

IBM Integration Designer
バージョン 8 リリース 5

**WebSphere Adapter for Flat
Files を使用した読み取りと書
き込み**

IBM

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、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 Integration Designer のバージョン 8 リリース 0 に適用されます。

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： Integration Designer
Version 8 Release 5
Reading and writing files using the
WebSphere Adapter for Flat Files

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

© Copyright IBM Corporation 2007, 2012.

目次

第 1 章 概要	1	第 3 章 サンプルの実行	21
概説	1	アダプターのインポート	21
第 2 章 ユーザーによるビルド	7	モジュールのデプロイ	22
ディレクトリーおよび入力ファイルの作成	7	モジュールのテスト	22
モジュールの作成	8	第 4 章 インポート	23
顧客ビジネス・オブジェクトの作成	8	特記事項	25
インバウンド・サービスの作成	9	ご利用条件	29
アウトバウンド・サービスの作成	12		
メディエーション・フローの作成	17		

第 1 章 概要

Advanced

このサンプルでは、IBM WebSphere Adapter for Flat Files を使用して、アプリケーションでファイル・システム上のファイルの読み取りと書き込みを行う方法の実例を示します。また、着信データを処理し、処理後のデータをファイル・システム上のファイルに書き込むアプリケーションを IBM® Integration Designer で作成する方法についても説明します。このサンプルは、アダプター全般、特にフラット・ファイル・アダプターを使用して作業する開発者向けに書かれています。

学習目標

このサンプルを完成させると、IBM Integration Designer を WebSphere Adapter for Flat Files と共に使用して、ファイル・システム上のファイルからデータを読み取り、そのデータをサービス指向アーキテクチャー (SCA) アプリケーションで処理し、処理済みのデータをファイル・システム上のファイルに書き込むアプリケーションを開発する方法を理解できます。具体的には、以下のタスクの実行方法を学習します。

- フラット・ファイル・アダプターで外部サービス・パターン・ウィザードを使用して、ファイル・システム上のファイルからデータを読み取るインバウンド・サービスを開発する
- 同じウィザードおよびアダプターを使用して、ファイル・システム上のファイルにデータを書き込むアウトバウンド・サービスを開発する
- ファイル・システム上のファイルから読み取ったデータを処理するビジネス・ロジックを実装する
- アプリケーションをテストする

このサンプルは、IBM Process Server で使用できます。

このアプリケーションをユーザーがビルドしてテストするために必要な時間は、約 30 分です。既にビルド済みのアプリケーションをインポートしてテストすることもでき、その場合に要する時間は 10 分です。

概説

Advanced

このチュートリアルを使用して作業するときは、フラット・ファイルやアダプターのほか、IBM Integration Designer で作成されたアプリケーションとの間でアダプターを使用して行うデータ処理に関連して、いくつかの概念を理解しておく必要があります。また、作成するアプリケーションについての高度な理解も必要です。

このセクションでは、以下の概念について説明します。

- フラット・ファイル
- WebSphere® Adapter for Flat Files
- インバウンド処理
- ファイル分割
- レコード区切り文字
- サイズによる分割

- カスタム・データ・バインディング
- アウトバウンド処理
- このチュートリアルでのアプリケーションの概要

フラット・ファイルとは

「フラット・ファイル」という用語は、ローカル・ファイル・システムに保管されているすべてのファイルを指し、構造化データベース内のファイルなどのより複雑なファイル・セットと対立する用語です。フラット・ファイル内のデータ・レコードは順次に保管されるのが一般的で、データベース・ストレージに見られるような索引、キー、リレーションシップなどのメタデータを伴っていません。Adapter for Flat Files を使用すると、すべてのタイプのファイルを読み書きできます。2 つの一般的なデータ・フォーマットは、XML レコードとコンマ区切り値 (CSV) レコードです。次の例は、各フォーマットのレコードを示しています。

XML フォーマット

```
<?xml version="1.0" encoding="UTF-8"?>
<customer>
  <title>Mr</title>
  <name>Smith</name>
  <city>Ottawa</city>
  <state>ON</state>
</customer>
<customer>
  <title>Mrs</title>
  <name>Jones</name>
  <city>Winnipeg</city>
  <state>MB</state>
</customer>
```

CSV フォーマット

```
Mr,Smith,Ottawa,ON
Mrs,Jones,Winnipeg,MB
```

WebSphere Adapter for Flat Files とは

あるディレクトリーにファイル・セットを出力する外部システムがあるとします。例えば、注文情報を格納したテキスト・ファイルを生成する注文処理システムなどが考えられます。それらの情報はオフピーク時間にバッチ・モードで処理されます。この場合は、WebSphere Adapter for Flat Files を IBM Integration Designer と一緒に使用して、それらのファイルの作成と管理に役立てます。

ご使用のアプリケーションは、Adapter for Flat Files を使用してファイルの作成と管理を行うか、ディレクトリーをモニターし、そのディレクトリーからファイルを読み取ることができます。Adapter for Flat Files に関連したサービスを作成するときは、次の 2 つのオプションがあり、どちらのオプションも製品のインフォメーション・センターに文書化されています。

- シンプル外部サービス・ウィザード: このウィザードを使用して、ユーザーは WebSphere Adapter for Flat Files が読み書きを行うディレクトリーとそれらのディレクトリーのデータ・フォーマットを指定し、サービスを作成します。その後、このウィザードによって、そのサービスが生成されます。多くのユーザーにとって、サービスを作成するには、この迅速で簡単な方法で十分です。このチュートリアルでは、この方法を使用します。
- 拡張サービス・ウィザード: このウィザードを使用して、ユーザーはより長時間の包括的なプロセスによるサービスを作成します。ユーザーによっては、このさらに詳細な方法の方が適している場合があります。最初に、必要となるアダプターのタイプを指定し、次に WebSphere Adapter for Flat Files が読み書

きを行うディレクトリーと、それらのディレクトリーのデータ・フォーマットを指定します。その後、拡張セクションのフィールドがある後続の各ページを使用して、生成されるサービスをさらに細かく制御します。

インバウンドおよびアウトバウンド処理とは

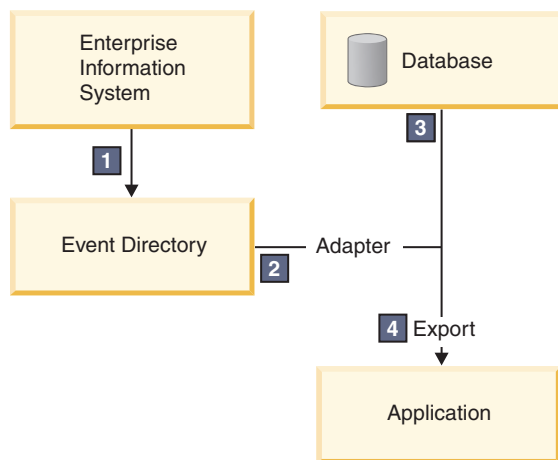
インバウンド処理は WebSphere Adapter for Flat Files がファイル・システムをモニターし、新規ファイルを読み取り、アプリケーション内の操作へデータを送信する操作モードです。

アウトバウンド処理は、WebSphere Adapter for Flat Files が (アプリケーションのコンポーネントから) 要求を受信し、ファイル操作を実行し、該当する場合は結果を呼び出し元に返す操作モードです。操作の例としては、ファイルの作成、ファイルへの書き込み、特定のファイルが存在するかどうかの検査が含まれます。

インバウンド処理

インバウンド処理のとき、Adapter for Flat Files はイベント・ディレクトリーによって生成されるイベント (例えば、イベント・ディレクトリーにファイルが置かれます) を listen します。イベントとは、イベント・ディレクトリー内でどのような変更が発生したかを記録したレコードのことです。Adapter for Flat Files が新規ファイルの有無をモニターするディレクトリーは、イベント・ディレクトリーと呼ばれます。

次の図は、インバウンド処理のときに発生する 4 つのステップを示しています。



1. 外部システムが、ファイルをイベント・ディレクトリーへ出力します。
2. Adapter for Flat Files がイベント・ディレクトリーにファイルがあるかどうかポーリングを行い、それらのファイルからのデータをイベントに変換します。特定のタイプのファイルの有無、または特定の期間に作成されたファイルの有無についてポーリングを行うよう、アダプターを構成できます。
3. イベントは、一時的にイベント・ストアに置かれます。これはデータベースであるか、イベント・テーブルのメモリー内表現です。データベースを使用した場合、仮にイベントが作成された後にサーバーがダウンしても、イベントがアプリケーションによる処理の前に失われることはありません。これは、保証付き配信と呼ばれます。メモリー内テーブルを使用するとイベントの処理が速くなりますが、イベント・リカバリー機能は失われます。
4. Adapter for Flat Files はイベント・ストアからイベントを取得し、外部サービス・ウィザードによって作成されてエクスポートされたインバウンド・インターフェースを介して、各イベントをビジネス・オブジェクトの形で受け渡します。

このデータ取得方式 (つまり、データをビジネス・オブジェクトの形で受け渡す方式) は、非パススルー、または *Data Transformation Framework (DTF)* と呼ばれ、構造化データに対して機能します。

アプリケーションがデータ・ファイルのフォーマットを認識しない場合は、*Adapter for Flat Files* をパススルー・モードで稼働するよう構成し、非構造化データに対して機能させることができます。その場合、イベントはビジネス・オブジェクトに変換されません。

ファイル分割

ファイル分割機能は、取得したいファイルが大きい場合、またはそれぞれのファイルに複数のレコードが入っている場合に使用します。区切り文字または固定サイズ値に基づいて、ファイルをより小さなチャンクに分割でき、ファイルの各パーツを並列で処理できます。各チャンクは別々のイベントと見なされ、エクスポートされたインバウンド・インターフェースを通じてアプリケーション内の操作へ個別に送信されます。

レコード区切り文字

一般に、入力はファイルごとに単一のレコードとして保管されます。しかし、入力ファイルに複数のレコードが含まれているときは、多くの場合、区切り文字によってファイル内のレコードが分離されます。区切り文字は任意のテキスト・ストリングで、通常は文字の組み合わせの後に `¥r¥n` (プラットフォームに依存する改行文字) を付けます。アダプターは、区切り文字を含んでいるファイルの読み取りと書き込みの両方ができます。

前述した XML フォーマットおよび CSV フォーマットの例を使用すると、次の例では、##### の後に改行を付けたストリングを XML ファイル内で区切り文字として使用しています。CSV ファイルでは、改行を区切り文字としています。

を区切り文字とした XML フォーマット

```
<?xml version="1.0" encoding="UTF-8"?>
<customer>
  <title>Mr</title>
  <name>Smith</name>
  <city>Ottawa</city>
  <state>ON</state>
</customer>
#####
<customer>
  <title>Mrs</title>
  <name>Jones</name>
  <city>Winnipeg</city>
  <state>MB</state>
</customer>
#####
```

改行を区切り文字とした CSV フォーマット

```
Mr,Smith,Ottawa,ON
Mrs,Jones,Winnipeg,MB
```

サイズによる分割

サイズによる分割フィーチャーは、ファイルをより小さなチャンクに分割し、アプリケーション内の操作へ 1 つずつ転送するために使用される点で、区切り文字による分割によく似ています。このフィーチャーはパススルー・シナリオで非構造化データに使用します。この非構造化データはビジネス・オブジェクトに変換されないからです。アダプターでの分割条件は、バイト数として指定します。アダプターはファイルをそのバイト・サイズのチャンク (イベント) として読み取ります。各チャンクは分割条件で定義されたサイズになりますが、最後のチャンクだけは例外で、それより小さくなる場合があります。

カスタム・データ・バインディング

フラット・ファイル内のデータは、前述した XML フォーマットや CSV フォーマットなど、さまざまなフォーマットになる可能性があります。それ以外のフォーマットとしては、名前と値のペア、タブ区切り、固定幅などのフォーマットがあります。データ・ハンドラー は、データ・ファイル内のフォーマットからビジネス・オブジェクトの属性へのマッピングを行います。

Adapter for Flat Files のデフォルトのデータ・バインディングを使用すると、ファイルを XML フォーマットとの間で変換することができます。それ以外のフォーマットでは、ファイル・フォーマットとビジネス・オブジェクトの間のマッピングを定義したカスタム・データ・ハンドラーを使用する必要があります。カスタム・データ・ハンドラーを実装するには、特定のフォーマットのデータを変換する Java™ クラスを作成し、アダプターを構成するときに、そのクラスを外部サービス・ウィザードに入力します。

カスタム・データ・ハンドラーについては、IBM Integration Designer の資料を参照してください。

アウトバウンド処理

ファイルの書き込みまたは変更を行うために、アプリケーションはアウトバウンド・インターフェースで定義された操作を使用します。アウトバウンド・インターフェースは、外部サービス・ウィザードを使用して作成します。以下のさまざまな操作タイプを使用して、非構造化データ (パススルー・モード) と構造化データ (非パススルー・モード) のどちらでも作成できます。

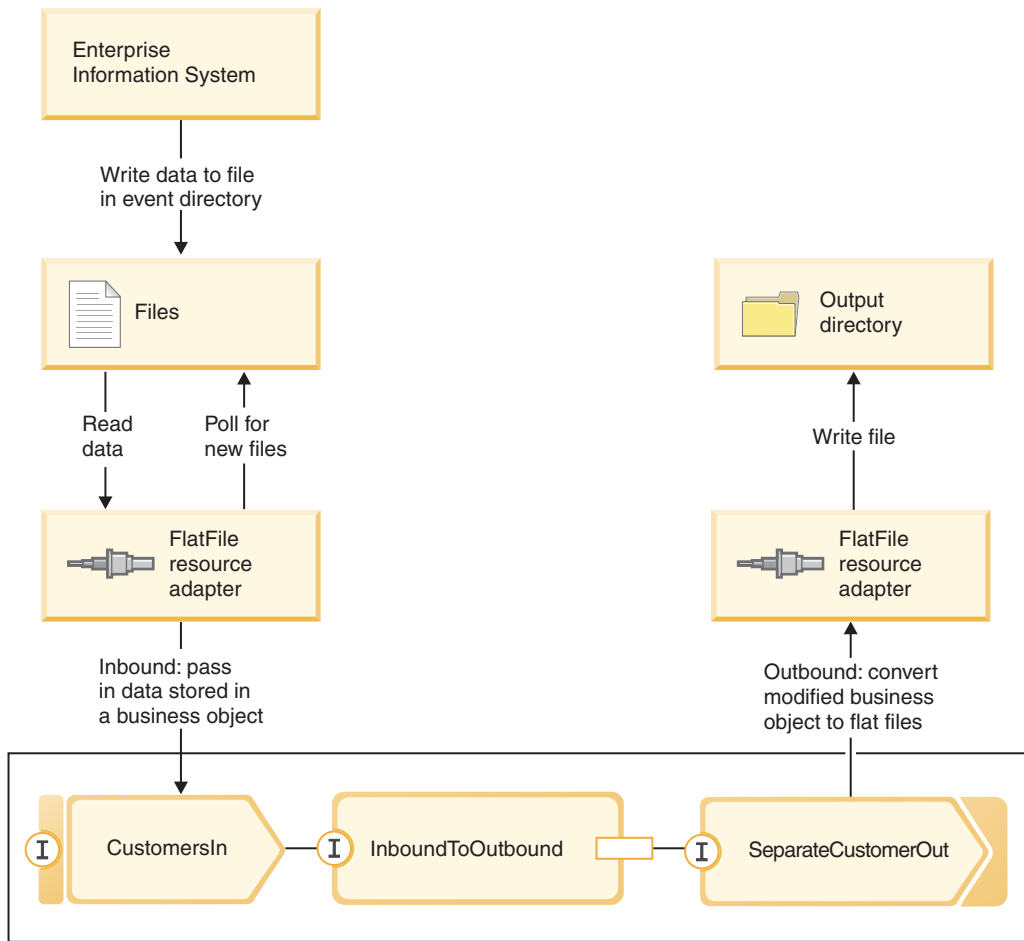
- 作成 - データを新規ファイルに保管するか、空のファイルを作成します。
- 追加 - 既存のファイルにデータを追加します。まだファイルが存在していなかった場合は、新規ファイルが作成されます。
- 上書き - 既存のファイルを新規データで上書きします。
- 削除 - 既存のファイルを削除します。
- 存在 - 特定のファイルが存在するかどうかを検査します。
- リスト - ディレクトリー内のファイルをリストします。
- 取得 - ファイルから内容を読み取ります。

シンプル外部サービス・ウィザードでは、作成操作と追加操作のみが使用されます。

このチュートリアルでのアプリケーションの概要

顧客レコード情報を収集するために、ファイル・システム内の特定のディレクトリーをモニターするアプリケーションを作成する場合を考えてみます。ファイルが作成されると、アダプターは指定されたファイル・スプリッター区切り文字を使用して、ファイルの内容をビジネス・オブジェクトに分割します。それぞれのビジネス・オブジェクトについてエクスポートが呼び出されます。エクスポートはメディエーション・フロー・ヘワイヤリングされており、そのフローはアダプターを呼び出して、レコードを出力ディレクトリー内の独自のファイルに書き込みます。

次の図は、このシナリオを示しています。



顧客レコードを含んでいるファイルがインバウンド・イベント・ディレクトリーに入っている場合、リソース・アダプターはそれらを取得し、顧客ビジネス・オブジェクトを作成し、モジュール内のエクスポートを呼び出します。このエクスポートはメディエーション・フローを開始します。メディエーション・フローはファイル・サービスを呼び出し、そのファイル・サービスは、既存のファイルの数に基づき、その名前を使用して、新しいファイル名を生成します。

第 2 章 ユーザーによるビルド

Advanced

ユーザーは、このサンプルを自分自身でビルドすることができます。

学習目標

IBM Integration Designer でウィザードとエディターを使用してこの実地サンプルをビルドする際は、以下のことを行います。

- ディレクトリーを作成する
- 開発するサービスを格納するためのモジュールを作成する
- 開発時間を節約するために、ビジネス・オブジェクトをインポートする
- アダプター・パターン・ウィザードを使用して、インバウンド・サービスを作成する
- 同じウィザードを使用してアウトバウンド・サービスを作成する
- メディエーション・フローを作成します。
- モジュールのデプロイ
- モジュールのテスト

ディレクトリーおよび入力ファイルの作成

Advanced

ファイルの取り出しと保管に、いくつかのディレクトリーが使用されます。入力ファイルは、テストに使用されます。

使用するファイル・システムに以下のディレクトリーを作成してください (Windows オペレーティング・システムを想定)。

1. `c:\%flatfiles` ディレクトリーを作成します。
2. フラット・ファイル・ディレクトリーに以下のサブディレクトリーを作成します。
 - a. `c:\%flatfiles\inboundevents`
 - b. `c:\%flatfiles\inboundarchive`
 - c. `c:\%flatfiles\outputdir`

Linux ユーザーの場合: Linux ファイル・システムを使用して同様なファイル構造を作成してください。

3. 次の `customer.txt` ファイルを作成し、`c:\%flatfiles` ディレクトリーに追加します。アプリケーションをテストするときは、そのファイルを `c:\%flatfiles\inboundevents` ディレクトリーにコピーできます。アプリケーションは、そのファイルを処理すると、このディレクトリーから削除します。

名前空間値 `SeparateCustomers` は、次のセクションで作成するモジュール名と一致する必要があることに注意してください。

```
<?xml version="1.0" encoding="UTF-8"?>
<p:Customer xsi:type="p:Customer"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:p="http://SeparateCustomers">
  <customerName>John Doe</customerName>
  <Address>170 Baseline Ave</Address>
  <City>Ottawa</City>
  <State>ON</State>
</p:Customer>
####
<p:Customer xsi:type="p:Customer"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:p="http://SeparateCustomers">
  <customerName>Susan Kidman</customerName>
  <Address>104 Candlestick Park</Address>
  <City>Ottawa</City>
  <State>ON</State>
</p:Customer>
####
<p:Customer xsi:type="p:Customer"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:p="http://SeparateCustomers">
  <customerName>Katheleen Black</customerName>
  <Address>530 Chanticleer Road</Address>
  <City>Ottawa</City>
  <State>ON</State>
</p:Customer>
####
<p:Customer xsi:type="p:Customer"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:p="http://SeparateCustomers">
  <customerName>Gary White</customerName>
  <Address>793 Morin Street</Address>
  <City>Ottawa</City>
  <State>ON</State>
</p:Customer>
####
```

モジュールの作成

Advanced

モジュールは、プロジェクト・フォルダーによく似ています。モジュールには、開発するサービスが格納されます。

ビジネス・インテグレーション・パースペクティブでモジュールを作成するには、以下のステップに従います。

1. メニューから、「ファイル」>「新規」>「モジュール」を選択します。「新規モジュール」ウィンドウが開きます。
2. 「モジュール名」フィールドに SeparateCustomers と入力します。「終了」をクリックします。
3. 「ビジネス・インテグレーション」ビューにモジュールが作成されます。

モジュール名 SeparateCustomers は、前のセクションで定義された入力ファイルに指定されている名前空間と、同じ値でなければならないことに注意してください。

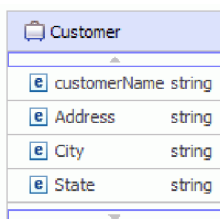
顧客ビジネス・オブジェクトの作成

Advanced

ビジネス・オブジェクトは、名前、住所、市町村や都道府県など、顧客データを保管するために使用されます。

顧客ビジネス・オブジェクトを作成するには、以下のステップに従います。

1. 「ビジネス・インテグレーション」ビューの「**SeparateCustomers**」モジュール内の「データ」を右クリックします。コンテキスト・メニューから、「新規」>「ビジネス・オブジェクト」を選択します。「新規ビジネス・オブジェクト」ウィンドウが開きます。
2. 「名前」には、Customer と入力します。「完了」をクリックします。ビジネス・オブジェクト・エディターが開きます。
3. 「フィールドをビジネス・オブジェクトに追加します。」アイコンをクリックします。新しいフィールドが追加されます。フィールドの名前を `customerName` に変更します。この手順を繰り返して、フィールド「Address」、「City」、「State」を追加します。タイプはデフォルトの `string` のままにします。変更内容を保存し、エディターを閉じます。



Customer	
customerName	string
Address	string
City	string
State	string

インバウンド・サービスの作成

Advanced

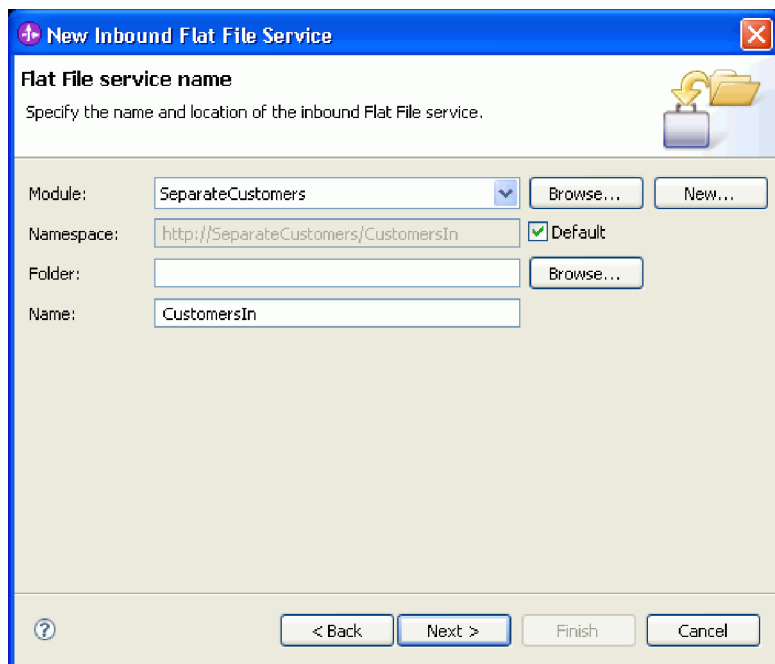
インバウンド・サービスは、処理のためにファイル・システムからファイルを受信します。

アダプター・パターン・ウィザードを使用してインバウンド・サービスを作成するには、以下のステップに従います。

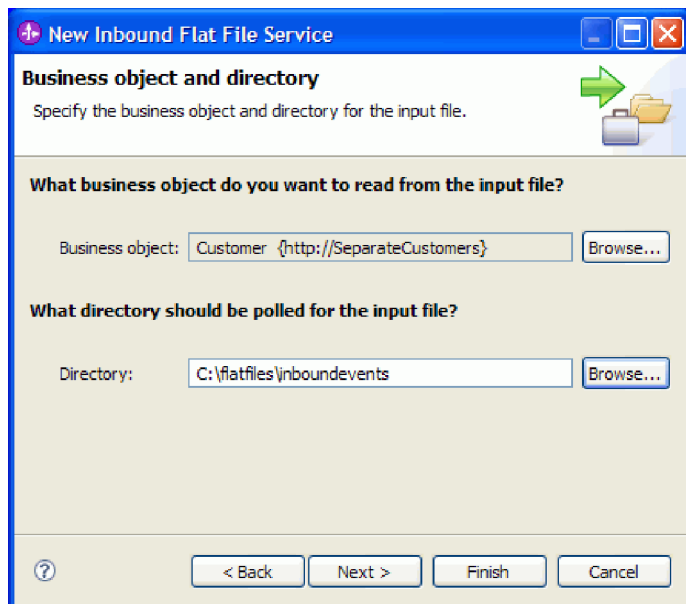
1. 「**SeparateCustomers**」を右クリックし、コンテキスト・メニューから「新規」>「外部サービス」を選択します。「外部サービス」ウィンドウが開きます。「アダプター」>「フラット・ファイル」を展開します。「シンプル: ローカル・ファイルから読み取るインバウンド・フラット・ファイル・サービスを作成する」を選択し、「次へ」をクリックします。

あるいは、「**SeparateCustomers**」 - 「アセンブリー・ダイアグラム」に切り替え、アセンブリー・エディターのパレットから「インバウンド・アダプター」を展開します。「フラット・ファイル」アダプターを選択してから、アセンブリー・エディターのキャンバスをクリックします。「外部サービス」ウィンドウが開きます。「アダプター」>「フラット・ファイル」を展開します。「シンプル: ローカル・ファイルから読み取るインバウンド・フラット・ファイル・サービスを作成する」を選択し、「次へ」をクリックします。

2. 「新規インバウンド・フラット・ファイル・サービス」ページが開きます。名前を **CustomersIn** に変更し、「次へ」をクリックします。



- 「ビジネス・オブジェクトおよびディレクトリー」ページが開きます。「参照」を使用して、前に作成したビジネス・オブジェクトの「Customer」までナビゲートし、これを選択します。ディレクトリー項目には、既にディレクトリーおよび入力ファイルの作成で入力ファイルを入れた flatfiles\inboundevents ディレクトリーを選択します。「次へ」をクリックします。

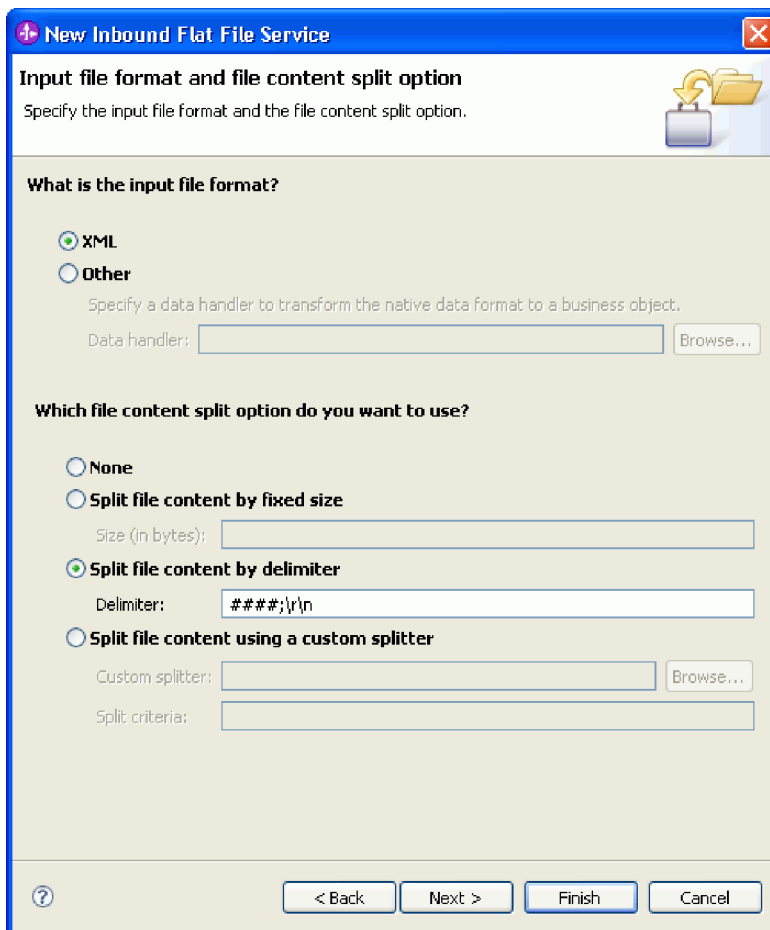


- 「入力ファイル・フォーマットとファイル内容の分割オプション」ページが開きます。デフォルトの入力ファイル・フォーマットの「XML」を受け入れます。XML ファイル・フォーマットは、ビジネス・オブジェクトを格納するためにサービス指向アーキテクチャー (SOA) アプリケーションで使用される標準ファイル・フォーマットです。ただし、前に作成したカスタム・データ・ハンドラーか、提供されたデータ・ハンドラーの 1 つを使用することもできます。データ・ハンドラーを使用すると、例えばバイト・ストリーム、区切り文字で区切られたデータ (コンマ区切り値など)、固定幅データ、または

JavaScript Object Notation (JSON) などのネイティブ入力データ・フォーマットを、ビジネス・オブジェクトに変換することができます。データ・ハンドラーについては、インフォメーション・センターの『データ・ハンドラー、フォールト、およびレジストリーの処理』を参照してください。作成方法を示す例も記載されています。

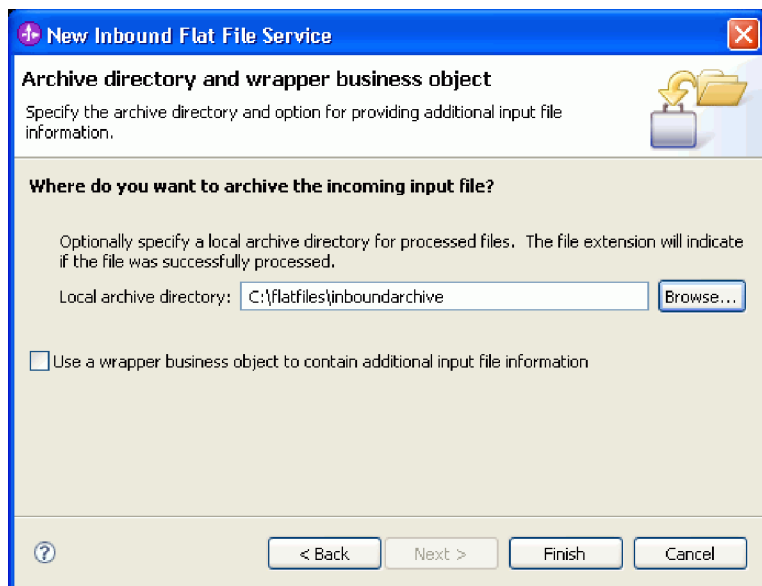
「区切り文字でファイル内容を分割する」を選択します。フィールドに `###;\r\n` と入力します。区切り文字は `####` です。セミコロン (;) は、`####` の後にさらに区切り文字があることを示します。Windows 環境の場合、`\r\n` は改行を示します。Linux ユーザーの場合は、`####;\n` を使用します。「次へ」をクリックします。

詳細および例については、ファイル分割に関するフラット・ファイルのドキュメンテーションを参照してください。



5. 「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」ページが開きます。既に作成した `flatfiles\inboundarchive` ディレクトリーまでナビゲートします。「終了」をクリックします。インバウンド・サービスが作成されます。

処理したファイルのレコードが必要な場合は、オプションのアーカイブ・ディレクトリーが使用されません。時間の経過とともにファイルのコレクションは増大するため、管理者またはアプリケーションは定期的にこれらを補助ストレージに移動する必要があります。入力ファイルの名前およびイベント・ディレクトリーにアクセスする必要がある場合は、ラッパー・ビジネス・オブジェクトを使用します。



いくつかの警告が作成されます。それらは、このサンプルで後続のステップを完了することによって訂正されます。

以下の成果物が作成されます。

表 1. アダプター・パターン・ウィザードによって作成される成果物

成果物	名前	説明
エクスポート	CustomersIn	エクスポートは、モジュールを外部（この例では WebSphere Adapter for Flat Files）に公開します。
インターフェース	CustomersIn	このインターフェースには、呼び出すことができる操作が格納されています。
操作	emit	emit がこのインターフェース内の唯一の操作です。

その他の生成された成果物を使用すると、サービス・コンポーネント・アーキテクチャー環境内で作動しているサービスと別の環境との間で、データを透過的に受け渡すことができます。それらは、データ・バインディング (FlatFileXMLDataBinding) とデータ・ハンドラー (UTF8XMLDataHandler) です。

アウトバウンド・サービスの作成

Advanced

アウトバウンド・サービスは、処理が済んだファイルをファイル・システムに出力します。

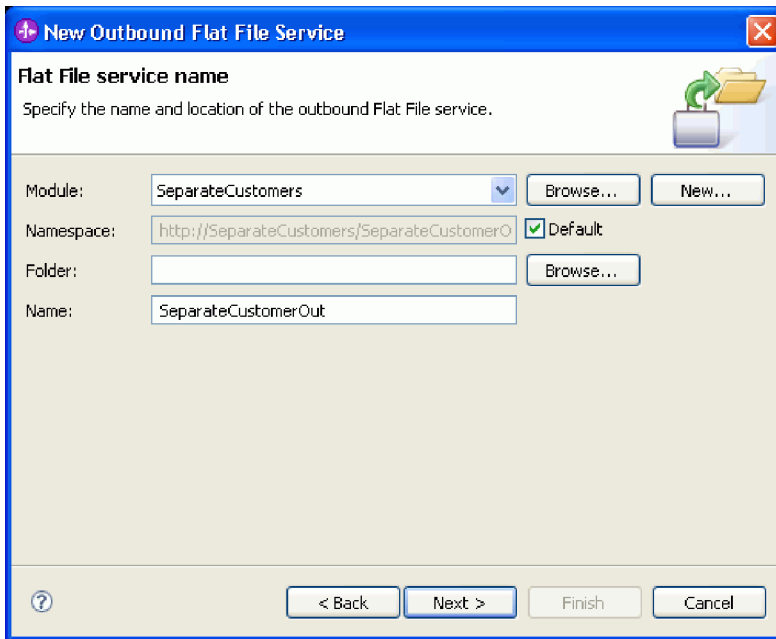
アダプター・パターン・ウィザードを使用してアウトバウンド・サービスを作成するには、以下のステップに従います。

1. 「**SeparateCustomers**」を右クリックし、コンテキスト・メニューから「新規」>「外部サービス」を選択します。「外部サービス」ウィンドウが開きます。「アダプター」>「フラット・ファイル」を展開します。「**シンプル: ローカル・ファイルに書き込むアウトバウンド・フラット・ファイル・サービスを作成する**」を選択し、「次へ」をクリックします。

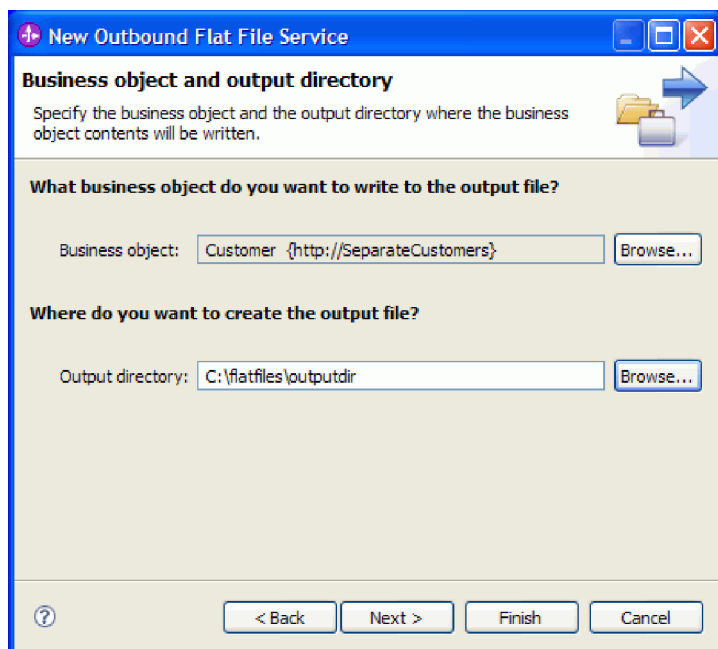
あるいは、「**SeparateCustomers**」 - 「アセンブリー・ダイアグラム」に切り替え、アセンブリー・エディターのパレットから「アウトバウンド・アダプター」を展開します。「フラット・ファイル」アダプ

ターを選択してから、アセンブリ・エディターのキャンバスをクリックします。「外部サービス」ウィンドウが開きます。「アダプター」>「フラット・ファイル」を展開します。「シンプル: ローカル・ファイルに書き込むアウトバウンド・フラット・ファイル・サービスを作成する」を選択し、「次へ」をクリックします。

2. 「新規アウトバウンド・フラット・ファイル・サービス」ページが開きます。名前を **SeparateCustomerOut** に変更し、「次へ」をクリックします。



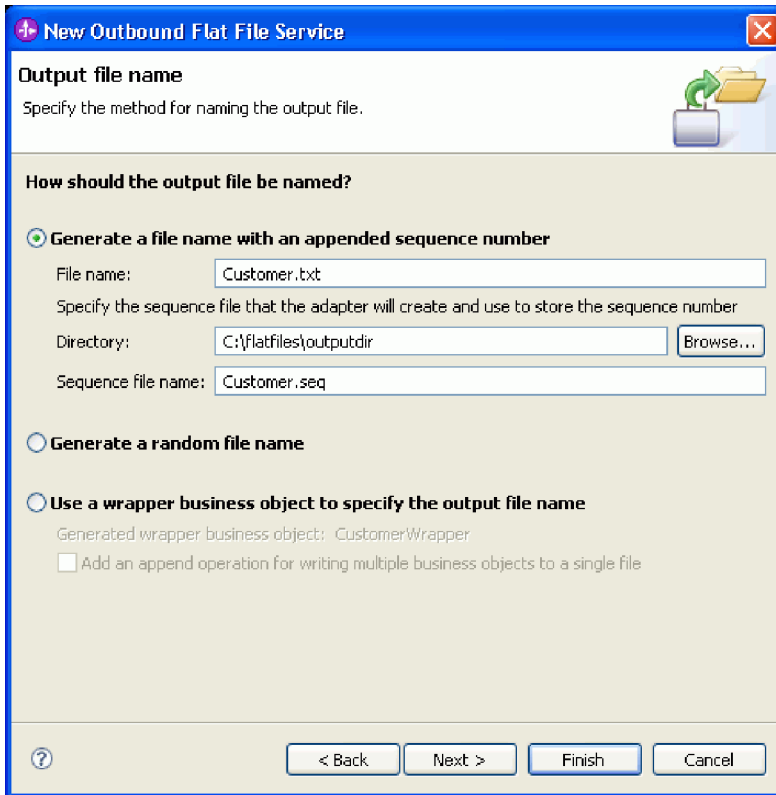
3. 「ビジネス・オブジェクトおよび出力ディレクトリー」ページが開きます。「参照」を使用して、前に作成したビジネス・オブジェクトの「**Customer**」までナビゲートし、これを選択します。ディレクトリー項目には、既にディレクトリーおよび入力ファイルの作成で作成した flatfiles%outputdir 出力ディレクトリーを選択します。「次へ」をクリックします。



4. 「出力ファイル名」ページが開きます。さまざまな方法を使用して、出力ファイル名を作成できます。デフォルトを受け入れると、それぞれの顧客レコードが Customer.<number of customer record>.txt のフォーマットの出力ファイルになります。Customer.seq ファイルには、処理された顧客レコードの数が格納されます。

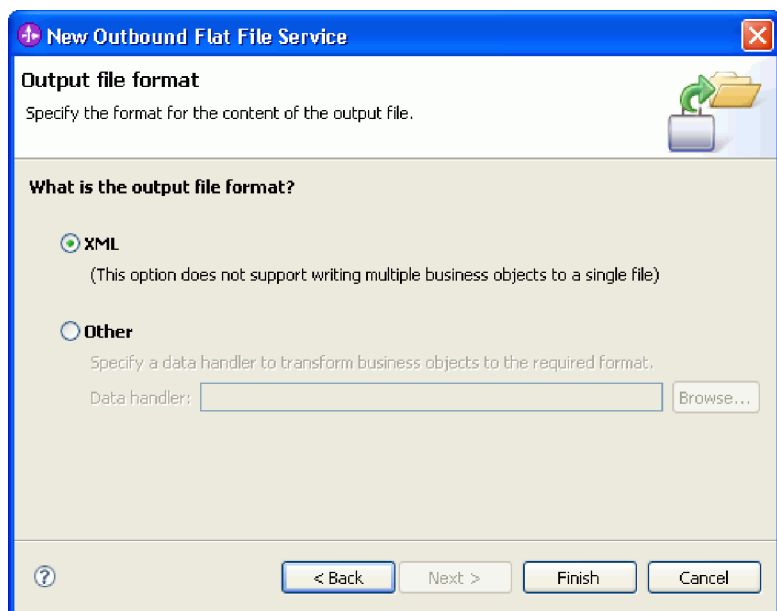
アプリケーションで固有の生成済み出力ファイル名が必要な場合は、ランダム・ファイル名を生成すると便利です。アプリケーションでビジネス・オブジェクト名を指定する場合は、「出力ファイル名の指定にラッパー・ビジネス・オブジェクトを使用する」を選択します。この場合は、追加操作を選択して、すべてのビジネス・オブジェクトを複数のファイルでなく、単一ファイルに書き込むこともできます。

「次へ」をクリックします。



5. 「出力ファイル・フォーマット」ページが開きます。デフォルトの「XML」を受け入れ、「完了」をクリックします。

インバウンド・サービスの作成に記載されているように、XML は SOA アプリケーションの標準ファイル・フォーマットです。ただし、ネイティブ・データが、例えばバイト・ストリーム、区切り文字で区切られたデータ (コンマ区切り値など)、固定幅データ、または JavaScript Object Notation (JSON) など、別のフォーマットになっている場合があります。そのような場合は、データ・ハンドラーを参照して選択できます。データ・ハンドラーについては、インフォメーション・センターの『データ・ハンドラー、フォールト、およびレジストリーの処理』を参照してください。作成方法を示す例も記載されています。



以下の成果物が作成されます。

表2. アダプター・パターン・ウィザードによって作成される成果物

成果物	名前	説明
インポート	SeparateCustomerOut	インポートを使用すると、サービスで WebSphere Adapter for Flat Files を使用して、ファイルを出力できます。
ビジネス・オブジェクト	CreateResponse、 UnstructuredContent	アプリケーションの出力の作成に使用される CreateResponse は、Customer ビジネス・オブジェクトを入力として使用します。UnstructuredContent ビジネス・オブジェクトは、汎用入力タイプに使用できます。
フォールト処理用の ビジネス・オブジェクト	DuplicateRecordFault、 MissingDataFault、 PrimaryKeyPairType、 RecordNotFoundFault、 WBIFault	これらのビジネス・オブジェクトは、例外の処理に使用できます。ビジネス例外がスローされると、アダプターはフォールト例外をスローし、フォールト情報を呼び出し側コンポーネントに渡します。この情報を使用したい場合は、フォールト・セレクター構成ファイルを作成する必要があります。
インターフェース	SeparateCustomerOut	このインターフェースには、呼び出すことができる操作が格納されています。
操作	作成	この操作は、出力ディレクトリー内にファイルを作成します。

ここで述べる特定のアプリケーションでは、片方向のインバウンド要求のみを使用するので、フォールト処理は必要ありません。以下のステップでは、インポート・バインディングおよびインターフェースからフォールトを除去します。

インポート・バインディングからフォールトを除去するには、以下のステップに従います。

1. アセンブリ・エディターで「SeparateCustomerOut」を選択し、「プロパティ」ビューで「フォールト構成」タブを選択します。
2. 「デフォルトのフォールト構成」を選択します。右側のパネルで、「クリア」をクリックします。
3. 「デフォルトのフォールト構成」、およびその下の「作成」操作を展開します。「MISSING_DATA」を選択します。右側のパネルで、「クリア」をクリックします。
4. 前のステップを「DUPLICATE_RECORD」と「RECORD_NOT_FOUND」についても繰り返します。

5. 変更内容を保存します。メニューから、「ファイル」>「保存」を選択します。アセンブリー・エディターを閉じます。

インターフェースからフォールトを除去するには、以下のステップに従います。

1. 「ビジネス・インテグレーション」ビューで、「インターフェース」を展開します。「SeparateCustomerOut」をダブルクリックして、インターフェース・エディターを開きます。
2. 「MISSING_DATA」フォールトを選択します。次に、「削除」アイコンを選択します。これは、エディターの上部右側にあります。
3. 同様に、「DUPLICATE_RECORD」と「RECORD_NOT_FOUND」のフォールトを削除します。
4. 変更内容を保存します。メニューから「ファイル」>「保存」を選択します。インターフェース・エディターを閉じます。

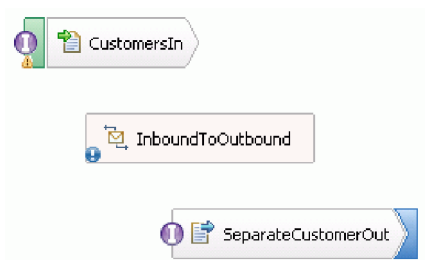
メディエーション・フローの作成

Advanced

インポートとエクスポートの間でデータの受け渡しを可能にするには、メディエーション・フローによってそれらを互いにリンクする必要があります。インターフェースが異なるためです。

メディエーション・フローを作成するには、以下のステップに従います。

1. アセンブリー・エディターを開きます (まだ開いていない場合)。「SeparateCustomers」の下の「アセンブリー・ダイアグラム」をダブルクリックします。
2. 「パレット」から「メディエーション・フロー」を選択し、マウス・ボタンを離します。メディエーション・フローをキャンバスに配置する場合は、マウス・ボタンを再クリックします。それを「CustomersIn」と「SeparateCustomerOut」の間に配置してください。メディエーション・フローを InboundToOutbound に名前変更します。作業を保存します。メニューから、「ファイル」>「保存」をクリックします。



「問題」ビューに表示される項目は、このセクションが終了すると非表示になります。

3. 「InboundToOutbound」をダブルクリックします。メッセージが表示され、コンポーネントをこの時点で実装するかどうかを尋ねてきます。「はい」をクリックします。「実装の生成」ウィンドウが開きます。デフォルトを受け入れ、「OK」をクリックします。メディエーション・フロー・エディターが開きます。
4. エディターの上部で、「インターフェースの追加」を選択します。



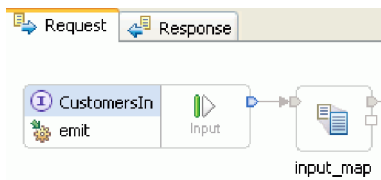
「インターフェースの追加」ウィンドウで、「**CustomersIn**」を選択して「**OK**」をクリックします。

5. エディターの上部で、「参照の追加」を選択します。



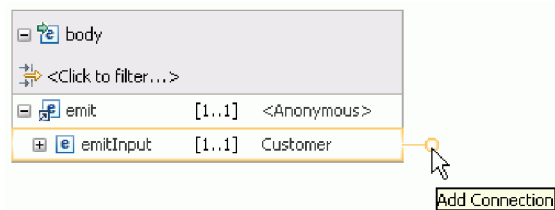
「参照の追加」ウィンドウで、「**SeparateCustomerOut**」を選択して「**OK**」をクリックします。

6. メディエーション・フロー・エディターで、「**emit**」リンクをクリックし、「**メディエーション・フロー・テンプレートの選択**」ウィンドウで「**操作マップ**」リンクをクリックします。「**参照操作の選択**」ウィンドウで、「**SeparateCustomerOutPartner**」を選択して「**OK**」をクリックします。作業を保存します。メニューから、「**ファイル**」>「**保存**」をクリックします。
7. メディエーション・フロー・エディターの「**要求**」タブで、「**input_map**」ノードをダブルクリックして入力マップを実装します。



「**新規 XML マップ**」ウィンドウが開きます。デフォルトを受け入れて「**完了**」をクリックします。マップ・エディターが開きます。

8. マップ・エディターで、「**emit**」エレメントを展開し「**emitInput**」エレメントを表示します。「**create**」エレメントを展開し「**createInput**」エレメントを表示します。「**emitInput**」エレメントの横の「**接続の追加**」ハンドルをクリックします。このハンドルを「**createInput**」エレメントまでドラッグして、「**移動**」変換を作成します。

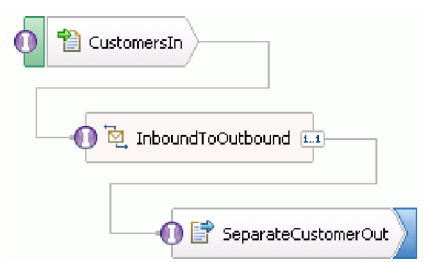


作業を保存します。マップ・エディターを閉じます。メディエーション・フロー・エディターで、作業を保存します。メディエーション・フロー・エディターを閉じます。アセンブリー・エディターで、作業を保存します。

「**emit**」は片方向操作であり、アプリケーションは内容を複数のファイルに分割し、分割された内容から出力ファイルを作成しているため、アプリケーションはこの時点で完了しています。したがって、応答をマップする必要はありません。

9. コンポーネントを接続します。アセンブリー・エディターで、「**CustomersIn**」の右端にあるハンドルを選択し、「**InboundToOutbound**」の左端までドラッグします。メッセージが表示され、エクスポートのサービス・インターフェースがターゲットに追加されることを知らせてきます。「**OK**」をクリックします。「**InboundToOutbound**」の右端にあるハンドルを選択し、「**SeparateCustomerOut**」の左端ま

でドラッグします。メッセージが表示され、ソース上にマッチング参照が作成されることを知らせてきます。「OK」をクリックします。作業を保存します。



第 3 章 サンプルの実行

Advanced

サンプルのビルドまたはインポートが完了したら、サンプルを実行できます。

既製のサンプルをインポートした場合は、ディレクトリーおよび入力ファイルの作成に示すように、ディレクトリーを作成し、入力ファイルを追加する必要があります。

サンプルを実行するには、以下のステップに従います。

- サンプルを自分自身でビルドするのではなく、インポートした場合は、WebSphere Adapter for Flat Files をインポートします。
- モジュールをサーバーにデプロイします。
- モジュールを実行し、テストします。

アダプターのインポート

Advanced

2 つのバージョンのサンプルが用意されています。1 つのバージョンには、まだワークスペースにフラット・ファイル・アダプターがない場合に備えて、それが含まれています。もう 1 つのバージョンには、フラット・ファイル・アダプターが含まれていません。前にサンプルに対して作業を行い、フラット・ファイル・イベントに応答するサービスを作成してある場合、またはフラット・ファイル・アダプターを使用するファイル処理したことがある場合は、フラット・ファイル・アダプターがワークスペースにインポートされています。しかし、アダプターをインポートする必要がある場合は、以下のステップが提供されます。

アダプターをインポートしたら、clean 関数 (以下のステップで同様に説明) を使用する必要があります。

WebSphere Adapter for Flat Files をインポートするには、以下のステップに従います。

1. メニューから、「ファイル」>「インポート」を選択します。「インポート」ウィンドウが開きます。「Java EE」を展開し、「RAR ファイル」を選択します。「次へ」をクリックします。「コネクター・インポート」ページが開きます。
2. 「コネクター・ファイル」フィールドの横にある「参照」をクリックします。次のディレクトリーまでナビゲートし、RAR ファイルを選択します。

```
<installdir>%ResourceAdapters%FlatFile<version number>
```

残りのフィールドには、自動的に入力されます。

「EAR にプロジェクトを追加」をクリアします。「終了」をクリックします。メッセージが表示され、Java EE パースペクティブに切り替えるかどうかを尋ねてきます。「いいえ」をクリックします。

3. アダプターがインポートされ、「ビジネス・インテグレーション」ビューに表示されます。
4. clean 関数を使用して値を正しく設定します。メニューから「プロジェクト」>「Clean」を選択します。
5. 「Clean」ウィンドウでデフォルトを受け入れ、「OK」をクリックします。

第 4 章 インポート

Advanced

このサンプルの完全な既製バージョンをインポートすることができます。

既製のサンプルをインポートするには、以下を実行します。

1. IBM Integration Designer で、「ヘルプ」>「サンプルおよびチュートリアル」>「**IBM Integration Designer**」を選択します。「サンプルおよびチュートリアル」ページが開きます。
2. 「ファイルの読み書き」セクションの下で「インポート」リンクをクリックします。「サンプルのインポート」ページが開きます。このページを使用して、単にそのサンプルだけをインポートすることも（ワークスペース内にフラット・ファイル・コネクタ・プロジェクト `CWYFF_FlatFile` が既に存在する場合）、フラット・ファイル・アダプター付きのサンプルをインポートすることもできます。該当するオプションを選択し、「**OK**」を選択します。
3. 既製のサンプルがワークベンチにインポートされます。

サンプルを実行する手順は、トピック『サンプルの実行』に書かれています。その説明には、インポートしたアプリケーションにアダプターを追加するための別のインポートが含まれています。アダプターのインポートを参照してください。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものであり、本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM® 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510
東京都中央区日本橋箱崎町19番21号
日本アイ・ビー・エム株式会社
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または保証条件は適用されないものとします。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。

*Intellectual Property Dept. for IBM Integration
Designer
IBM Canada Ltd.
8200 Warden Avenue
Markham, Ontario
L6G 1C7 Canada*

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、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。お客様は、IBM 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。

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的創作物にも、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。

© (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。 © Copyright IBM Corp. 2000, 2009. 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、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

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。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

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。

Linux は、Linus Torvalds の米国およびその他の国における登録商標です。

Adobe は、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

ご利用条件

Advanced

これらの資料は、以下の条件に同意していただける場合に限りご使用いただけます。

個人使用: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、これらの資料またはその一部について、二次的著作物を作成したり、配布（頒布、送信を含む）または表示（上映を含む）することはできません。

商業的使用: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することができます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

ここで明示的に許可されているもの以外に、資料や資料内に含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。

資料の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入関連法規を含む、すべての関連法規を遵守するものとします。

IBM は、これらの資料の内容についていかなる保証もしません。これらの資料は、特定物として現存するままの状態を提供され、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なしで提供されます。

© Copyright IBM Corporation 2005, 2012. All Rights Reserved.