関連のオフライン
- テクニカル・ノート

推奨される修正

適用すべき、推奨される修正のリストは、ロケーション <http://www.ibm.com/support/docview.wss?fdoc=aimadp&rs=695&uid=swg27010397> で入手できます。

テクニカル・ノート

テクニカル・ノートは、Adapter for Flat Files 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・ノートのリストが必要な場合は、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>

IBM Support Assistant のプラグイン

Adapter for Flat Files 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関する情報については、以下のアドレスから参照してください。

<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・オブジェクト情報

ビジネス・オブジェクトの目的は、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と、ビジネス・オブジェクトの名前の両方を調べることによって判断できます。アプリケーション固有情報は、ローカル・ファイル・システムで実行可能な操作を示します。名前には通常、実行される操作とビジネス・オブジェクトの構造が反映されます。

ビジネス・オブジェクトの構造

Adapter for Flat Filesは、外部サービス時にビジネス・オブジェクトを定義および生成します。ビジネス・オブジェクト構造は、基本 XML スキーマとしてモデル化された、汎用 WebSphere Business Integration ビジネス・オブジェクト構造を基にしています。

汎用 FlatFileBG オブジェクト

エンタープライズ・メタデータ・ディスカバリー中には、2 つのタイプのビジネス・オブジェクト (コンテンツ固有および汎用) が生成されます。

汎用 FlatFileBG ビジネス・オブジェクトは、汎用 XSD ファイル (UnstructuredContent など) で使用されます。FlatFileBG ビジネス・オブジェクトは、FlatFile ビジネス・オブジェクトを子として含むラッパー・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。

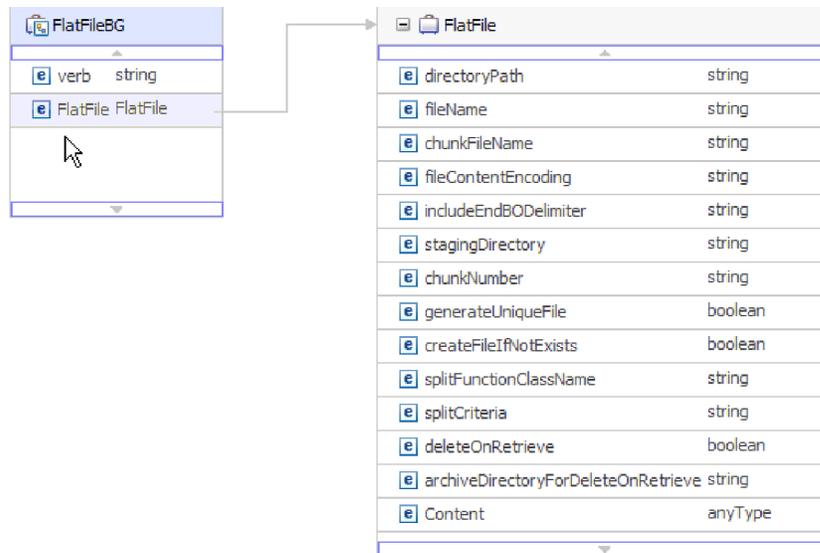


図 85. 汎用 FlatFileBG ビジネス・オブジェクト構造

CustomerWrapperBG オブジェクト

この例では、CustomerWrapperBG がコンテンツ固有の XSD ファイルを表しています。CustomerWrapperBG は、CustomerWrapper ビジネス・オブジェクトを子として含むラッパー・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。

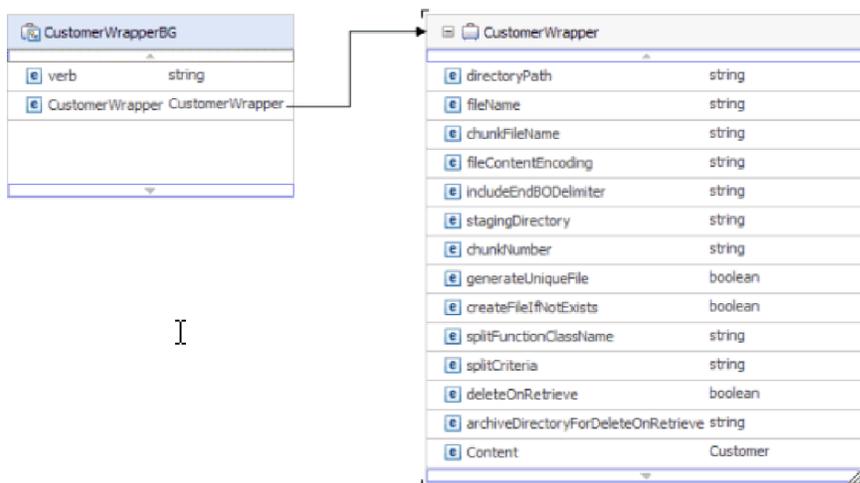


図 86. CustomerWrapperBG ビジネス・オブジェクト構造

Append 操作応答ビジネス・オブジェクト

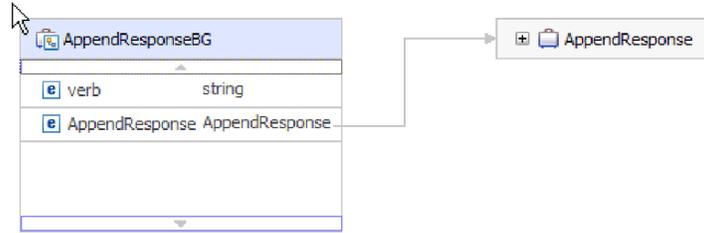


図 87. Append 操作応答ビジネス・オブジェクト構造

Create 操作応答ビジネス・オブジェクト

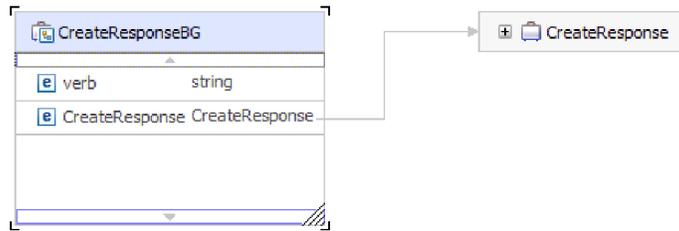


図 88. Create 操作応答ビジネス・オブジェクト構造

Exists 操作応答ビジネス・オブジェクト

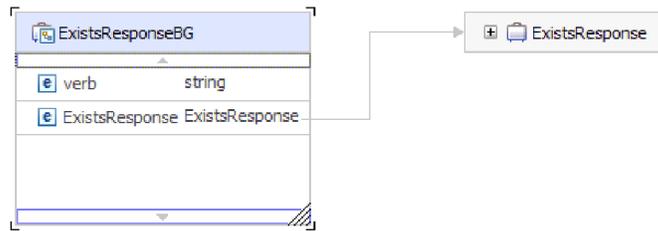


図 89. Exists 操作応答ビジネス・オブジェクト構造

List 操作応答ビジネス・オブジェクト

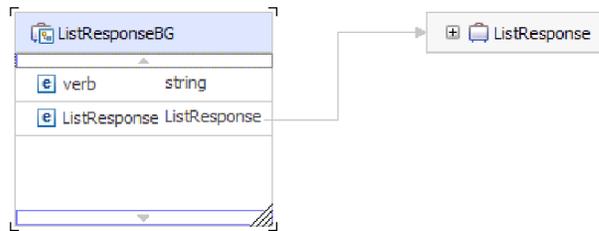


図 90. List 操作応答ビジネス・オブジェクト構造

Overwrite 操作応答ビジネス・オブジェクト



図 91. Overwrite 操作応答ビジネス・オブジェクト構造

Retrieve 操作応答ビジネス・オブジェクト

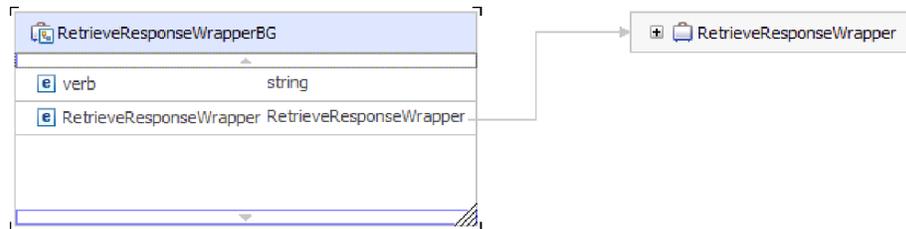


図 92. Retrieve 操作応答ビジネス・オブジェクト構造

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の説明を示します。

表 12. 属性プロパティ

属性プロパティ	説明
カーディナリティー	子ビジネス・オブジェクトまたは子ビジネス・オブジェクトの配列を表す各ビジネス・オブジェクト属性は、値 1 または複数 (n) のカーディナリティーを持ちます。単一の基数を持つフラット・ビジネス・オブジェクトだけがサポートされる。
キーおよび外部キー	これらの属性はアダプターで使用されません。
名前	属性の固有の名前を示します。
必須	この属性はアダプターで使用されません。

表 12. 属性プロパティ (続き)

属性プロパティ	説明
タイプ	<p>属性タイプは、単純または複合のいずれかです。</p> <p>単純型は、Boolean、String、LongText、Integer、Float、Double および Byte[] です。</p> <p>標準的な複合型は、別のビジネス・オブジェクト型です。</p>

命名規則

外部サービス・ウィザードは、ビジネス・オブジェクトを生成する場合、ビジネス・オブジェクトの作成に使用するローカル・ファイル・システムにあるオブジェクト名に基づいてビジネス・オブジェクトの名前を指定します。

外部サービス・ウィザード でビジネス・オブジェクトの名前が提供される場合、そのオブジェクト名は大/小文字混合に変換されます。これは、スペースや下線などのあらゆる分離文字が除去され、各単語の先頭文字が大文字にされ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 という名前のローカル・ファイル・システム・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、アダプターにとって意味のある値はありません。これは、ビジネス・オブジェクト名を変更しても、ビジネス・オブジェクトの動作は変わらないことを意味します。

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

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

以下の表に、外部サービス・ウィザードが Adapter for Flat Filesのビジネス・オブジェクトを生成するときに使用する命名規則の説明を示します。

表 13. 命名規則

エレメント	命名規則	例
ビジネス・グラフの名前	親ビジネス・オブジェクトを含むビジネス・グラフには、内包するビジネス・オブジェクトに従って名前が付けられ、その後に BG スtringが続きます。ビジネス・グラフが使用できるのは、ラッパーがある場合に限りです。CustomerWrapperBG は、CustomerWrapper ビジネス・オブジェクトを子として含むラッパー・ビジネス・オブジェクトです。	CustomerWrapperBG

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

カスタム・ファイル分割

分割論理を含むカスタム・クラスを実装することができます。アダプターは、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 に指定されるイベント・ファイルに存在するビジネス・オブジェクトの総数を戻します。

- `public void setSplitCriteria(String splitCriteria)`

このメソッドは、イベント・ファイル内のビジネス・オブジェクトの数に基づく splitCriteria を使用します。各ビジネス・オブジェクトは、next() 呼び出しの間に戻されます。

- `public void setLogUtils(LogUtils logUtils)`

このメソッドは、LogUtils オブジェクトを設定する際に使用されます。このオブジェクトは、ユーザーがファイルへのトレースの書き込みやメッセージの記録に使用できるクラスです。

- `public void setEncoding(String encoding)`

このメソッドは、イベント・ファイル内容のエンコード方式を設定する場合に使用します。このエンコード方式は、ファイル内容を読み取る際に使用します。SplitCriteria の場合にもこのエンコード方式を使用します。

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

このメソッドは、現行のビジネス・オブジェクト数を設定する場合に使用します。このメソッドにより、`next()` 呼び出しが行われた場合は常に、`currentPosition` に設定したビジネス・オブジェクト数が戻されます。このメソッドでは、`includeEndBODelimiter` パラメーターも使用します。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最後に `SplitCriteria` が組み込まれます。このメソッドは、すべての `next()` 呼び出しの前に呼び出す必要があります。これにより、`next()` メソッドは、このメソッドに設定したビジネス・オブジェクトにビジネス・オブジェクトの内容を戻します。

- イテレーターには、`hasNext()`、`next` および `remove()` の 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()` に設定したビジネス・オブジェクトの位置にビジネス・オブジェクトの内容を (`byte[]` として) 戻します。ビジネス・オブジェクトの位置を設定しないと、正常に実行されません。 `hasNext()` メソッドは、`setBODetails()` にビジネス・オブジェクトの位置が設定されているかどうかを示します。 `hasNext()` を呼び出す前に、`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・テーブルから削除中であるビジネス・オブジェクト・エンタリーごとに呼び出されます。このメソッド内のイベント・ファイルは削除しないでください。使用中であるリソースのクリーンアップのみを行ってください。
- `public boolean isSplitBySize()`

このメソッドは、イベント・ファイルの構文解析が、サイズまたは区切り文字のどちらに基づいて行われるかを示します。

関連概念

15 ページの『Inbound 処理』

Adapter for Flat Files は Inbound イベント処理をサポートします。これは、イベント (ファイルの作成や変更など) に指定された間隔で、ローカル・ファイル・システムのポーリングを行います。イベントを検出すると、イベント・データをビジネス・オブジェクトに変換し、これをモジュールに送信して処理します。

関連タスク

105 ページの『Inbound 処理のモジュールの構成』

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

Outbound 構成プロパティ

WebSphere Adapter for Flat Files 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があります。WebSphere Integration Developer または管理コンソールを使用してモジュールを WebSphere Process Server または WebSphere Enterprise Service Bus にデプロイした後に、リソース・アダプターおよ

び Managed Connection Factory のプロパティーを変更することができますが、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 Flat Files の構成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、Managed Connection Factory プロパティーなどの各構成プロパティー・トピックに含まれている表で詳しく説明されています。これらの表を使用するのに役立つように、表示される可能性のある各行についての情報が以下で説明されます。

以下の表は、構成プロパティーについての表に表示される可能性のある各行の意味を説明しています。

行	説明
必須	<p>アダプターが機能するためには、必須フィールド (プロパティー) に値が必要です。場合によっては、外部サービス・ウィザードによって必須プロパティーのデフォルト値が指定されることもあります。</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全く含まれていない場合、外部サービス・ウィザードは、割り当てられたデフォルト値を使用してフィールドを処理し、そのデフォルト値も管理コンソールに表示されます。</p> <p>指定可能な値は、Yes および No です。</p> <p>場合によっては、別のプロパティーで特定の値が指定された場合にのみ、必須のプロパティーになることもあります。この場合には、表にこの従属関係が示されます。以下に例を示します。</p> <ul style="list-style-type: none"> • Yes. EventQueryType プロパティーが Dynamic に設定された場合 • Yes. Oracle データベースの場合
使用可能な値	そのプロパティーで選択可能な値をリストし、説明を示します。
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。そのプロパティーが必須の場合は、デフォルト値を受け入れるか、または別の値を指定する必要があります。プロパティーにデフォルト値が設定されていない場合、表には「デフォルト値なし」と表示されます。</p> <p>None は許容されるデフォルト値であり、デフォルト値が存在しないという意味ではありません。</p>
計測単位	プロパティーの計測方法 (キロバイトや秒など) を指定します。
プロパティー・タイプ	<p>プロパティー・タイプについて説明します。有効なプロパティー・タイプは以下のとおりです。</p> <ul style="list-style-type: none"> • Boolean • String • Integer

行	説明
使用法	<p>プロパティに適用される使用条件または制約事項について説明しています。例えば、制約事項は次のように説明されます。</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>この項には、このプロパティに影響を与える他のプロパティ、およびこのプロパティによる影響を受けるプロパティのリストと、その条件関係の性質についての説明が示されます。</p>
例	<p>次のようなサンプル・プロパティ値が示されます。</p> <p>「言語を JA (日本語) に設定する場合、コード・ページ番号は 8000 に設定されます。」</p>
グローバル化	<p>グローバル化される場合、プロパティには各国語サポートがあるので、自国の言語に設定できます。</p> <p>有効値は Yes および No です。</p>
BIDI 対応	<p>プロパティが双方向 (bidi) 処理でサポートされているかどうかを示します。双方向処理は、1 つのファイルに右から左 (ヘブライ語やアラビア語など) と左から右 (URL やファイル・パスなど) の両方の意味構造を含むデータを処理するタスクに関係します。</p> <p>有効値は Yes および No です。</p>

ウィザードの接続プロパティ

接続プロパティはサービス記述を作成して、組み込みの成果物を保存するために使用されます。これらのプロパティは、外部サービス・ウィザードで構成されません。

以下の表に、外部サービス・ウィザードの接続プロパティのリストを示します。これらの構成は、外部サービス・ウィザードを使用してのみ行うことができ、デプロイメント後には変更できません。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、172 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 14. 外部サービス・ウィザードの接続プロパティ

ウィザードのプロパティ名	説明
174 ページの『BiDi フォーマット・ストリング』	コンテンツ・データの BiDi フォーマット・ストリング
174 ページの『データ・バインディング』	すべての操作で使用されるデータ・バインディングを指定するか、操作別にデータ・バインディングを選択することを指定します。

表 14. 外部サービス・ウィザードの接続プロパティ (続き)

ウィザードのプロパティ名	説明
『関数セクター』	Inbound 処理中に使用される関数セクター構成の名前。
175 ページの『ログ・ファイルの出力ロケーション』	外部サービス・ウィザードで生成されるログ・ファイルの絶対パス名
175 ページの『ロギング・レベル』	アダプターで使用されるロギングのレベル
176 ページの『ネーム・スペース』	生成されるビジネス・オブジェクトのネーム・スペース
176 ページの『操作名』	外部サービス・ウィザードで定義される操作
176 ページの『処理指示』	Inbound または Outbound の処理指示

BiDi フォーマット・ストリング

コンテンツ・データの BiDi フォーマット・ストリング。

表 15. BiDi フォーマット・ストリング

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String

データ・バインディング

すべての操作で使用されるデータ・バインディングを指定するか、操作別にデータ・バインディングを選択することを指定します。

表 16. データ・バインディングの詳細

必須	いいえ
デフォルト	すべての操作にデフォルトのデータ・バインディング FlatFileBaseDataBinding を使用する
使用法	このプロパティの値は以下のように設定できます。 <ul style="list-style-type: none"> すべての操作にデフォルトのデータ・バインディング FlatFileBaseDataBinding を使用する すべての操作に 1 つのデータ・バインディング構成を使用 操作ごとにデータ・バインディングを指定
グローバル化	いいえ
BIDI 対応	いいえ

関数セクター

Inbound 処理中に使用される関数セクター構成の名前。

表 17. 関数セクターの詳細

必須	はい
デフォルト	FilenameFunctionSelector
プロパティ・タイプ	String

表 17. 関数セクターの詳細 (続き)

使用法	<p>関数セクターはサービスで呼び出される適切な操作を返します。アダプターは、<code>FilenameFunctionSelector</code> および <code>EmbeddedNameFunctionSelector</code> の 2 つの関数セクターを提供します。</p> <ul style="list-style-type: none"> <code>FilenameFunctionSelector</code> は、ファイル名の正規表現をオブジェクト名と突き合わせる、ルール・ベースの関数セクターです。<code>FilenameFunctionSelector</code> は汎用 <code>FlatFile</code> ビジネス・オブジェクトに使用されますが、この場合、オブジェクト名はイベント・ファイルから決定できません。 <p><code>FilenameFunctionSelector</code> は、<i>N</i> 行からなる 2 列のテーブルとしてプロパティーで表されません。<code>.txt</code> 拡張子を持つイベント・ファイルの場合、対応するオブジェクト名は <code>FlatFile</code> になり、関数セクターで生成されるエンドポイント・メソッド名は <code>emitFlatFile</code> になります。操作を追加した後、この同じ名前を <code>EISFunctionName</code> プロパティーに設定する必要があります。</p> <p>それぞれがオブジェクト名とファイル名と突き合わせる正規表現を含む複数のルールを使用して、<code>FilenameFunctionSelector</code> を構成することができます。複数のルールが一致する場合、関数セクターは最初に一致したルールに基づいてオブジェクト名を返します。</p> <ul style="list-style-type: none"> <code>EmbeddedNameFunctionSelector</code> はコンテンツ固有のビジネス・オブジェクトに使用されますが、この場合、オブジェクト名はイベント・ファイルに埋め込まれています。<code>EmbeddedNameFunctionSelector</code> は、ラッパーではなく、必要なコンテンツ・データに基づく関数名を返します。例えば、コンテンツ固有のビジネス・オブジェクトが <code>CustomerWrapperBG</code> の場合、関数セクターで戻される関数は <code>emitCustomer</code> です。 <p><code>EmbeddedNameFunctionSelector</code> はデータ・ハンドラーと一緒に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 <code>WrapperDataBinding</code> でなければならず、関数セクターと一緒に構成されたものと同じデータ・ハンドラーを使用するように構成する必要があります。</p>
グローバル化	はい
BIDI 対応	いいえ

ログ・ファイルの出力ロケーション

外部サービス・ウィザードで生成されるログ・ファイルの絶対パス名。

表 18. 「ログ・ファイルの出力ロケーション」の詳細

必須	いいえ
デフォルト	<code>¥.metadata ¥FlatFileMetadataDiscoveryImpl.log</code>
プロパティー・タイプ	<code>String</code>
使用法	
グローバル化	いいえ
BIDI 対応	いいえ

ロギング・レベル

アダプターで使用されるロギングのレベル。

表 19. ログイン・レベルの詳細

必須	いいえ
使用可能な値	重大 警告 監査 情報 構成 詳細
デフォルト	重大
プロパティ・タイプ	値リスト
グローバル化	いいえ
BIDI 対応	いいえ

ネーム・スペース

生成されるビジネス・オブジェクトのネーム・スペース。

表 20. 「ネーム・スペース」の詳細

必須	はい
デフォルト	http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile
プロパティ・タイプ	String
グローバル化	はい
BIDI 対応	いいえ

操作名

このモジュールに定義された操作に指定する名前。

表 21. 操作名の詳細

必須	いいえ
デフォルト	ServiceType プロパティを Outbound に設定すると、操作 Create、Append、Retrieve、Delete、List、Overwrite、および Exists がリストされます。
プロパティ・タイプ	String
グローバル化	いいえ
BIDI 対応	いいえ

処理指示

Inbound または Outbound の処理指示

表 22. 処理指示の詳細

必須	はい
使用可能な値	Outbound Inbound

表 22. 処理指示の詳細 (続き)

デフォルト	Outbound
プロパティ・タイプ	String
グローバル化	いいえ
BIDI 対応	いいえ

関連概念

218 ページの『グローバリゼーションおよび双方向データ変換』
 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。アダプターは双方向のスクリプト・データ変換も実行します。双方向変換とは、1 つのファイルに右から左 (ヘブライ語やアラビア語など) と左から右 (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のことを指します。

21 ページの『関数セクター』

Inbound 処理中に、関数セクターはサービスで呼び出される適切な操作を返します。外部サービス・ウィザードで Inbound 処理用にアダプターを構成した場合は、関数セクターを選択してください。アダプターは、FilenameFunctionSelector および EmbeddedNameFunctionSelector の 2 つの関数セクターを提供します。

関連タスク

156 ページの『ロギング・プロパティの構成』

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

Managed Connection Factory プロパティ

Managed Connection Factory プロパティでは、アダプターがローカル・ファイル・システムとの Outbound 通信の実行時に必要とする情報を指定します。

以下の表は、Outbound 通信用の Managed Connection Factory プロパティをリストしています。外部サービス・ウィザードを使用して Managed Connection Factory プロパティを設定します。これらのプロパティは、WebSphere Integration Developer アセンブリ・エディターを使用して変更することも、WebSphere Process Server または WebSphere Enterprise Service Bus 管理コンソールを使用してデプロイ後に変更することもできます。

各プロパティの詳細については、表の後の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、172 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。

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 または WebSphere Enterprise Service Bus 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 23. 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。
179 ページの『デフォルト・ターゲット・ファイル名』	OutputFileName	出力ディレクトリーに作成されるファイルの名前、またはこのファイルを表す WebSphere Application Server 環境変数
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	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。
180 ページの『出力ディレクトリー』	OutputDirectory	Outbound 操作時にアダプターがファイルを作成するディレクトリーの絶対パス名、またはこのファイルを表す WebSphere Application Server 環境変数
181 ページの『シーケンス・ファイル』	FileSequenceLog	Outbound Create 操作時にシーケンスが保管されるファイルの絶対パス名、またはこのファイルを表す WebSphere Application Server 環境変数
182 ページの『ステージング・ディレクトリー』	StagingDirectory	Outbound 処理時に、アダプターが Create および Overwrite 操作の初期出力ファイルを書き込む一時ディレクトリーの絶対パス名、またはこのディレクトリーを表す WebSphere Application Server 環境変数

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 24. 「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 24. 「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、FFRA 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、FFRA001 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 9 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 Flat Files 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、FFRA001 および FFRA002 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 FFRAInstance0 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、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>

デフォルト・ターゲット・ファイル名

出力ディレクトリーに作成されるファイルの名前、またはこのファイルを表す WebSphere Application Server 環境変数です。

表 25. 「デフォルト・ターゲット・ファイル名」の詳細

<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>

表 25. 「デフォルト・ターゲット・ファイル名」の詳細 (続き)

使用法	OutputFileName の値がレコード・オブジェクトで指定された場合、この値はオーバーライドされます。WebSphere Application Server 環境変数を使用して、デフォルトのターゲット・ファイル名を示すことができます。環境変数の名前を中括弧で囲み、前に \$ 記号を付けて指定します。例えば、\${OUTPUT_FILENAME} のようになります。この資料の環境変数の作成に関する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
BIDI 対応	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 26. 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
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

出力ディレクトリー

Outbound 操作時にアダプターがファイルを作成するディレクトリーの絶対パス名、またはこのファイルを表す WebSphere Application Server 環境変数です。

表 27. 「出力ディレクトリー」の詳細

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String

表 27. 「出力ディレクトリー」の詳細 (続き)

使用法	出力ディレクトリーは、最終出力ファイルを書き込むためにアダプターが使用します。WebSphere Application Server 環境変数を使用して、出力ディレクトリーを示すことができます。環境変数の名前を中括弧で囲み、前に \$ 記号を付けて指定します。例えば、\${OUTPUT_DIRECTORY} のようになります。この資料の環境変数の作成に関する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
BIDI 対応	はい

シーケンス・ファイル

このプロパティーは、Outbound Create 操作時にシーケンスが保管されるファイルの絶対パス名、またはこのファイルを表す WebSphere Application Server 環境変数を指定します。

表 28. 「シーケンス・ファイル」の詳細

必須	いいえ
デフォルト	なし
プロパティー・タイプ	String

表 28. 「シーケンス・ファイル」の詳細 (続き)

<p>使用法</p>	<p>アダプターは Create 要求を受け取ると、ファイル・シーケンス・ログを検査し、指定した名前のファイルが既に存在するかどうかを検査します。その名前のファイルが存在する場合、アダプターはファイル・シーケンス番号を使用して新規ファイル名を作成します。例えば、要求内の出力ファイル名が Customer.txt の場合、アダプターは Customer.n.txt (n はシーケンス番号) という名前のファイルを作成します。出力ファイル名に拡張子がない場合は、ファイル名の後にシーケンスが付加されず (例えば、Customerm)。すべてのシーケンスは 1 から始まります。</p> <p>このプロパティーが指定されていないときに、既に存在している名前で作成する要求を受け取ると、アダプターは DuplicateRecordException エラーを生成します。</p> <p>シーケンス番号は、アダプターの再始動後まで増分され続けます。ファイル・シーケンスは、シーケンス・ファイル内のシーケンス値を変更することによってリセットできます。</p> <p>注:</p> <ol style="list-style-type: none"> 1. 特定の要求タイプに対するファイル・シーケンスを生成するには、管理接続レベルで出力ディレクトリーとファイル名を設定します。 2. アダプターをクラスター環境で使用する場合は、シーケンス・ファイルが、すべてのクラスターからアクセス可能なマップされたドライブに存在することを確認してください。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許可が必要です。この許可がない場合は、IOException エラーが返されます。 3. FileSequenceLog プロパティーが指定されており、GenerateUniqueFile プロパティーが有効になっている場合は、GenerateUniqueFile プロパティーが FileSequenceLog プロパティーより優先されます。 4. ビジネス・オブジェクトで指定されるディレクトリー・パスとファイル名は、管理接続レベルで指定される値よりも優先されます。 <p>WebSphere Application Server 環境変数を使用して、シーケンス・ファイルを示すことができます。環境変数の名前を中括弧で囲み、前に \$ 記号を付けて指定します。例えば、\${SEQUENCE_FILE} のようになります。この資料の環境変数の作成に関するトピックを参照してください。</p> <p>重要: それらがクラスターの一部でない場合、2 つのアダプター・インスタンスは同じシーケンス・ファイルを使用してはなりません。これにより、バッチ要求の処理が遅延する可能性があるからです。</p>
<p>グローバル化されるかどうか</p>	<p>はい</p>
<p>BIDI 対応</p>	<p>はい</p>

ステージング・ディレクトリー

Outbound 処理時に、アダプターが Create および Overwrite 操作の初期出力ファイルを書き込む一時ディレクトリーの絶対パス名、またはこのディレクトリーを表す WebSphere Application Server 環境変数。

表 29. 「ステージング・ディレクトリー」の詳細

<p>必須</p>	<p>いいえ</p>
<p>デフォルト</p>	<p>なし</p>
<p>プロパティー・タイプ</p>	<p>String</p>

表 29. 「ステージング・ディレクトリー」の詳細 (続き)

使用法	<p>このプロパティーを指定すると、出力ファイルは最初にステージング・ディレクトリーに書き込まれ、次いで名前変更されて出力ディレクトリーに書き込まれます。Outbound 処理時の書き込み競合を防ぐため、アダプターは、Create および Overwrite 操作の初期出力ファイルをステージング・ディレクトリーに一時的に保管します。</p> <p>WebSphere Application Server 環境変数を使用して、ステージング・ディレクトリーを示すことができます。環境変数の名前を中括弧で囲み、前に \$ 記号を付けて指定します。例えば、<code>\${STAGING_DIRECTORY}</code> のようになります。この資料の環境変数の作成に関するトピックを参照してください。</p>
グローバル化されるかどうか	はい
BIDI 対応	はい

関連概念

28 ページの『WebSphere Application Server 環境変数』

外部サービス・ウィザードで WebSphere Application Server 環境変数を使用して、ディレクトリー値を指定できます。

55 ページの『必須のローカル・フォルダーの作成』

Inbound モジュールまたは Outbound モジュールを作成する前に、イベントおよび出力用のフォルダーをローカル・ファイル・システムに作成する必要があります。またオプションで、ステージングおよびアーカイブ用のフォルダーを作成することができます。

関連タスク

58 ページの『WebSphere Application Server 環境変数の定義』

WebSphere Process Server または WebSphere Enterprise Service Bus 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。

リソース・アダプター・プロパティー

リソース・アダプター・プロパティーでは、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・アダプターのプロパティーは、アダプターの構成時に外部サービス・ウィザードを使用して設定します。アダプターをデプロイしたあとは、管理コンソールを使用してこれらのプロパティーを変更します。

以下に示すロギングおよびトレースのプロパティーは、バージョン 6.1.0 では必要なくなりました。ただし、旧バージョンとの互換性を維持するため、管理コンソールで表示することは可能です。

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

以下の表に、リソース・アダプター・プロパティーとその目的のリストを示します。各プロパティーの完全な説明は、表に続くセクションで説明します。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、172 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 30. Adapter for Flat Files 用のリソース・アダプター・プロパティ

名前		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。
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	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)	HA サポートの使用可能化	このプロパティは変更しないでください。
(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
(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 31. 「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 31. 「アダプター ID」の詳細 (続き)

<p>使用法</p>	<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、FFRA 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、FFRA001 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 9 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 Flat Files 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、FFRA001 および FFRA002 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 FFRAInstance0 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>
<p>グローバル化</p>	<p>はい</p>
<p>BIDI 対応</p>	<p>いいえ</p>

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 32. 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

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

表 32. 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

高可用性サポートを使用可能にする (Enable high availability support) (enableHASupport)

このプロパティは変更しないでください。true に設定してください。

対話仕様プロパティ

対話仕様プロパティには、アダプターがファイル・システムとのインターフェースとなるときに使用する Outbound 接続プロパティが含まれています。外部サービス・ウィザード を使用して、これらのプロパティを構成します。アプリケーションがデプロイされた後に対話仕様プロパティを変更するには、WebSphere Integration Developer でアセンブリー・エディターを使用します。

対話仕様プロパティでは、操作での対話方法を制御します。対話仕様プロパティは、アダプターを構成するときに外部サービス・ウィザードを使用して設定します。通常は、これらのプロパティを変更する必要はありません。ただし、Outbound 操作の一部のプロパティは、ユーザーが変更可能です。アプリケーションがデプロイされた後にこれらのプロパティを変更するには、WebSphere Integration Developer でアセンブリー・エディターを使用します。プロパティは、インポートのメソッド・バインディング内にあります。

以下の表に、対話仕様プロパティのリストを示します。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は、172 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 33. 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	
187 ページの『retrieve 操作用のアーカイブ・ディレクトリー』	ArchiveDirectoryForDeleteOnRetrieve	DeleteOnRetrieve プロパティが true に設定されている場合に、取り出されたファイルが削除されるまでの間、それらを保管しておくディレクトリー

表 33. 対話仕様プロパティ (続き)

プロパティ名		説明
ウィザード内	管理コンソール内	
『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	このプロパティを true に設定すると、ファイルが存在しない場合に、アダプターは Append および Overwrite 操作時に新規ファイルを作成します。
188 ページの『デフォルト・ターゲット・ファイル名』	OutputFileName	作成または変更される出力ファイルの名前
188 ページの『retrieve 操作後のファイルの削除』	DeleteOnRetrieve	このプロパティを true に設定すると、Retrieve 操作時に、ファイル内容が取り出された後にファイルがファイル・システムから削除されます
189 ページの『ファイル内のビジネス・オブジェクト間の区切り文字』	IncludeEndBODelimiter	ファイル内容は、この値と一緒に付加されます。
189 ページの『ファイル内容のエンコード』	FileContentEncoding	イベント・ファイルの読み書きに使用するエンコード設定を指定します。
189 ページの『固有ファイルの生成』	GenerateUniqueFile	Create、Append、および Overwrite 操作時に、アダプターが固有ファイルを作成するように指定します。
190 ページの『出力ディレクトリ』	OutputDirectory	アダプターが出力ファイルを書き込む、ローカル・ファイル・システム上のディレクトリの絶対パス名。
190 ページの『ファイル内容を分割するための基準の指定』	SplitCriteria	取り出されたファイルのビジネス・オブジェクトを分離する区切り文字、または取り出されたファイルを分割するときのチャンク・サイズを指定します。
191 ページの『関数クラス名の分割』	SplittingFunctionClassName	Outbound Retrieve 操作時に、取り出されるファイルの分割方法 (区切り文字ごと、またはサイズごと) を指定します。
192 ページの『ステージング・ディレクトリ』	StagingDirectory	アダプターが Create および Overwrite 操作時に初期出力ファイルを保管する一時ディレクトリ

retrieve 操作のアーカイブ・ディレクトリ

DeleteOnRetrieve プロパティが true に設定されている場合に、取り出されたファイルが削除されるまでの間、それらを保管しておくディレクトリ。

表 34. 「retrieve 操作のアーカイブ・ディレクトリ」の詳細

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	はい
BIDI 対応	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

このプロパティを true に設定すると、ファイルが存在しない場合に、アダプターは Append および Overwrite 操作時に新規ファイルを作成します。

表 35. 「ファイルが存在しない場合に新規ファイルを作成する」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
使用法	このプロパティが false に設定されており、ファイルが存在しない場合、アダプターは RecordNotFoundException エラーを生成します。 注: ラッパーのこのプロパティに値が設定されていないと、ここに設定されている値が使用されます。
グローバル化	いいえ
BIDI 対応	いいえ

デフォルト・ターゲット・ファイル名

作成または変更される出力ファイルの名前。

表 36. 「デフォルト・ターゲット・ファイル名」の詳細

必須	List 操作を除くすべての Outbound 操作で必須
デフォルト	なし
プロパティタイプ	String
グローバル化	はい
BIDI 対応	はい

retrieve 操作後のファイルの削除

Retrieve 操作時に、このプロパティが true に設定されると、ファイル内容が取り出された後にファイル・システムからファイルが削除されます。

表 37. 「retrieve 操作後のファイルの削除」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	削除される前にファイルをアーカイブするには、ArchiveDirectoryForDeleteOnRetrieve プロパティでディレクトリーを指定します。 注: ラッパーのこのプロパティに値が設定されていないと、ここに設定されている値が使用されます。
グローバル化	いいえ
BIDI 対応	いいえ

ファイル内のビジネス・オブジェクト間の区切り文字

ファイル内容は、この値と一緒に付加されます。

表 38. ファイル内のビジネス・オブジェクト間の区切り文字の詳細

必須	いいえ
デフォルト	なし
プロパティタイプ	String
使用法	このプロパティは、Outbound の Create、Append、および Overwrite 操作時に使用します。このプロパティに指定した値はファイルに付加されます。指定された値にエスケープ・シーケンス文字やユニコード・エスケープ文字があっても、それらは解析されて対応する制御文字がファイルに挿入されます。エスケープ・シーケンス文字には、復帰 (¥r)、改行 (¥n)、復帰改行 (¥r¥n)、タブ・スペース (¥t)、バックスペース (¥b)、用紙送り (¥f) などがあります。例としては、「?」を表すユニコード・エスケープ文字「¥u2297」です。
グローバル化	はい
BIDI 対応	はい

ファイル内容のエンコード

イベント・ファイルの読み書きに使用するエンコード設定。

注: Create 操作時に、アダプターは指定したエンコード方式でファイルを作成します。

表 39. 「ファイル内容のエンコード」の詳細

必須	いいえ
使用可能な値	Java でサポートされるエンコードされた文字セット。
デフォルト	UTF-8
プロパティタイプ	String
使用法	Java でサポートされているエンコード・セット (UTF-8 など) を指定できます。アダプターでバイナリー・イベント・データを処理する場合は、このプロパティを BINARY に設定します。アダプターがテキストや XML などのバイナリー以外のイベント・データを処理している場合は、このプロパティを UTF-8 または UTF-16 などの有効なファイル・エンコード方式の値に設定します。 注: 対話仕様プロパティに設定された値は、ラッパーに値が設定されていない場合に限り使用されます。
グローバル化	いいえ
BIDI 対応	いいえ

固有ファイルの生成

Create、Append、および Overwrite 操作時に、アダプターが固有ファイルを作成するように指定します。

表 40. 「固有ファイルの生成」の詳細

必須	いいえ
----	-----

表 40. 「固有ファイルの生成」の詳細 (続き)

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、アダプターは Create 操作時に固有ファイルを作成し、ファイル名プロパティーに設定されている値をすべて無視します。 注: ラッパーのこのプロパティーに値が設定されていないと、ここに設定されている値が使用されます。
グローバル化	はい
BIDI 対応	いいえ

出力ディレクトリー

アダプターが出力ファイルを書き込む、ローカル・ファイル・システム上のディレクトリーの絶対パス名。

表 41. 「出力ディレクトリー」の詳細

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	このプロパティーを指定しない場合、アダプターは、要求時に OutputFileName プロパティーによって指定されたディレクトリーに出力ファイルを書き込みます。
グローバル化	はい
BIDI 対応	はい

ファイル内容を分割するための基準の指定

このプロパティーにより、取り出されたファイルのビジネス・オブジェクトを分離する区切り文字、または取り出されたファイルを分割するときのチャンク・サイズを指定します。

表 42. 「ファイル内容を分割するための基準の指定」の詳細

必須	いいえ
使用可能な値	区切り文字または有効な数値
デフォルト	0
プロパティー・タイプ	String

表 42. 「ファイル内容を分割するための基準の指定」の詳細 (続き)

使用法	<p>このプロパティにより、取り出されたファイルのビジネス・オブジェクトを分離する区切り文字、または取り出されたファイルを分割するときのチャンク・サイズを指定します。このプロパティの値は、<code>SplittingFunctionClassName</code> プロパティに設定された値によって決まります。</p> <ul style="list-style-type: none"> • <code>SplittingFunctionClassName</code> プロパティが <code>com.ibm.j2ca.utils.filesplit.SplitByDelimiter</code> に設定されている場合は、<code>SplitCriteria</code> プロパティに、取り出されたファイル内のビジネス・オブジェクトを分離する区切り文字が含まれている必要があります。 • <code>SplittingFunctionClassName</code> プロパティが <code>com.ibm.j2ca.utils.filesplit.SplitBySize</code> に設定されている場合は、<code>SplitCriteria</code> プロパティに、バイト単位のサイズを表す有効な数値が含まれている必要があります。取り出されたファイルのサイズがこの値よりも大きい場合、イベント・ファイルはこの値のチャンクに分割され、そのチャンクの数を送られます。ファイルのサイズがこの値よりも小さい場合、イベント・ファイル全体が送られます。 <p><code>SplitCriteria</code> プロパティを 0 に設定すると、チャンク化が無効になります。</p> <p><code>SplitCriteria</code> プロパティには、イベント・ファイルの場合と同じ改行文字の値が含まれている必要があります。例えば、イベント・ファイルが Macintosh システムで作成された場合、改行文字は <code>¥r</code> であり、<code>SplitCriteria</code> プロパティには <code>¥r</code> が含まれている必要があります。プラットフォーム固有の改行文字は以下のとおりです。</p> <p>Macintosh - <code>¥r</code></p> <p>Microsoft Windows - <code>¥r¥n</code></p> <p>UNIX - <code>¥n</code></p> <p><code>SplitCriteria</code> プロパティに複数の区切り文字がある場合は、それぞれの区切り文字をセミコロン (;) で分離する必要があります。セミコロン (;) 自体も区切り文字の一部になっている場合は、<code>¥;</code> のようにしてセミコロン (;) をエスケープしてください。例えば、区切り文字が <code>##¥;##</code> の場合は、<code>##;##</code> と見なされます。</p>
グローバル化	はい
BIDI 対応	はい

関数クラス名の分割

このプロパティにより、Outbound Retrieve 操作時に取り出されるファイルの分割方法 (区切り文字ごと、またはサイズごと) を指定します。

表 43. 「関数クラス名の分割」の詳細

必須	いいえ
使用可能な値	<p><code>com.ibm.j2ca.utils.filesplit.SplitByDelimiter</code> ファイルは、イベント・ファイル内のビジネス・オブジェクトを分離する区切り文字に基づいて分割されます。</p> <p><code>com.ibm.j2ca.utils.filesplit.SplitBySize</code> ファイルは、イベント・ファイルのサイズに基づいて分割されます。</p>
デフォルト	<code>com.ibm.j2ca.utils.filesplit.SplitBySize</code>
プロパティ・タイプ	String
使用法	区切り文字またはファイル・サイズは、 <code>SplitCriteria</code> プロパティで設定します。
グローバル化	いいえ
BIDI 対応	いいえ

ステージング・ディレクトリー

書き込み競合を防ぐため、アダプターが Create および Overwrite 操作時に初期出力ファイルを保管する一時ディレクトリー。

表 44. 「ステージング・ディレクトリー」の詳細

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	ステージング・ディレクトリーを指定すると、出力ディレクトリーからステージング・ディレクトリーに操作対象のファイルがコピーされます。ステージング・ディレクトリーのファイルに対して操作が実行されたあと、そのファイルは名前変更され、出力ディレクトリーにコピーされます。
グローバル化	はい
BIDI 対応	はい

関連タスク

123 ページの『第 5 章 アセンブリー・エディターの使用による対話仕様プロパティーの変更』

サービスを生成した後、アダプター・モジュールの対話仕様プロパティーを変更するには、WebSphere Integration Developer のアセンブリー・エディターを使用します。

Inbound 構成プロパティー

WebSphere Adapter for Flat Files 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 Inbound 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用してモジュールをデプロイした後に、リソース・アダプターおよび活動化仕様のプロパティーを変更することができますが、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 Flat Files の構成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、Managed Connection Factory プロパティーなどの各構成プロパティー・トピックに含まれている表で詳しく説明されています。これらの表を使用するのに役立つように、表示される可能性のある各行についての情報が以下で説明されます。

以下の表は、構成プロパティーについての表に表示される可能性のある各行の意味を説明しています。

行	説明
必須	<p>アダプターが機能するためには、必須フィールド (プロパティ) に値が必要です。場合によっては、外部サービス・ウィザードによって必須プロパティのデフォルト値が指定されることもあります。</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全く含まれていない場合、外部サービス・ウィザードは、割り当てられたデフォルト値を使用してフィールドを処理し、そのデフォルト値も管理コンソールに表示されます。</p> <p>指定可能な値は、Yes および No です。</p> <p>場合によっては、別のプロパティで特定の値が指定された場合にのみ、必須のプロパティになることもあります。この場合には、表にこの従属関係が示されます。以下に例を示します。</p> <ul style="list-style-type: none"> • Yes。EventQueryType プロパティが Dynamic に設定された場合 • Yes。Oracle データベースの場合
使用可能な値	そのプロパティで選択可能な値をリストし、説明を示します。
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。その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、または別の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設定されていない場合、表には「デフォルト値なし」と表示されます。</p> <p>None は許容されるデフォルト値であり、デフォルト値が存在しないという意味ではありません。</p>
計測単位	プロパティの計測方法 (キロバイトや秒など) を指定します。
プロパティ・タイプ	<p>プロパティ・タイプについて説明します。有効なプロパティ・タイプは以下のとおりです。</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使用条件または制約事項について説明しています。例えば、制約事項は次のように説明されます。</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>この項には、このプロパティに影響を与える他のプロパティ、およびこのプロパティによる影響を受けるプロパティのリストと、その条件関係の性質についての説明が示されます。</p>

行	説明
例	次のようなサンプル・プロパティ値が示されます。 「言語を JA (日本語) に設定する場合、コード・ページ番号は 8000 に設定されます。」
グローバル化	グローバル化される場合、プロパティには各国語サポートがあるので、自国の言語に設定できます。 有効値は Yes および No です。
BIDI 対応	プロパティが双方向 (bidi) 処理でサポートされているかどうかを示します。双方向処理は、1 つのファイルに右から左 (ヘブライ語やアラビア語など) と左から右 (URL やファイル・パスなど) の両方の意味構造を含むデータを処理するタスクに関係します。 有効値は Yes および No です。

ウィザードの接続プロパティ

接続プロパティはサービス記述を作成して、組み込みの成果物を保存するために使用されます。これらのプロパティは、外部サービス・ウィザードで構成されます。

以下の表に、外部サービス・ウィザードの接続プロパティのリストを示します。これらの構成は、外部サービス・ウィザードを使用してのみ行うことができ、デプロイメント後には変更できません。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は、172 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 45. 外部サービス・ウィザードの接続プロパティ

ウィザードのプロパティ名	説明
『BiDi フォーマット・ストリング』	コンテンツ・データの BiDi フォーマット・ストリング
195 ページの『データ・バインディング』	すべての操作で使用されるデータ・バインディングを指定するか、操作別にデータ・バインディングを選択することを指定します。
195 ページの『関数セクター』	Inbound 処理中に使用される関数セクター構成の名前。
196 ページの『ログ・ファイルの出力ロケーション』	外部サービス・ウィザードで生成されるログ・ファイルの絶対パス名
196 ページの『ロギング・レベル』	アダプターで使用されるロギングのレベル
197 ページの『ネーム・スペース』	生成されるビジネス・オブジェクトのネーム・スペース
197 ページの『操作名』	外部サービス・ウィザードで定義される操作
197 ページの『処理指示』	Inbound または Outbound の処理指示

BiDi フォーマット・ストリング

コンテンツ・データの BiDi フォーマット・ストリング。

表 46. BiDi フォーマット・ストリング

必須	いいえ
----	-----

表 46. BiDi フォーマット・ストリング (続き)

デフォルト	なし
プロパティ・タイプ	String

データ・バインディング

すべての操作で使用されるデータ・バインディングを指定するか、操作別にデータ・バインディングを選択することを指定します。

表 47. データ・バインディングの詳細

必須	いいえ
デフォルト	すべての操作にデフォルトのデータ・バインディング <code>FlatFileBaseDataBinding</code> を使用する
使用法	このプロパティの値は以下のように設定できます。 <ul style="list-style-type: none"> • すべての操作にデフォルトのデータ・バインディング <code>FlatFileBaseDataBinding</code> を使用する • すべての操作に 1 つのデータ・バインディング構成を使用 • 操作ごとにデータ・バインディングを指定
グローバル化	いいえ
BIDI 対応	いいえ

関数セクター

Inbound 処理中に使用される関数セクター構成の名前。

表 48. 関数セクターの詳細

必須	はい
デフォルト	<code>FilenameFunctionSelector</code>
プロパティ・タイプ	String

表 48. 関数セクターの詳細 (続き)

<p>使用法</p>	<p>関数セクターはサービスで呼び出される適切な操作を返します。アダプターは、<code>FilenameFunctionSelector</code> および <code>EmbeddedNameFunctionSelector</code> の 2 つの関数セクターを提供します。</p> <ul style="list-style-type: none"> <code>FilenameFunctionSelector</code> は、ファイル名の正規表現をオブジェクト名と突き合わせる、ルール・ベースの関数セクターです。 <code>FilenameFunctionSelector</code> は汎用 <code>FlatFile</code> ビジネス・オブジェクトに使用されますが、この場合、オブジェクト名はイベント・ファイルから決定できません。 <p><code>FilenameFunctionSelector</code> は、<i>N</i> 行からなる 2 列のテーブルとしてプロパティーで表されません。 <code>.txt</code> 拡張子を持つイベント・ファイルの場合、対応するオブジェクト名は <code>FlatFile</code> になり、関数セクターで生成されるエンドポイント・メソッド名は <code>emitFlatFile</code> になります。操作を追加した後、この同じ名前を <code>EISFunctionName</code> プロパティーに設定する必要があります。</p> <p>それぞれがオブジェクト名とファイル名と突き合わせる正規表現を含む複数のルールを使用して、<code>FilenameFunctionSelector</code> を構成することができます。複数のルールが一致する場合、関数セクターは最初に一致したルールに基づいてオブジェクト名を返します。</p> <ul style="list-style-type: none"> <code>EmbeddedNameFunctionSelector</code> はコンテンツ固有のビジネス・オブジェクトに使用されますが、この場合、オブジェクト名はイベント・ファイルに埋め込まれています。 <p><code>EmbeddedNameFunctionSelector</code> は、ラッパーではなく、必要なコンテンツ・データに基づく関数名を返します。例えば、コンテンツ固有のビジネス・オブジェクトが <code>CustomerWrapperBG</code> の場合、関数セクターで戻される関数は <code>emitCustomer</code> です。</p> <p><code>EmbeddedNameFunctionSelector</code> はデータ・ハンドラーと一緒に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 <code>WrapperDataBinding</code> でなければならず、関数セクターと一緒に構成されたものと同じデータ・ハンドラーを使用するように構成する必要があります。</p>
<p>グローバル化</p>	<p>はい</p>
<p>BIDI 対応</p>	<p>いいえ</p>

ログ・ファイルの出力ロケーション

外部サービス・ウィザードで生成されるログ・ファイルの絶対パス名。

表 49. 「ログ・ファイルの出力ロケーション」の詳細

<p>必須</p>	<p>いいえ</p>
<p>デフォルト</p>	<p><code>¥.metadata ¥FlatFileMetadataDiscoveryImpl.log</code></p>
<p>プロパティー・タイプ</p>	<p>String</p>
<p>使用法</p>	
<p>グローバル化</p>	<p>いいえ</p>
<p>BIDI 対応</p>	<p>いいえ</p>

ロギング・レベル

アダプターで使用されるロギングのレベル。

表 50. ログイン・レベルの詳細

必須	いいえ
使用可能な値	重大 警告 監査 情報 構成 詳細
デフォルト	重大
プロパティ・タイプ	値リスト
グローバル化	いいえ
BIDI 対応	いいえ

ネーム・スペース

生成されるビジネス・オブジェクトのネーム・スペース。

表 51. 「ネーム・スペース」の詳細

必須	はい
デフォルト	http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile
プロパティ・タイプ	String
グローバル化	はい
BIDI 対応	いいえ

操作名

このモジュールに定義された操作に指定する名前。

表 52. 操作名の詳細

必須	いいえ
デフォルト	ServiceType プロパティを Outbound に設定すると、操作 Create、Append、Retrieve、Delete、List、Overwrite、および Exists がリストされます。
プロパティ・タイプ	String
グローバル化	いいえ
BIDI 対応	いいえ

処理指示

Inbound または Outbound の処理指示

表 53. 処理指示の詳細

必須	はい
使用可能な値	Outbound Inbound

表 53. 処理指示の詳細 (続き)

デフォルト	Outbound
プロパティ・タイプ	String
グローバル化	いいえ
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、エクスポート用の Inbound イベント処理の構成情報を保持しています。活動化仕様プロパティは、外部サービス・ウィザードまたは管理コンソールのいずれかを使用して設定します。

以下に示す活動化仕様プロパティはバージョン 6.1.0 では必要なくなりましたが、旧バージョンとの互換性のためにサポートされています。

- ArchivingProcessed
- DefaultObjectName
- EventContentType

以下の表は、Inbound 通信の活動化仕様プロパティを示します。外部サービス・ウィザードを使用して活動化仕様プロパティを設定します。これらのプロパティは、WebSphere Integration Developer アセンブリー・エディターを使用してデプロイメント前に変更することも、WebSphere Process Server 管理コンソールを使用してデプロイメント後に変更することもできます。

各プロパティの詳細については、表の後の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、172 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 54. 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	
201 ページの『アーカイブ・ディレクトリー』	ArchiveDirectory	アダプターが処理済みのイベント・ファイルをアーカイブするディレクトリー。
(なし)	ArchivingProcessed	非推奨
201 ページの『自動作成イベント・テーブル』	EP_Create Table	イベント・パーシスタンス・テーブルを自動的に作成するか手動で作成するかを決定する。
202 ページの『イベント・パーシスタンス・プロパティの双方向変換』	EP_BiDiFormat	アダプターがイベント・パーシスタンス・プロパティのいずれかを変換するかどうかを決定する。
(なし)	DefaultObjectName	非推奨
送達タイプ	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 54. 活動化仕様プロパティ (続き)

プロパティ名		説明
ウィザード内	管理コンソール内	
イベントを一度のみ送達する	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。
202 ページの『データベース・スキーマ名』	EP_SchemaName	イベント・パースタンス処理によって使用されるデータベースのスキーマ名。
将来のタイム・スタンプを持つイベントを処理しない	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。
(なし)	EventContentType	非推奨
204 ページの『イベント・ディレクトリー』	EventDirectory	イベント・ファイルが保管されるディレクトリー。
204 ページの『イベント・リカバリー DataSource (JNDI) 名』	EP_DataSource_JNDIName	JDBC データベース接続を取得するためにイベント・パースタンス処理で使用されるデータ・ソースの JNDI 名。データ・ソースを WebSphere Process Server に作成する必要があります。
205 ページの『イベント・リカバリー・テーブル名』	EP_TableName	イベント・パースタンス処理にアダプターが使用するテーブルの名前。
処理するイベント・タイプ	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。
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。
206 ページの『アーカイブ用の障害ファイル拡張子』	FailedArchiveExtension	処理に失敗したビジネス・オブジェクトを入力イベント・ファイルにアーカイブするのに使用するファイル拡張子。
206 ページの『ファイル内容のエンコード』	FileContentEncoding	アダプターによって読み取られるファイルのエンコード方式。
207 ページの『アーカイブ用のファイル拡張子』	OriginalArchiveExtension	オリジナル・イベント・ファイルをアーカイブするのに使用するファイル拡張子。
207 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』	IncludeEndBO Delimiter	SplitCriteria プロパティで指定された区切り文字の値が、さらに処理を行うためにビジネス・オブジェクトの内容と一緒に送信されるかどうかを指定する。

表 54. 活動化仕様プロパティ (続き)

プロパティ名		説明
ウィザード内	管理コンソール内	
ポーリング期間の間隔	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
システム接続を再試行する回数	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。
209 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』	FilePassByReference	アダプターがファイルの内容をエクスポートに送達するかどうかを指定する。
209 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』	EP_Password	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・パーシスタンス処理で使用されるパスワード。
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
209 ページの『イベント・ディレクトリー内のサブディレクトリーのポーリング』	PollSubDirectories	アダプターが、イベント・ディレクトリー内のサブディレクトリーをポーリングするかどうかを指定する。
210 ページの『ソート順でのファイルの取得』	SortEventFiles	ポーリングされたイベント・ファイルのソート順。
210 ページの『パターンを使用したファイルの取得』	EventFileMask	イベント・ファイル用のファイル・フィルター。
開始時の接続の再試行	RetryConnectionOnStartup	アダプターが始動時に ローカル・ファイル・システム 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。
接続が失敗した場合の再試行間隔	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
212 ページの『ファイル内容を分割するための基準の指定』	SplitCriteria	イベント・ファイル内でビジネス・オブジェクトを分離する区切り文字か、またはイベント・ファイルの最大サイズ。「分割機能クラス名 (Splitting Function Class Name)」に設定した値によってどちらか決まります。
213 ページの『関数クラス名の分割』	SplittingFunctionClassName	イベント・ファイルの分割方法 (区切り文字またはサイズ) を指定する。
214 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。

表 54. 活動化仕様プロパティ (続き)

プロパティ名		説明
ウィザード内	管理コンソール内	
214 ページの『アーカイブ用の成功ファイル拡張子』	SuccessArchiveExtension	処理に成功したビジネス・オブジェクトをアーカイブするのに使用するファイル拡張子。
215 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』	EP_UserName	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・パーシスタンス処理で使用されるユーザー名。
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

アーカイブ・ディレクトリー

このプロパティでは、アダプターが処理済みのイベント・ファイルをアーカイブするディレクトリーを指定します。

表 55. 「アーカイブ・ディレクトリー」の詳細

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、アーカイブ・ディレクトリーを示すことができます。環境変数の名前を中括弧で囲み、前に \$ 記号を付けて指定します。例えば、\${ARCHIVE_DIRECTORY} のようになります。この資料の環境変数の作成に関するトピックを参照してください。
グローバル化	はい
BIDI 対応	はい

自動作成イベント・テーブル

このプロパティでは、イベント・パーシスタンス・テーブルを自動的に作成するか手動で作成するかを決定します。

表 56. 「自動作成イベント・テーブル」の詳細

必須	いいえ
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	この値を True に設定すると、アダプターはイベント・パーシスタンス・テーブルを作成します。この値を False に設定した場合、アダプターはテーブルを作成せず、ユーザーが手動でテーブルを作成する必要があります。推奨される設定値は True です。
グローバル化	いいえ

イベント・パーシスタンス・プロパティの双方向変換

このプロパティは、アダプターがイベント・パーシスタンス・プロパティのいずれかを変換するかどうかを決定します。

表 57. イベント・パーシスタンス・プロパティの双方向変換

必須	いいえ
使用可能な値	VRYYN など、ストリング値を指定できます。
デフォルト	なし
プロパティ・タイプ	String
使用法	イベント・パーシスタンス双方向形式プロパティ (EP_BiDiFormat) に設定した値により、双方向変換が決定されます。 VRYYN などのストリング値を指定して、イベント・パーシスタンス・プロパティの双方向変換を有効にできます。EP_BiDiFormat プロパティを指定しないと、アダプターはヌル値を表示します。 注: 双方向変換は、値が双方向コンテキスト・エンタープライズ情報システム (EIS) プロパティに設定されているイベント・プロパティのみ行うことができます。
グローバル化	いいえ
BIDI 対応	はい

データベース・スキーマ名

このプロパティでは、イベント・パーシスタンス処理によって使用されるデータベースのスキーマ名を指定します。

表 58. 「データベース・スキーマ名」の詳細

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	はい
BIDI 対応	はい

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 59. 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

表 59. 送達のタイプの詳細 (続き)

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

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 60. 「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

イベント送達は 1 回のみ (AssuredOnceDelivery)

このプロパティーでは、Inbound イベントに対して、「イベントを一度のみ送達する」の機能を提供するかどうかを指定します。

表 61. 「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
プロパティー・タイプ	Boolean

表 61. 「イベント送達は 1 回のみ」の詳細 (続き)

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にのみ使用されます。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

イベント・ディレクトリー

このプロパティでは、イベント・ファイルが保管されるローカル・ファイル・システムのディレクトリーを指定します。

表 62. 「イベント・ディレクトリー」の詳細

必須	はい
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、イベント・ディレクトリーを示すことができます。環境変数の名前を中括弧で囲み、前に \$ 記号を付けて指定します。例えば、<code>\${EVENT_DIRECTORY}</code> のようになります。この資料の環境変数の作成に関するトピックを参照してください。</p>
グローバル化	はい
BIDI 対応	はい

イベント・リカバリー DataSource (JNDI) 名

このプロパティでは、JDBC データベース接続を取得するためにイベント・パーシスタンス処理で使用されるデータ・ソースの JNDI 名を指定します。

表 63. 「イベント・リカバリー DataSource (JNDI) 名」の詳細

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>データ・ソースを WebSphere Process Server に作成する必要があります。データベースを使用せずにイベント・ポーリングを有効化するには、この値を空のままにします。</p>
グローバル化	はい
BIDI 対応	はい

イベント・リカバリー・テーブル名

このプロパティでは、イベント・パーシスタンス処理にアダプターが使用するテーブルの名前を指定します。

表 64. 「イベント・リカバリー・テーブル名」の詳細

必須	いいえ
デフォルト	なし
プロパティ・タイプ	String
使用法	複数の活動化仕様インスタンスが使用される時、この値は活動化仕様インスタンスごとに固有である必要があります。
グローバル化	はい
BIDI 対応	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 65. 「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

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 66. 「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 66. 「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

アーカイブ用の障害ファイル拡張子

このプロパティでは、処理に失敗したビジネス・オブジェクトを入力イベント・ファイルにアーカイブするのに使用するファイル拡張子を指定します。

表 67. 「アーカイブ用の障害ファイル拡張子」の詳細

必須	いいえ
デフォルト	fail
プロパティ・タイプ	String
グローバル化	はい
BIDI 対応	はい

ファイル内容のエンコード

このプロパティでは、アダプターによって読み取られるファイルのエンコード方式を指定します。

表 68. 「ファイル内容のエンコード」の詳細

必須	いいえ
デフォルト	UTF-8
プロパティ・タイプ	String
使用法	<p>Java でサポートされているエンコード・セット (UTF-8 など) を指定できます。 FileContentEncoding プロパティを指定しない場合、アダプターはデフォルトのシステム・エンコード方式を使用します。</p> <p>アダプターでバイナリー・イベント・データを処理する場合は、このプロパティを BINARY に設定します。アダプターがテキストや XML などのバイナリー以外のイベント・データを処理している場合は、このプロパティを UTF-8 などの有効なファイル・エンコード方式の値に設定します。</p>

表 68. 「ファイル内容のエンコード」の詳細 (続き)

グローバル化	いいえ
BIDI 対応	いいえ

アーカイブ用のファイル拡張子

このプロパティーでは、オリジナル・イベント・ファイルをアーカイブするのに使用するファイル拡張子を指定します。

表 69. 「アーカイブ用のファイル拡張子」の詳細

必須	いいえ
デフォルト	original
プロパティー・タイプ	String
使用法	このプロパティーでは、ビジネス・オブジェクトの処理が失敗した場合に備えて、イベント・ファイル全体を参照用に保持します。
グローバル化	はい
BIDI 対応	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

このプロパティーでは、さらに処理を行うために SplitCriteria プロパティーで指定された区切り文字の値がビジネス・オブジェクトの内容と一緒に送信されるかどうかを指定します。

表 70. 「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」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、さらに処理を行うために SplitCriteria プロパティーで指定された区切り文字の値がビジネス・オブジェクトの内容と一緒に送信される。このプロパティーは、イベント・ファイルの分割が区切り文字に基づく場合、つまり、SplittingFunctionClassName プロパティーが com.ibm.j2ca.extension.utils.filesplit.SplitByDelimiter に設定されている場合のみ有効です。 注: このプロパティーは、コンテンツの終了 BO 区切り文字を処理できるカスタム・データ・バインディングと一緒に使用する必要があります。これを XMLDataHandler と一緒に使用すると、データ・バインディング・レベルで障害が発生します。
グローバル化	いいえ
BIDI 対応	いいえ

ポーリング期間の間隔 (ポーリング間隔)

このプロパティーでは、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 71. 「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
プロパティー・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)

このプロパティーでは、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 72. 「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
プロパティー・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

システム接続を再試行する回数 (RetryLimit)

このプロパティーでは、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 73. 「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
プロパティー・タイプ	Integer

表 73. 「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 ローカル・ファイル・システム 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、ローカル・ファイル・システム 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

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

表 74. 「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」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	このプロパティを True に設定した場合、アダプターはディレクトリー名とファイル名を送信しますが、ファイルの内容は読み込みません。イベント・ファイルにはタイム・スタンプが付加され、アーカイブ・ディレクトリーにアーカイブされます。例えば、a.txt がイベント・ファイルである場合、このファイルは a.txt.yyyy_MM_dd_HH_mm_ss_SSS としてアーカイブ・ディレクトリーにアーカイブされます。 注: このプロパティは、コンテンツが設定されていない場合にランタイムで失敗しないカスタム・データ・バインディングと一緒に使用できますが、パススルー・シナリオでも使用できません。このプロパティを XMLDataHandler と一緒に使用すると、データ・バインディング・レベルで障害が発生します。これは、XMLDataHandler が、ファイル名とディレクトリーのパスに加えてコンテンツがあることを予期するためです。
グローバル化	いいえ

イベント DataSource に接続するのに使用されるパスワード

このプロパティでは、データ・ソースからの JDBC データベース接続を取得するためにイベント・パーシスタンス処理で使用されるパスワードを指定します。

表 75. 「イベント DataSource に接続するのに使用されるパスワード」の詳細

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	はい
BIDI 対応	はい

イベント・ディレクトリー内のサブディレクトリーのポーリング

このプロパティは、アダプターがイベント・ディレクトリー内のサブディレクトリーをポーリングするかどうかを指定します。

表 76. 「イベント・ディレクトリー内のサブディレクトリーのポーリング」の詳細

必須	いいえ
デフォルト	False
プロパティー・タイプ	Boolean
使用法	<p>このプロパティーを True に設定すると、アダプターにより、イベント・ディレクトリー内のファイルと、またそのサブディレクトリー内のファイルもポーリングされます。このプロパティーを False に設定すると、アダプターにより、ルート・ディレクトリー内のファイルのみがポーリングされ、すべてのサブディレクトリーは無視されます。</p> <p>ポーリング周期中は、アダプターにより、まずルート・ディレクトリー内のファイルがポーリングされ、その後にサブディレクトリー内のファイルがポーリングされます。SortEventFiles プロパティーに設定されている値を基準にソートされ、PollQuantity プロパティーに設定されている値に従って処理されます。その後、ビジネス・オブジェクトがダウンストリーム・コンポーネントに送信されます。</p> <p>PollSubDirectories プロパティーが True に設定され、アーカイブが使用可能になっていると、サブディレクトリーからポーリングされたファイルを含め、ポーリングされたすべてのファイルがアーカイブ・ディレクトリーにアーカイブされます。</p>
グローバル化	いいえ
BIDI 対応	いいえ

ソート順でのファイルの取得

このプロパティーでは、ポーリングされたイベント・ファイルのソート順を指定します。

表 77. 「ソート順でのファイルの取得」の詳細

必須	いいえ
使用可能な値	File name - ファイル名で昇順にソートする Time stamp- 最終変更日時時のタイム・スタンプで昇順にソートする No sort- ソートなし
デフォルト	No sort
プロパティー・タイプ	String
使用法	グローバル化をサポートする目的で、ファイル名のソート機能はシステム・ロケールに応じて提供されます。ICU4J パッケージを使用してロケールおよびロケールに対応した規則を追跡します。
グローバル化	いいえ
BIDI 対応	いいえ

パターンを使用したファイルの取得

このプロパティーでは、イベント・ファイル用のファイル・フィルターを指定します。

表 78. 「パターンを使用したファイルの取得」の詳細

必須	はい
----	----

表 78. 「パターンを使用したファイルの取得」の詳細 (続き)

デフォルト	*.*
プロパティ・タイプ	String
使用法	ファイル・フィルタは正しく修飾された有効な正規表現で、英数字とワイルドカード文字「*」を使用できます。*。例えば event* と指定すると、event で始まる名前のファイルのみが処理されます。
グローバル化	はい
BIDI 対応	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 ローカル・ファイル・システム 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 79. 「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
使用法	<p>このプロパティは、アダプターの始動時に、ローカル・ファイル・システム 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、ローカル・ファイル・システム 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 False 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 Stopped 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 True 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、RetryInterval プロパティで再試行の頻度を判別して再接続を試み、RetryLimit 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 Started 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 80. 再試行間隔の詳細

必須	はい
デフォルト	2000

表 80. 再試行間隔の詳細 (続き)

計測単位	ミリ秒
プロパティタイプ	Integer
使用法	正の値のみが有効です。このプロパティでは、アダプターが Inbound 接続に関連したエラーを検出した場合に、アダプターが新規接続を確立しようとするまで待機する時間の長さを指定します。
グローバル化	はい
BIDI 対応	いいえ

ファイル内容を分割するための基準の指定

このプロパティでは、イベント・ファイル内のビジネス・オブジェクトを分離する区切り文字またはイベント・ファイルの最大サイズのいずれかを指定します。

表 81. 「ファイル内容を分割するための基準の指定」の詳細

必須	いいえ
デフォルト	0
プロパティタイプ	String

表 81. 「ファイル内容を分割するための基準の指定」の詳細 (続き)

<p>使用法</p>	<p>このプロパティでは、イベント・ファイル内のビジネス・オブジェクトを分離する区切り文字またはイベント・ファイルの最大サイズを指定します。このプロパティの値は、<code>SplittingFunctionClassName</code> プロパティに設定された値によって決まります。</p> <ul style="list-style-type: none"> • <code>SplittingFunctionClassName</code> プロパティを <code>com.ibm.j2ca.extension.utils.filesplit.SplitByDelimiter</code> に設定した場合、<code>SplitCriteria</code> プロパティにはイベント・ファイル内でビジネス・オブジェクトを分離する区切り文字が含まれている必要があります。 • <code>SplittingFunctionClassName</code> プロパティを <code>com.ibm.j2ca.utils.filesplit.SplitBySize</code> に設定した場合、<code>SplitCriteria</code> プロパティには最大ファイル・サイズをバイト単位で表す有効な数字が含まれている必要があります。イベント・ファイルのサイズがこの値よりも大きい場合、イベント・ファイルはこの値のチャンクに分割され、そのチャンクの数を送られます。イベント・ファイルのサイズがこの値よりも小さい場合、イベント・ファイル全体が送られます。 <p><code>SplitCriteria</code> プロパティの値を 0 に設定すると、ファイル分割が無効になります。 注: <code>Inbound</code> パススルー・シナリオ中に、ファイルの分割がサイズに基づく場合、<code>FilePassByReference</code> プロパティが使用可能な場合、イベント・ファイルはチャンクに分割されません。 注: 複数の <code>COBOL</code> コピーブック・レコードを格納する入力ファイルの場合、サイズ指定でファイルを分割できるようにするには、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、次の方法を使用してください。</p> <ol style="list-style-type: none"> 1. ビジネス・オブジェクトをテキスト・エディターで開きます。 2. 名前属性にビジネス・オブジェクト名の値が指定されている複合タイプのタグを探します。以下の例では、ビジネス・オブジェクト名は <code>DFHCOMMAREA</code> です。 3. <code>aggregateInstanceTD</code> と呼ばれるネーム・スペースが付加されたタグを探し、属性 <code>contentSize</code> の値を使用します。この例では、値は 117 です。これが、タイプ <code>DFHCOMMAREA</code> の各レコードのサイズです。 <pre><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"></pre>
<p>グローバル化</p>	<p>はい</p>
<p>BIDI 対応</p>	<p>はい</p>

関数クラス名の分割

このプロパティでは、イベント・ファイルの分割方法を決定します。

表 82. 「関数クラス名の分割」の詳細

<p>必須</p>	<p>いいえ</p>
<p>使用可能な値</p>	<p><code>com.ibm.j2ca.extension.utils.filesplit.SplitByDelimiter</code> ファイルは、イベント・ファイル内のビジネス・オブジェクトを分離する区切り文字に基づいて分割されます。 <code>com.ibm.j2ca.utils.filesplit.SplitBySize</code> ファイルは、イベント・ファイルのサイズに基づいて分割されます。</p>

表 82. 「関数クラス名の分割」の詳細 (続き)

デフォルト	com.ibm.j2ca.utils.filesplit.SplitBySize
プロパティ・タイプ	String
使用法	区切り文字またはファイル・サイズは、SplitCriteria プロパティで設定します。 注: EventContentType プロパティが NULL の場合、SplittingFunctionClassName プロパティは com.ibm.j2ca.utils.filesplit.SplitBySize に自動的に設定されます。
グローバル化	いいえ
BIDI 対応	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 83. 「ポーリング時にエラーが検出された場合はアダプターを停止する」の詳細

必須	いいえ
使用可能な値	True False
デフォルト	False
プロパティ・タイプ	Boolean
使用法	このプロパティを True に設定した場合、アダプターはエラーを検出するとポーリングを停止します。 このプロパティを False に設定した場合、アダプターはポーリング時にエラーを検出すると例外をログに記録し、ポーリングを続行します。
グローバル化	いいえ
BIDI 対応	いいえ

アーカイブ用の成功ファイル拡張子

このプロパティでは、処理に成功したビジネス・オブジェクトをアーカイブするのに使用するファイル拡張子を指定します。

表 84. 「アーカイブ用の成功ファイル拡張子」の詳細

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	はい
BIDI 対応	はい

イベント DataSource に接続するのに使用されるユーザー名

このプロパティーでは、データ・ソースからの JDBC データベース接続を取得するためにイベント・パーシスタンス処理で使用されるユーザー名を指定します。

表 85. 「イベント DataSource に接続するのに使用されるユーザー名」の詳細

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	はい
BIDI 対応	はい

ルール・ベースのイベント・フィルター処理 (ruleTable)

このプロパティーは、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 86. 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
プロパティー・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
BIDI 対応	いいえ

関連概念

28 ページの『WebSphere Application Server 環境変数』

外部サービス・ウィザードで WebSphere Application Server 環境変数を使用して、ディレクトリー値を指定できます。

55 ページの『必須のローカル・フォルダーの作成』

Inbound モジュールまたは Outbound モジュールを作成する前に、イベントおよび出力用のフォルダーをローカル・ファイル・システムに作成する必要があります。またオプションで、ステージングおよびアーカイブ用のフォルダーを作成することができます。

関連タスク

58 ページの『WebSphere Application Server 環境変数の定義』

WebSphere Process Server または WebSphere Enterprise Service Bus 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。

105 ページの『デプロイメントおよびランタイム・プロパティーの設定』

エンタープライズ情報システム (ローカル・ファイル・システム) とのOutbound または Inbound 通信でご使用のモジュールを使用することを決定した後、エクスポート用の Inbound イベント処理構成情報を保持する活動化仕様プロパティーを構成する必要があります。

リソース・アダプター・プロパティー

リソース・アダプター・プロパティーでは、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・アダプターのプロパティーは、アダプターの構成時に外部サービス・ウィザードを使用して設定します。アダプターをデプロイしたあとは、管理コンソールを使用してこれらのプロパティーを変更します。

以下に示すロギングおよびトレースのプロパティーは、バージョン 6.1.0 では必要なくなりました。ただし、旧バージョンとの互換性を維持するため、管理コンソールで表示することは可能です。

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

以下の表に、リソース・アダプター・プロパティーとその目的のリストを示します。各プロパティーの完全な説明は、表に続くセクションで説明します。後続セクションのプロパティー詳細表の見方について詳しくは、172 ページの『プロパティーの詳細についてのガイド』を参照してください。

表 87. Adapter for Flat Files 用のリソース・アダプター・プロパティー

名前		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。
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	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)	HA サポートの使用可能化	このプロパティーは変更しないでください。
(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
(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 88. 「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
使用法	<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、FFRA 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、FFRA001 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 9 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 Flat Files 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、FFRA001 および FFRA002 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 FFRAInstance0 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、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>
グローバル化	はい
BIDI 対応	いいえ

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 89. 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
----	-----

表 89. 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せん。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

高可用性サポートを使用可能にする (Enable high availability support) (enableHASupport)

このプロパティは変更しないでください。true に設定してください。

グローバリゼーション

WebSphere Adapter for Flat Files 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。

グローバリゼーションおよび双方向データ変換

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。アダプターは双方向のスクリプト・データ変換も実行します。双方向変換とは、1 つのファイルに右から左 (ヘブライ語やアラビア語など) と左から右 (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のことを指します。

グローバリゼーション

グローバル化されたソフトウェア・アプリケーションは、単一環境ではなく複数の言語環境や国/地域別環境を使用することを目的として設計され、開発されています。WebSphere Adapters、WebSphere Integration Developer、および WebSphere Process Server または WebSphere Enterprise Service Bus は、Java で作成されています。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。そのため、これらの統合システム・コンポーネント間でデータを転送するときに文字を変換する必要はありません。

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。

双方向スクリプト・データ変換

アラビア語やヘブライ語などの言語は右から左に書きますが、テキストには左から右に書かれる部分も埋め込まれるため、双方向スクリプトになります。ソフトウェア・アプリケーションで双方向スクリプト・データを扱う場合は、その表示と処理のためにさまざまな規格を使用します。双方向スクリプト・データ変換の適用対象は、ストリング型のデータのみです。WebSphere Process Server または WebSphere Enterprise Service Bus は Windows 標準形式を使用しますが、サーバーとの間でデータをやり取りするアプリケーションやファイル・システムは別の形式を使用する可能性があります。アダプターは、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。スクリプト・データの変換は、スクリプト・データの形式を定義する 1 組のプロパティーと、変換の適用先となるコンテンツまたはメタデータを指定するプロパティーを使用して行われます。

双方向スクリプト・データ形式

WebSphere Process Server または 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 の使用する形式です。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。双方向プロパティーを設定する場合、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 90. 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	

表 90. 双方向形式の属性 (続き)

文字の位置	目的	値	説明	デフォルト設定
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)	

変換するデータを指定する双方向プロパティ

変換するビジネス・データを指定するには、BiDiContextEIS プロパティを設定します。この操作を行うには、このプロパティの 5 つの双方向形式属性 (前の表に記載) のそれぞれに対して値を指定します。BiDiContextEIS プロパティは、Managed Connection Factory および活動化仕様に対して設定できます。

変換するイベント・パーシスタンス・データを指定するには、EP_BiDiFormat プロパティを設定します。EP_BiDiFormat プロパティの値は、BiDiContextEIS プロパティに指定した値を使用して設定します。EP_BiDiFormat プロパティを、活動化仕様に設定できます。

変換の対象となるアプリケーション固有のデータを指定するには、ビジネス・オブジェクト内部の BiDiContextEIS プロパティおよび BiDiMetadata プロパティに注釈を付けます。この操作を行うには、WebSphere Integration Developer 内部のビジネス・オブジェクト・エディターを使用して、ビジネス・オブジェクトのアプリケーション固有の要素としてプロパティを追加します。

関連資料

198 ページの『活動化仕様プロパティ』

活動化仕様プロパティは、エクスポート用の Inbound イベント処理の構成情報を保持しています。活動化仕様プロパティは、外部サービス・ウィザードまたは管理コンソールのいずれかを使用して設定します。

177 ページの『Managed Connection Factory プロパティ』

Managed Connection Factory プロパティでは、アダプターがローカル・ファイル・システムとの Outbound 通信の実行時に必要とする情報を指定します。

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 Flat Files 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、FlatFile.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、Flat Files ビジネス・オブジェクトのすべての属性に適用できます。FlatFileBaseDataBinding 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 (FlatFile) 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・システムと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・データの正しい形式を強制するものです。これらのプロパティが設定されると、双方向スクリプト・データは、WebSphere Integration Developer および WebSphere Process Server または WebSphere Enterprise Service Bus で正しく処理および表示されます。

Managed Connection Factory プロパティ

以下の Managed Connection Factory プロパティは双方向スクリプト・データ変換を制御します。

- FileSequenceLog
- OutputDirectory
- OutputFilename
- StagingDirectory

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スクリプト・データ変換を制御します。

- ArchiveDirectory
- EventDirectory
- EventFileMask
- FailedArchiveExtension
- OriginalArchiveExtension
- SplitCriteria
- SuccessArchiveExtension

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スクリプト・データ変換を制御します。

- EPDatabasePassword
- EPDatabaseSchemaName
- EPDatabaseUsername
- EPDataSourceJNDIName

- EPEventTableName

ビジネス・オブジェクト・ラッパー・プロパティ

以下のビジネス・オブジェクト・ラッパー・プロパティは双方向スクリプト・データ変換を制御します。

- DirectoryPath
- FileName
- IncludeEndBODElimiter
- StagingDirectory
- ArchiveDirectoryForDeleteOnRetrieve
- ChunkFileName

アダプター・メッセージ

以下のロケーションで WebSphere Adapter for Flat Files により発行されたメッセージを表示します。

メッセージへのリンク: <http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.wbit.620.help.messages.doc/messages.html>

表示される Web ページには、メッセージ接頭語のリストが示されています。メッセージ接頭語をクリックすると、その接頭語を持つすべてのメッセージを表示できます。

- 接頭語 CWYFF を持つメッセージは、WebSphere Adapter for Flat Files により発行されます。
- 接頭部が CWYBSのメッセージは、すべてのアダプターが使用するアダプター・ファウンデーション・クラスにより発行されます。

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 Flat Files の関連情報が含まれています。

サンプルとチュートリアル

ユーザーが、WebSphere Adapters を円滑に使用できるようにするため、「Business Process Management Samples and Tutorials」Web サイトからサンプルおよびチュートリアルが利用できます。サンプルおよびチュートリアルには、以下のいずれかの方法でアクセスできます。

- WebSphere Integration Developer を始動したときに開くウェルカム・ページから。WebSphere Adapter for Flat Files のサンプルとチュートリアルを表示するには、「取得 (Retrieve)」をクリックします。次に、表示されたカテゴリーを参照して選択します。
- Web 上のロケーション <http://publib.boulder.ibm.com/bpcsamp/index.html> から。

情報リソース

- WebSphere Business Process Management 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、WebSphere Adapters にはついて学習するのに役立つ項目、Redbooks、文書、および教育関連のオフリングへのリンクが含まれています。
- WebSphere Adapters ライブラリー・ページ <http://www.ibm.com/software/integration/wbiadapters/library/infocenter/> には、すべてのバージョンの文書へのリンクが含まれています。

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 <http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/index.jsp>。このインフォメーション・センターには、WebSphere Process Server、WebSphere Enterprise Service Bus、および WebSphere Integration Developer 情報が含まれます。
- 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
- WebSphere Business Integration ゾーン (WebSphere business integration zone)

サポートおよび支援

- WebSphere Adapters テクニカル・サポート <http://www.ibm.com/software/integration/wbiadapters/support/>
- WebSphere Adapters 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、**検索ボタン**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 に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテス

トを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生した創作物にも、次のように、著作権表示を入れていただく必要があります。「(c) (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。 (c) Copyright IBM Corp. _年を入れる_. All rights reserved.

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

プログラミング・インターフェース情報

プログラミング・インターフェース情報は、プログラムを使用してアプリケーション・ソフトウェアを作成する際に役立ちます。

一般使用プログラミング・インターフェースにより、お客様はこのプログラム・ツール・サービスを含むアプリケーション・ソフトウェアを書くことができます。

ただし、この情報には、診断、修正、および調整情報が含まれている場合があります。診断、修正、調整情報は、お客様のアプリケーション・ソフトウェアのデバッグ支援のために提供されています。

警告:

診断、修正、調整情報は、変更される場合がありますので、プログラミング・インターフェースとしては使用しないでください。

商標

IBM、IBM ロゴ、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]

アダプターのテクニカル・ノート 224
アダプターのパフォーマンス 148
アダプター・アプリケーション
開始 147
停止 148
アダプター・パターン・ウィザード 82
アダプター・メッセージ 223
イベント・アーカイブ値 21
イベント・ストア
概要 19
構造 20
インストール、EAR ファイルの 132
エクスポート、モジュールの EAR ファイルとしての 131

[カ行]

開始、アダプター・アプリケーションの 147
外部サービス
概要 29
Inbound 成果物の生成 120
外部サービス 接続プロパティ
関数セレクター 173, 194
処理指示 173, 194
操作名 173, 194
データ・バインディング 173, 194
ネーム・スペース 173, 194
ロギング・レベル 173, 194
ログ・ファイルの出力ロケーション 173, 194
BiDi フォーマット・ストリング 173, 194
外部サービス・ウィザード
開始 87
外部サービス・ディスクバリアー、接続プロパティ 90, 105
カスタム・プロパティ
活動化仕様 140, 146
リソース・アダプター 135, 142
Managed Connection Factory 138, 144
活動化仕様プロパティ
アーカイブ用の障害ファイル拡張子 198
アーカイブ用の成功ファイル拡張子 198
アーカイブ用のファイル拡張子 198
アーカイブ・ディレクトリー 198
イベント DataSource に接続するのに使用されるパスワード
198

活動化仕様プロパティ (続き)

イベント DataSource に接続するのに使用されるユーザー名
198
イベントを一度のみ送達する 198
イベント・ディレクトリー 198
イベント・ディレクトリー内のサブディレクトリーのポー
リング 198
イベント・リカバリー DataSource (JNDI) 名 198
イベント・リカバリー・テーブル名 198
開始時の接続の再試行 198
関数クラス名の分割 198
管理コンソールでの設定 140, 146
システム接続を再試行する回数 198
自動作成イベント・テーブル 198
将来のタイム・スタンプを持つイベントを処理しない 198
処理するイベント・タイプ 198
接続が失敗した場合の再試行間隔 198
ソート順でのファイルの取得 198
送達タイプ 198
データベース・スキーマ名 198
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 198
パターンを使用したファイルの取得 198
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 198
ファイル内容のエンコード 198
ファイル内容を分割するための基準の指定 198
ポーリング期間の間隔 198
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 198
ポーリング数量 198
関数セレクター 22
管理接続プロパティ
シーケンス・ファイル 177
出力ディレクトリー 177
ステー징・ディレクトリー 177
デフォルト・ターゲット・ファイル名 177
関連情報 223
関連製品、情報 223
技術概要 2
機密データ、隠蔽 31
機密トレース 31
教育、WebSphere Adapters 223
組み込みアダプター
活動化仕様プロパティ、設定 140
考慮事項、使用する際の 34
説明 32
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 138
クラスター環境
説明 35

クラスター環境 (続き)

- デプロイ 35
- Inbound 処理 36
- Outbound 処理 36

高可用性環境

- 説明 35
- デプロイ 35
- Inbound 処理 36
- Outbound 処理 36

構成

- トレース 156
- ロギング 156
- Performance Monitoring Infrastructure (PMI) 149

後方互換性

- プロジェクト 41
- プロジェクト交換ファイル 41

互換性マトリックス 2

固有ファイル名、生成 13

[サ行]

再試行制限プロパティ 208

サポート

- 概要 155
- セルフ・ヘルプ・リソース 162
- テクニカル 224

サポートされる操作 4, 5, 6, 7

サンプル 51

実装、Java 126

スタンドアロン・アダプター

- 活動化仕様プロパティ、設定 146
- 考慮事項、使用する際の 34
- 説明 32
- リソース・アダプター・プロパティ、設定 142
- Managed Connection Factory プロパティ、設定 144

成果物、生成 102

成果物の生成 102

セキュリティー 31

- 重要データの隠蔽 31

接続、コンポーネントの 125

接続プロパティ、Inbound 105

接続プロパティ、Outbound 90

セルフ・ヘルプ・リソース 162

操作 4, 5, 6, 7

ソフトウェア要件 2

[タ行]

ターゲット・コンポーネント 125

対話仕様プロパティ

- 関数クラス名の分割 186
- 固有ファイルの生成 186
- 出力ディレクトリー 186
- ステージング・ディレクトリー 186
- デフォルト・ターゲット・ファイル名 186

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 186
- ファイル内のビジネス・オブジェクト間の区切り文字 186
- ファイル内容のエンコード 186
- ファイル内容を分割するための基準の指定 186
- 変更 123
- retrieve 操作後のファイルの削除 186
- retrieve 操作のアーカイブ・ディレクトリー 186

チュートリアル 51

データ変換 (Inbound) 26

データ変換 (Outbound) 8

データ・バインディングの構成、Inbound 114

データ・バインディングの構成、Outbound 95

停止、アダプター・アプリケーションの 148

テクニカル・サポート 224

テクニカル・ノート 2, 162, 224

テクニカル・ノート、WebSphere Adapters 223

テスト環境

- 追加、モジュールの 127
- テスト、モジュールの 128
- デプロイ 125, 127

デバッグ

- セルフ・ヘルプ・リソース 162
- org.xml.sax.SAXParseException 例外 162
- XAResourceNotAvailableException 例外 161

デプロイメント

- オプション 32
- 環境 125
- 実稼働環境への 129
- テスト環境への 125

トラブルシューティング

- 概要 155
- セルフ・ヘルプ・リソース 162
- org.xml.sax.SAXParseException 例外 162
- XAResourceNotAvailableException 例外 161

トレース

- 管理コンソールによるプロパティの構成 156

トレース・ファイル

- 使用可能化 156
- 詳細レベル 156
- 使用不可化 156
- ファイル名の変更 158
- ロケーション 158

[ハ行]

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

パターン 82

パッケージ・ファイル、アダプターの 157

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 148

ビジネス・インテグレーション・アダプターから JCA 準拠アダプター 42

ビジネス・オブジェクト 4, 27

- 構造 165

- ビジネス・オブジェクト (続き)
 - 属性プロパティ 168
 - 命名規則 169
- ビジネス・オブジェクト、事前定義 56, 62
- ビジネス・オブジェクト、COBOL コピーブック・ファイルからの変換 73
- 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 64
- ビジネス・オブジェクト情報 165
- ビジネス・オブジェクトの命名規則 169
- ビジネス・フォールト 160
- 必要なローカル・フォルダー 55
- ファイル
 - SystemOut.log ログ・ファイル 158
 - trace.log トレース・ファイル 158
- ファイル分割
 - 区切り文字ベース 11, 23
 - サイズ・ベース 11, 23
- フォールト
 - 説明 160
- プロジェクト、作成 87
- プロジェクト交換 (PI) ファイル
 - 更新、マイグレーションせずに 41
- プロパティ
 - 活動化仕様 140, 146
 - 構成プロパティ
 - Inbound 192
 - Outbound 172
 - リソース・アダプター 135, 142
 - Inbound 構成 192
 - Managed (J2C) Connection Factory 138, 144
 - Outbound 構成 172

[マ行]

- マイグレーション 42
 - WebSphere InterChange Server マイグレーション・ウィザード 45
- マイグレーションに関する考慮事項 37
- マイグレーションの概要
 - WebSphere InterChange Server アプリケーション 43
 - マイグレーションのロードマップ
 - WebSphere InterChange Server アプリケーション 42
- マトリックス、互換性 2
- メッセージ、アダプター 223
- モジュール、作成 56
- モジュールの構成のためのロードマップ 53
- 問題判別
 - セルフ・ヘルプ・リソース 162
 - org.xml.sax.SAXParseException 例外 162
 - XAResourceNotAvailableException 例外 161

[ヤ行]

- 要件、ハードウェアおよびソフトウェア 2

[ラ行]

- ランタイム環境
 - デプロイ、EAR ファイルの 129
- リソース・アダプター・アーカイブ (RAR) ファイル
 - インストール、サーバーへの 129
 - 説明 129
- リソース・アダプター・プロパティ
 - アダプター ID 183, 216
 - 管理コンソールでの設定 135, 142
 - 詳細 183, 216
 - HA サポートの使用可能化 183, 216
- 例外
 - org.xml.sax.SAXParseException 162
 - XAResourceNotAvailableException 161
- ロギング
 - 管理コンソールによるプロパティの構成 156
- ログおよびトレース・ファイル 155
- ログ・アナライザー 156
- ログ・ファイル
 - 使用可能化 156
 - 詳細レベル 156
 - 使用不可化 156
 - ファイル名の変更 158
 - ロケーション 158

A

- Adapter for Flat Files
 - 管理 135
- Adapter for Flat Files モジュール
 - インストール、EAR ファイルのサーバーへの 132
 - エクスポート、EAR ファイルとしての 131
 - 開始 147
 - 停止 148
- Append 4

C

- CEI (Common Event Infrastructure) 153
- 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 64
- 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 73
- Common Event Infrastructure (CEI) 153
- Create 4, 5

D

- Delete 4, 6
- developerWorks 224
- developerWorks リソース、WebSphere Adapters 223

E

EAR ファイル
インストール、サーバーへの 132
エクスポート 131
EmbeddedNameFunctionSelector 22
enableHASupport プロパティ 36
Exists 4, 6

F

FFDC (First Failure Data Capture) 159
FilenameFunctionSelector 22
First Failure Data Capture (FFDC) 159

I

IBM WebSphere Adapter Toolkit 224
Inbound 構成プロパティ 192

J

Java 実装 126

L

List 4, 6
Log and Trace Analyzer、サポート 155

M

Managed (J2C) Connection Factory プロパティ
管理コンソールでの設定 138, 144

O

org.xml.sax.SAXParseException 162
Outbound 4, 5, 6, 7
サポートされる操作 4
処理 3
Outbound 構成プロパティ 172
Outbound 操作
Append 4
Create 4
Delete 4
Exists 4
List 4
Overwrite 4
Retrieve 4
Overwrite 4, 6

P

Performance Monitoring Infrastructure (PMI)
構成 149
説明 148
パフォーマンスに関する統計の表示 151
PMI (Performance Monitoring Infrastructure)
構成 149
説明 148
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
インストール、サーバーへの 129
説明 129
Redbooks、WebSphere Adapters 223
Retrieve 4, 7

S

SystemOut.log ファイル 158

T

trace.log ファイル 158

W

WebSphere Adapter for Flat Files 177, 183, 216
アダプター実装の計画 31
概要 1
技術概要 2
セキュリティ 31
Inbound 処理 15
Outbound 処理 3
WebSphere Adapters、バージョン 6.0、情報 224
WebSphere Adapters、バージョン 6.0.2.x、情報 224
WebSphere Application Server 環境変数 28, 55, 177, 198
WebSphere Application Server 環境変数、定義 58
WebSphere Application Server の情報 224
WebSphere Business Integration Adapters の情報 224
WebSphere Business Process Management、バージョン 6.1.x、情報 224
WebSphere Enterprise Service Bus
情報 224
WebSphere Extended Deployment 35
WebSphere Integration Developer
開始 56, 62, 87
情報 224
テスト環境 125
WebSphere Process Server
情報 224

WebSphere Process Server または WebSphere Enterprise Service
Bus
デプロイ 129
WebSphere ビジネス・インテグレーション・アダプター 42

X

XAResourceNotAvailableException 161



Printed in Japan