

**IBM Integration Designer**  
バージョン 7 リリース 5

## **Hello World Part 1: 入門**

**IBM**

注

本書および本書で紹介する製品をご使用になる前に、35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。

このエディションは、IBM Integration Designer のバージョン 7.5 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： IBM Integration Designer  
Version 7 Release 5  
Hello World Part 1: Getting Started

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 2005, 2011.

---

## 目次

<b>第 1 章 概要</b> . . . . .	<b>1</b>	メディエーション・フロー実装の作成 . . . . .	17
概要 . . . . .	1	<b>第 3 章 サンプルの実行</b> . . . . .	<b>27</b>
概念 . . . . .	6	モジュールのデプロイ . . . . .	27
<b>第 2 章 ユーザーによるビルド</b> . . . . .	<b>9</b>	モジュールのテスト . . . . .	27
モジュールの作成 . . . . .	9	サーバーからのモジュールの除去 . . . . .	30
ライブラリー・プロジェクトの作成 . . . . .	9	<b>第 4 章 インポート</b> . . . . .	<b>33</b>
Web サービス・ファイルのコピー . . . . .	10	<b>特記事項</b> . . . . .	<b>35</b>
新しいサービス・インターフェースの作成 . . . . .	11	<b>ご利用条件</b> . . . . .	<b>39</b>
ライブラリーを参照するメディエーション・モジュール・プロジェクトの作成 . . . . .	13		
メディエーション・モジュールの組み立て . . . . .	14		



---

## 第 1 章 概要

このサンプルでは、IBM® Integration Designer を使用して、既存の Web サービスに対するメディエーション呼び出しを作成し、デプロイし、実行する方法を示します。

このサンプルでは、以下のアクティビティの実行方法を説明します。

- ワークベンチをナビゲートします。
- 共有成果物用のライブラリー・プロジェクトを作成します。
- 統合ロジック用のメディエーション・モジュール・プロジェクトを作成します。
- Web サービスを呼び出します。
- メディエーション・フローでの Web サービス呼び出しのメディエーションにより、そのインターフェースを別のインターフェースにマップします。
- 異なるインターフェースを持つ新しいサービスとして、メディエーション・フローを公開します。
- メディエーション・モジュールをデプロイします。
- メディエーション・モジュールを実行およびテストします。

このサンプルは、WebSphere® Enterprise Service Bus または IBM Process Server で実行されます。

---

### 概要

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

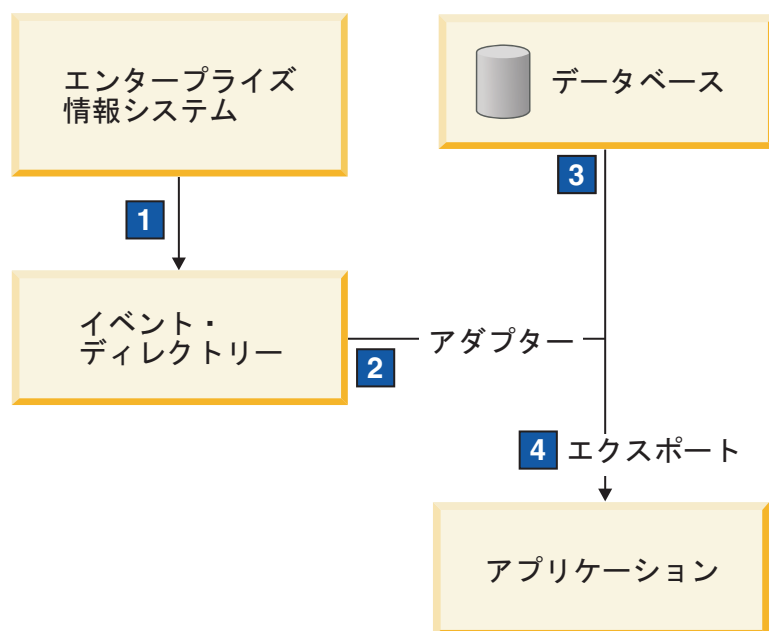
インバウンド処理 は WebSphere Adapter for Flat Files がファイル・システムをモニターし、新規ファイルを読み取り、アプリケーション内の操作ヘデータを送信する操作モードです。

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戻す操作モードです。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。

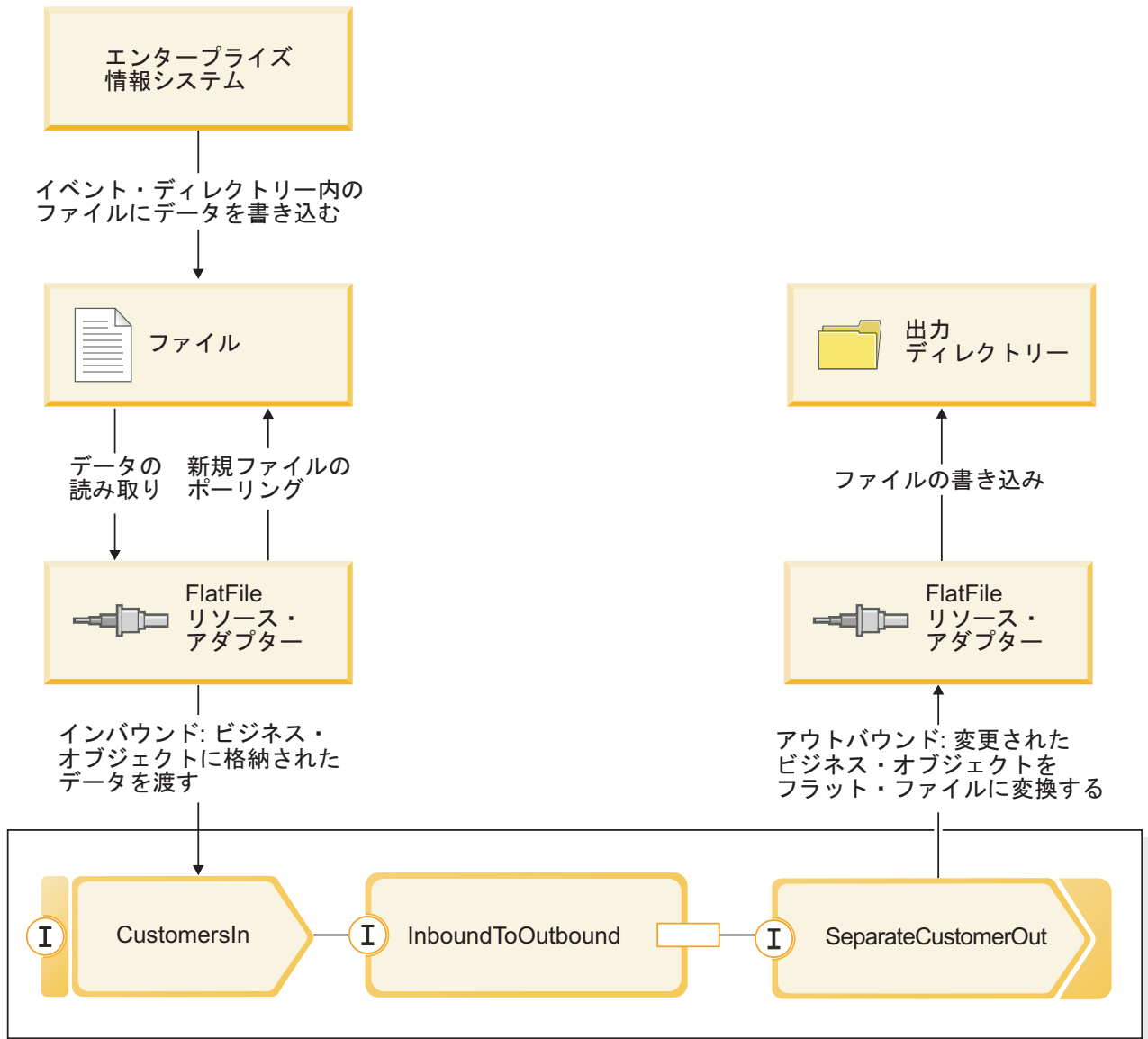
- 作成 - データを新規ファイルに保管するか、空のファイルを作成します。
- 追加 - 既存のファイルにデータを追加します。まだファイルが存在していなかった場合は、新規ファイルが作成されます。
- 上書き - 既存のファイルを新規データで上書きします。
- 削除 - 既存のファイルを削除します。
- 存在 - 特定のファイルが存在するかどうかを検査します。
- リスト - ディレクトリー内のファイルをリストします。
- 取得 - ファイルから内容を読み取ります。

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・フローへ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。

## 概念

IBM Integration Designer は、サービスを呼び出して公開するための統合ロジックを作成するとともに、アプリケーションとデータを統合するビジネス・プロセスを作成するためのビジネス・インテグレーション製品です。

メディエーション・フローは、既存のサービス (プロバイダー) と、これらのサービスを使用したいクライアント (リクエスター) の間で受け渡されるメッセージを傍受し、変更します。メディエーション・フローは、モジュールまたはメディエーション・モジュールに実装できます。メディエーション・モジュールは、

WebSphere Enterprise Service Bus に加え、IBM Process Server にデプロイできます。このサンプルでは、ユーザーのデプロイメント環境に関係なくすべての IBM Integration Designer ユーザーに適用できるように、メディエーション・モジュールを使用します。

ビジネス・プロセスとは、定義された一連のビジネス・アクティビティのことであり、これらのアクティビティは、ビジネス目標の達成に必要な手順を表しています。ビジネス・プロセスは、標準のビジネス・インテグレーション・モジュールでのみ実装することができ、メディエーション・モジュールで実装することはできません。このため、ビジネス・プロセスおよびビジネス・プロセスを含むモジュールは、IBM Process Server でのみデプロイおよび実行できます。(「Hello World パート 2」サンプルで、ビジネス・プロセスを紹介します。)

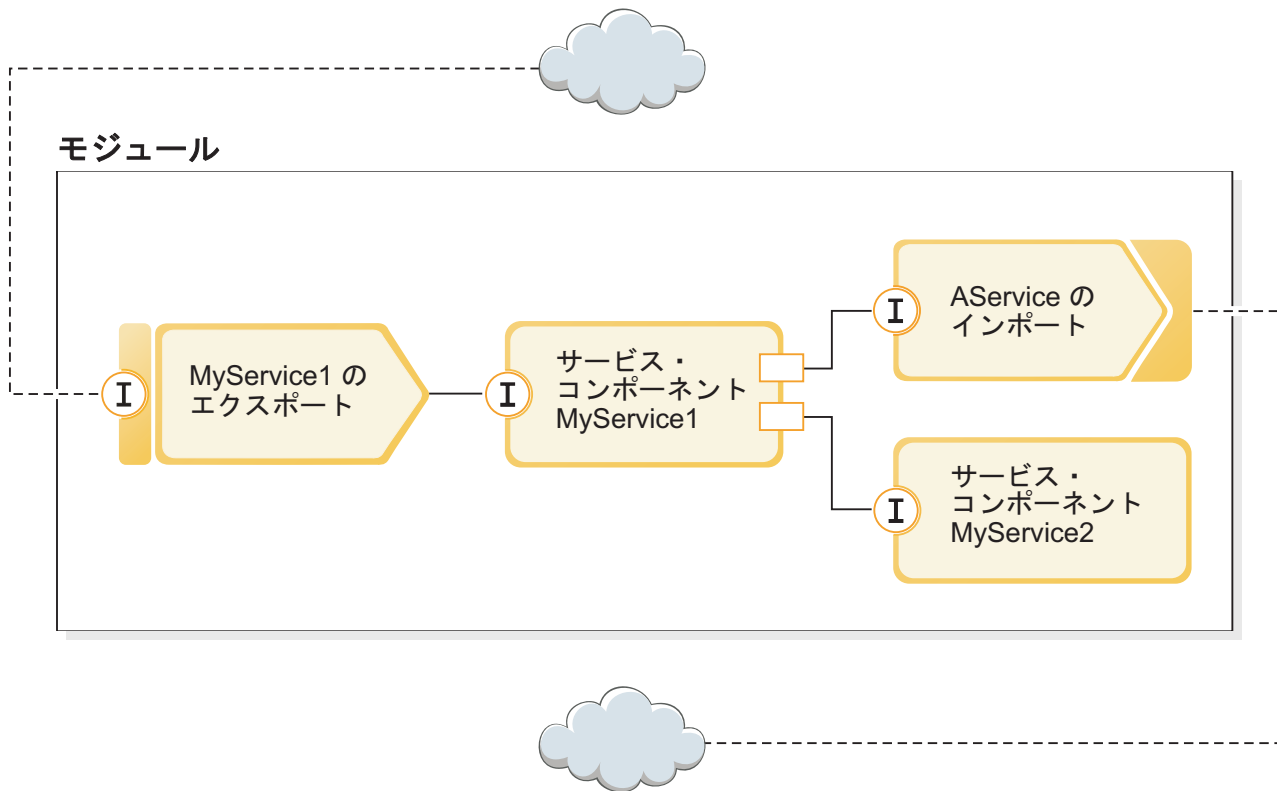
IBM Integration Designer 内のモジュールとメディエーション・モジュールは、疎結合方式で互いを呼び出すコンポーネントで構成されています。この疎結合を実現するには、各コンポーネントで、コンポーネント自身の呼び出しに使用できるインターフェースだけでなく、呼び出し先または参照先にするコンポーネントのインターフェースも宣言します。実際のコンポーネントではなく、これらの必要なコンポーネントのインターフェースのみを定義することにより、これらの参照に対応するために使用するコンポーネントを簡単に変更できます。

これらのコンポーネントは、メディエーション・フロー、ビジネス・プロセス、または Java など、サポートされているいくつかの方法で実装されています。各実装には独自のエディターがあり、このサンプルでは、メディエーション・フロー・エディターを紹介します。

これらのコンポーネントは、アセンブリ・エディターを使用して、モジュール内で一緒に定義およびワイヤリングされます。コンポーネントの定義とは、コンポーネントに名前を付け、そのインターフェースと参照のほかに実装タイプも識別し、適切なエディターでコンポーネントを開いて、その実装を定義することを意味しています。コンポーネントのワイヤリングとは、1 つのコンポーネントの参照を他のコンポーネントに接続して、その要件を満足することを意味しています。

特定サーバーへのデプロイメント用にコンポーネントをグループ化するには、モジュールを使用します。デプロイされたモジュールをクライアント・アプリケーションまたは別のモジュールから呼び出すには、エクスポート と呼ばれる別の種類のアセンブリ・ダイアグラム・ノードを使用してモジュール内のコンポーネントの 1 つをエクスポートし、このコンポーネントをリモート側で呼び出すことができますようにします。モジュール境界を超えてコンポーネントを公開するのに使用できるいくつかのオプションがあります。コンポーネントは、Web サービスとして、または Java Message Service などのキューイング・テクノロジーを使用して公開できます。これらのオプションは、バインディングと呼ばれています。モジュール間通信のために、SCA バインディングと呼ばれる最適化されたオプションがあります。SCA (Service Component Architecture) は、この疎結合機能の基礎となる標準です。このサンプルでは SCA バインディングを使用するので、「Hello World パート 1」サンプル内で作成したモジュールを、「Hello World パート 2」サンプル内のモジュールが呼び出すことができます。

場合によっては、モジュール内のコンポーネントが、モジュール外の既存サービスを呼び出すことも必要になります。これは、インポート と呼ばれる別の種類のアセンブリ・ダイアグラム・ノードによって行われます。インポートは、モジュール内の外部サービスを表すために使用されます。これにより、これらのサービスを使用して、任意のローカル・コンポーネントと同様にコンポーネント参照を行うことができますようになります。以下の図では、中央に、2 つの参照を持つコンポーネント **MyService1** があります。一方には外部サービスのインポートで対応し、他方には別のローカル・コンポーネントで対応しています。また、以下の図に示すように、**MyService1** コンポーネントはエクスポートされ、モジュール外の他のサービスが、**MyService1** コンポーネントをリモート側で呼び出すことができますようにしています。



このサンプルでは、Web サービス・バインディング付きインポートを使用して、提供される既存の、ただし外部の Web サービスをメディエーション・フロー・コンポーネントが呼び出すことができるようにします。

IBM Integration Designer のすべてのコンポーネント (インポートとエクスポートを含む) はインターフェースを宣言するので、他のコンポーネントやクライアントは宣言側のコンポーネントを呼び出す方法を認識できます。一般に、これらのインターフェースは Web サービス記述言語 (WSDL) で定義されています。ただし、Java コンポーネントは Java インターフェースを使用することもできます。インターフェース・エディターを使用して、IBM Integration Designer で WSDL インターフェースを宣言できます。また、既存の WSDL ファイルを、使用するライブラリー・プロジェクトおよびモジュール・プロジェクトにインポートすることもできます。このサンプルでは、新しいインターフェースを作成し、既存のインターフェースに導入します。

インターフェースには 1 つ以上の操作が含まれています。各操作には 1 つ以上の入出力パラメーターが含まれており、それぞれ標準の組み込みデータ型またはビジネス・オブジェクトと呼ばれるユーザー定義のデータ型を使用します。このサンプルでは、ビジネス・オブジェクトを作成します。

---

## 第 2 章 ユーザーによるビルド

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

---

### モジュールの作成

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、以下の手順に従います。

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。「完了」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。

---

### ライブラリー・プロジェクトの作成

IBM Integration Designer では、ライブラリーは、複数のモジュールに必要なファイルを格納することができるプロジェクトです。最終的には、新しいサービスのインターフェースとビジネス・オブジェクトの成果物を「Hello World パート 2」サンプルと共有する必要があるため、それらを保持するライブラリーが必要です。それ以外の場合、使用するモジュール内に成果物を直接作成することもできます。

ライブラリー・プロジェクトを作成するには、以下の手順を実行します。

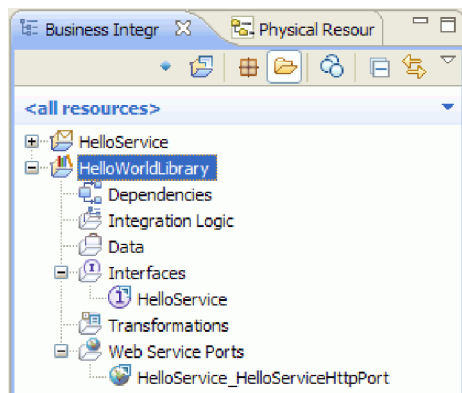
1. 「ビジネス・インテグレーション」ビューで、「HelloService」プロジェクトを右クリックして、「新規」>「プロジェクト」>「ライブラリー」を選択します。「新規ライブラリー」ウィザードが開きます。
2. 「ライブラリー名」フィールドで、`HelloWorldLibrary` と入力して、「終了」をクリックします。ライブラリー・プロジェクトが、「ビジネス・インテグレーション」ビューに表示されます。

## Web サービス・ファイルのコピー

新しいライブラリーの用途の 1 つは、提供されている呼び出し対象の Web サービスに対するエンドポイント WSDL ファイルを格納することです。

Web サービス・ファイルをライブラリーにコピーするには、以下の手順を実行します。

1. 「ビジネス・インテグレーション」ビューで、「HelloService」プロジェクトを展開し、次に「Web サービス・ポート」カテゴリを展開します。
2. 「HelloService\_HelloServiceHttpPort」を右クリックし、「ファイルの表示」を選択します。このアクションにより、「物理リソース」ビューが開きます。このビューには、プロジェクト内の基礎となるファイルが表示されます。「プロジェクト・リソース (Project Resources)」ビューと「ビジネス・インテグレーション」ビューの両方におけるプロジェクトの内容および構造について詳しくは、『ビジネス・インテグレーション・プロジェクト』およびそのページからリンクされているトピックを参照してください。
3. 「ビジネス・インテグレーション」ビューに戻ります。「HelloService\_HelloServiceHttpPort」ファイルを右クリックし、「コピー」を選択します。「複数成果物のコピー」ダイアログ・ボックスが開いたら、「OK」をクリックします。このアクションで、WSDL ファイルがクリップボードに格納されます。
4. 「HelloWorldLibrary」を右クリックし、「貼り付け」を選択します。これで、WSDL ファイルがライブラリー・プロジェクトにコピーされます。
5. これ以上表示しておく必要はないので、「HelloService」プロジェクトを省略し、「HelloWorldLibrary」プロジェクトを展開します。以下に示すように、その「インターフェース」と「Web サービス・ポート」のカテゴリを展開して、その WSDL ファイルからコピーされた内容を確認します。



注: 既に独自の Web サービスがある場合、それをライブラリー・プロジェクトまたはモジュール・プロジェクトに同様にコピーできます。コピーするには、クリップボードを使用するか、Windows ファイル・システムからドラッグ・アンド・ドロップするか、または「ファイル」>「インポート」>「ビジネス・インテグレーション」>「WSDL および XSD (WSDL and XSD)」を使用します。最後のオプションは、他のファイルへの参照を含むより複雑な WSDL ファイルを対象としています。

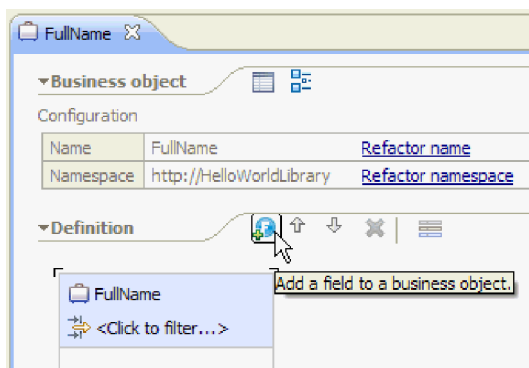
注: インターフェースとポートのアイコンの右上にある注釈は何かと考えるかもしれませんが。この注釈は、このインターフェースとポートが、複数のオブジェクトを含む WSDL ファイルの一部であることを示しており、これらのオブジェクトは、右クリックして「影響のリファクタリングまたは分析 (Refactor or Analyze Impact)」>「WSDL コンポーネントの抽出」を選択することにより抽出できます。ただし、このサンプルでは、そのタスクを実行する必要はありません。

## 新しいサービス・インターフェースの作成

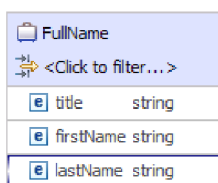
サービスは、3つの入力ストリング、つまり 敬称、名、および姓 を連結します。これらのフィールドを保持するためには、ビジネス・オブジェクトを作成し、これらのビジネス・オブジェクトの 1 つを入力として取り、連結されたストリング「Hello title firstname lastname」を返すインターフェースを作成する必要があります。

サービス・インターフェースを作成するには、以下の手順を実行します。

1. 「ビジネス・インテグレーション」ビューの「HelloWorldLibrary」で、「データ」カテゴリを右クリックして、「新規」>「ビジネス・オブジェクト」を選択します。「新規ビジネス・オブジェクト」ウィザードが開きます。
2. 「名前」フィールドに FullName と入力して、「終了」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. 新しいフィールドを作成するには、以下の図に示すように、ローカル・ツールバーで小さな「F」アイコンをクリックします (または「FullName」ボックスを右クリックして、「フィールドの追加」を選択します)。



4. 生成されたフィールドの名前「field1」を上書きし、title で置き換えます。名前が選択されていない場合、まずその名前をクリックして選択します。
5. 前のステップを繰り返して、さらに 2 つのフィールドを作成し、1 つの名前を「firstName」、もう 1 つの名前を「lastName」にします。最終的なビジネス・オブジェクトは、以下のようになります。



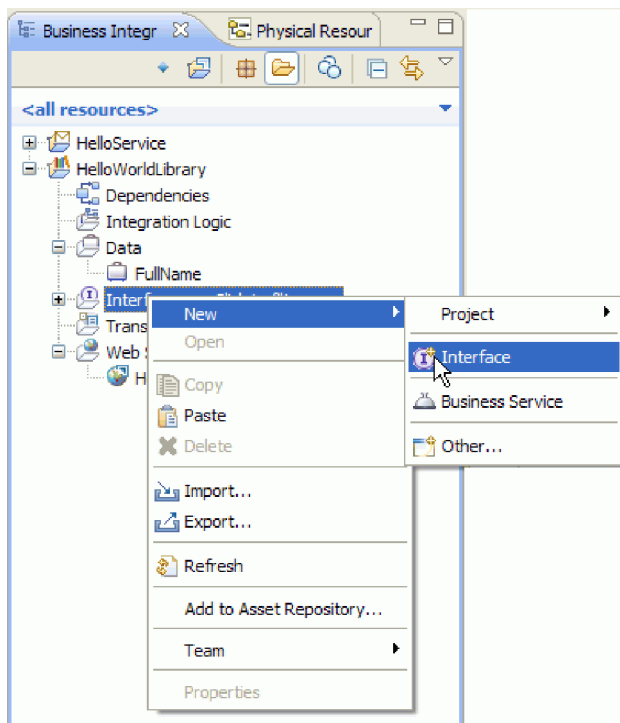
**オプション:** 「型」列で、「string」セルの 1 つを選択します。型のリストが表示されます。このサンプルでは、string 型のフィールドのみ必要ですが、ここで他の型も指定できます。Esc キーを押してリストを閉じます。

**オプション:** フィールドを選択して、エディターの下方の「プロパティ」ビューを調べます。このサンプルでは、これらのフィールドを設定する必要はありませんが、ここでフィールドの繰り返しや最大長など、フィールドの特定のプロパティを指定できます。

6. **Ctrl+S** を押して作業を保管し、ビジネス・オブジェクト・エディターを閉じます。

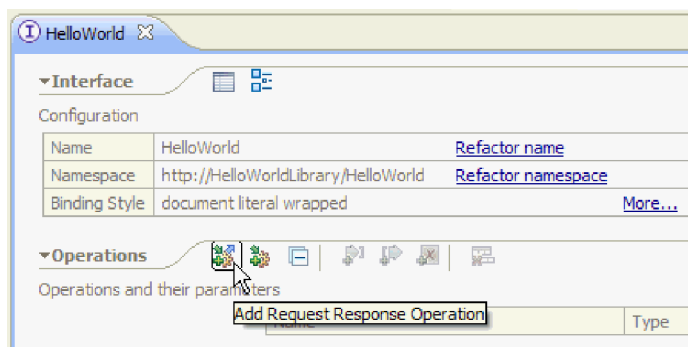
**オプション:** 内部的に、複合型を持つ新しい XSD スキーマ・ファイルまたは XML スキーマ・ファイルを作成したところです。このファイルを見たい場合は、FullName ビジネス・オブジェクトを右クリックして、「アプリケーションから開く」>「XML スキーマ・エディター」を選択し、次に「ソース」タブを選択して、ファイルを表示できます。

7. 以下に示すように、「ビジネス・インテグレーション」ビューに戻って、「HelloWorldLibrary」で「インターフェース」カテゴリを右クリックして、「新規」>「インターフェース」を選択します。

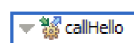


「新規インターフェース」ウィザードが開きます。

8. 「名前」フィールドに HelloWorld と入力して、「終了」をクリックします。インターフェース・エディターが開きます。
9. 要求/応答操作を追加するには、ローカル・ツールバーで「要求/応答操作の追加」アイコンをクリックするか、または右クリックして「要求/応答操作の追加」を選択します。

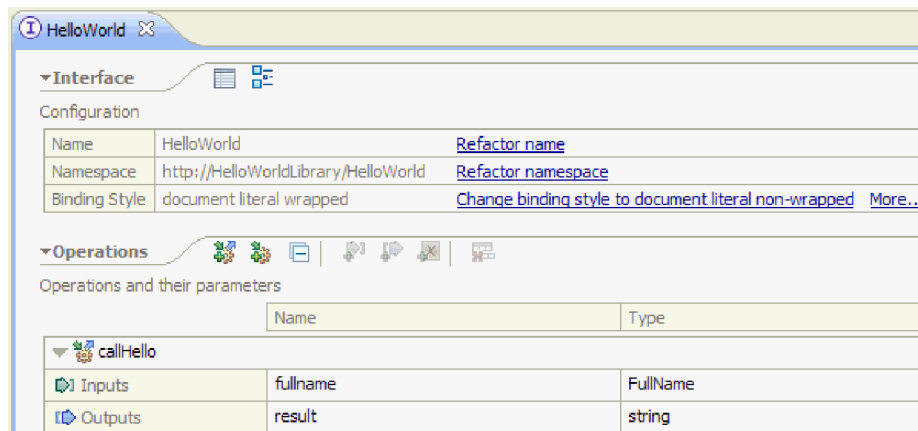


10. 以下に示すように、生成された操作名「operation1」をダブルクリックし、callHello で上書きします。





11. 生成されたパラメーター名「input1」をダブルクリックし、fullname で上書きします。
12. テーブルの **string** 型のセルをクリックし、「入力」行で型を変更します。ポップアップ・リストで、下部にスクロールし、先程作成したビジネス・オブジェクト「FullName」を選択します。
13. 生成されたパラメーター名「output1」をダブルクリックし、result で上書きします。インターフェースは、以下のようになります。



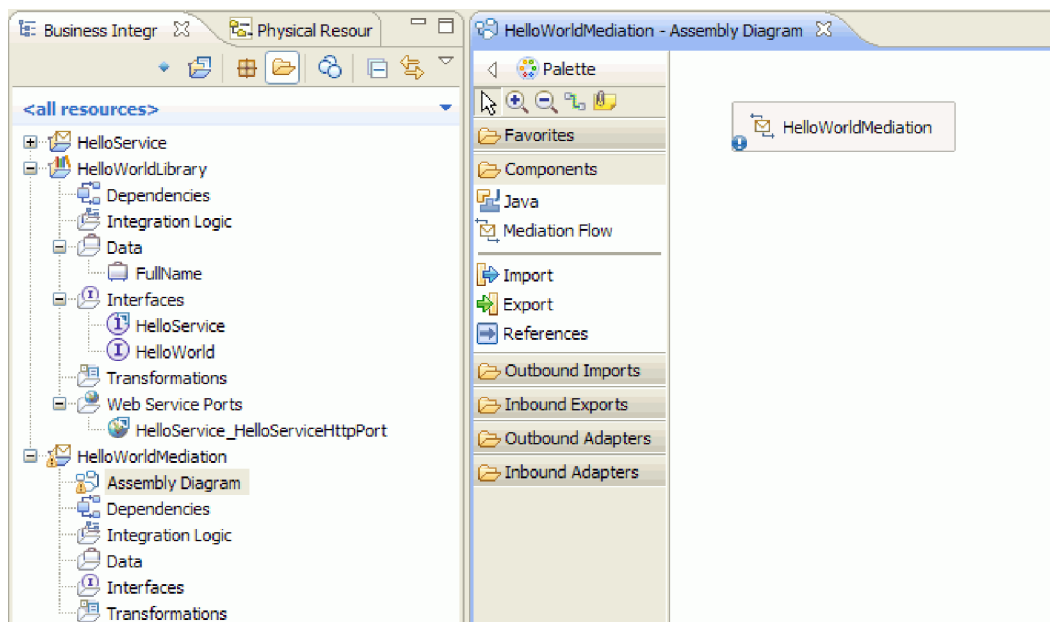
14. インターフェース・エディターを保管して閉じます。

## ライブラリーを参照するメディエーション・モジュール・プロジェクトの作成

メディエーション・モジュールは、サービス統合ロジックを含むプロジェクトであり、IBM Process Server または WebSphere Enterprise Service Bus にデプロイできます。

メディエーション・モジュールを使用して、提供された **HelloService** Web サービスにアクセスし、それをサービスとして他のモジュールに公開します。呼び出すサービス (HelloService) のインターフェースは公開するサービスのインターフェース (HelloWorld) とは異なるため、その間でマップを行う必要があります。

1. 「HelloWorldLibrary」ライブラリーを右クリックし、「新規」>「プロジェクト」>「メディエーション・モジュール」を選択します。「新規メディエーション・モジュール」ウィザードが表示されます。
2. 「モジュール名」フィールドで、HelloWorldMediation と入力して「次へ」をクリックします。「必要なライブラリーの選択」ページが開きます。
3. 「HelloWorldLibrary」プロジェクトが選択されていることを確認します。この選択によってライブラリーがこの新しいモジュールに関連付けられ、このモジュールが、そのライブラリー内の任意の成果物を使用できるようになります。「終了」をクリックします。新しいモジュール用にアSEMBリー・エディターが開き、以下に示すように、メディエーション・フロー・コンポーネントが含まれています。



## メディエーション・モジュールの組み立て

次に、他のモジュールからメディエーション・フロー・コンポーネントを呼び出すためのエクスポートを作成し、さらに HelloWorld サービスを呼び出すためのインポートを作成します。エクスポート、メディエーション・フロー、およびインポートを一緒にワイヤリングし、デPLOY可能なモジュールを作成します。

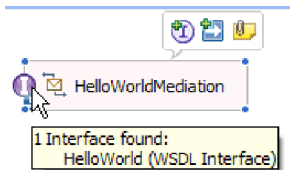
メディエーション・モジュールを組み立てるには、以下の手順を実行します。

1. **HelloWorldMediation** コンポーネントは、最終的にはメディエーション・フローによって実装されますが、まず他のコンポーネントから呼び出すことができるようにインターフェースを提供する必要があります。コンポーネントを選択すると、以下に示すように、そのコンポーネントの上に吹き出しバーが表示されることに注意してください。

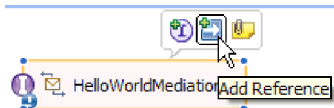


**注:** 誤ってコンポーネントをダブルクリックすると、「開く」ウィンドウで、実装を作成するかどうかを尋ねてきます。このウィンドウが開いた場合には、「いいえ」をクリックして閉じてください。

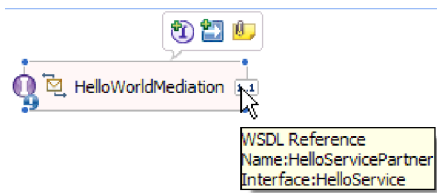
吹き出しバーから円で囲まれた「I」をクリックして、インターフェースを作成します (またはコンポーネントを右クリックして、「追加」>「インターフェース」を選択します)。「インターフェースの追加」ウィンドウが開きます。先程作成したインターフェースである「HelloWorld」を選択します。「OK」をクリックします。これで、以下の図に示すように、コンポーネントの左端に円で囲まれた「I」が表示され、吹き出しヘルプにインターフェース・タイプが表示されます。



- これで、このメディエーションを呼び出すことができますが、メディエーションが呼び出すサービスを識別する必要もあります。もう一度コンポーネントを選択しますが、今回は、以下の図に示すように、吹き出しバーから右矢印を選択して参照を追加します。



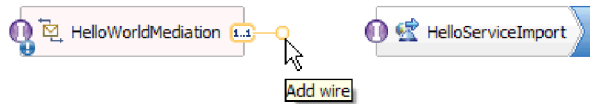
「参照の追加」ウィンドウが開きます。「一致するインターフェース」リストで、提供されている呼び出し対象のサービスのインターフェースである「**HelloService**」を選択します。「**OK**」をクリックします。ここで、以下の図に示すように、内部に「1..1」と表示された小さなボックスがコンポーネントの右端に表示されます。これは、このコンポーネントが呼び出すことを宣言した必須サービスを示しています。



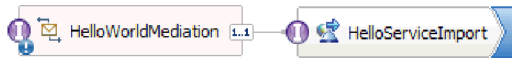
呼び出される実際のサービスがまだ識別されていません。この時点で分かっているのは、そのサービスのインターフェースまたは形態のみです。（「1..1」は参照の多重度であり、この参照には、別のコンポーネントに対して正確に 1 つのワイヤーで対応する必要があることを意味しています。参照は、複数のワイヤーを許可するように構成することもできます。）

これは、SCA (Service Component Architecture) の優れた点です。実装の詳細に関係なく、アプリケーションは一連のコンポーネントとして定義され、これらのコンポーネントはインターフェースを公開し、参照を通して他のコンポーネントまたはサービスを取り込みます。

- ここで、ライブラリーにエンドポイント WSDL をコピーしてある、提供された Web サービス **HelloService** を呼び出すために、インポート・コンポーネントを追加する必要があります。「ビジネス・インテグレーション」ビューで、「**HelloWorldLibrary**」を展開します。「**Web サービス・ポート**」カテゴリーで、ポート (HelloService\_HelloServiceHttpPort) を、アセンブリ・ダイアグラム・キャンバス内の任意の位置にドラッグします。
- 新規インポートの名前をクリックして編集モードに入り、名前を **HelloServiceImport** に変更します (つまり、接尾部「1」を削除します)。
- 以下の図に示すように、黄色のボーダーとその右に黄色の円が表示されるまで、**HelloWorldMediation** コンポーネントの右ボーダーの参照ボックス上にカーソルを合わせます。



以下の図に示すように、マウスで円をグラブし、**HelloServiceImport** コンポーネントの左端の円で囲まれた「I」にドラッグして 2 つのコンポーネントを一緒にワイヤリングします。



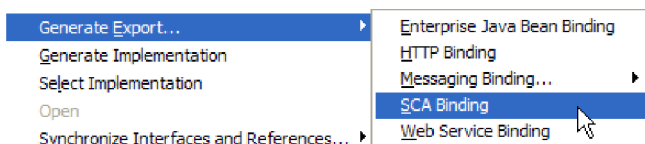
注: キャンバスで右クリックすると、「自動レイアウト」設定を切り替え、キャンバスのノードが自動で整列されるようにすることができます。

これで、最初のコンポーネントの参照を、サービスの実際のプロバイダー (この場合は外部 Web サービス) で解決したことになります。これは、最初のコンポーネントの実装がその参照を呼び出すと、実際に Web サービス HelloService が呼び出されることを意味しています。(すぐに、最初のコンポーネントの実装を作成します。)

注: Web サービス・インポートを即時作成する方法の 1 つは、「インポート」をパレットからアセンブリー・ダイアグラムにドラッグした後、構成することです。この方法でインポートを作成し、他のタイプのサービス呼び出します。サービスには、ネイティブ SCA バインディングによって呼び出されるもの、HTTP、JMS、または MQ によってメッセージを送信することにより呼び出されるもの、リモート・エンタープライズ Java Bean を呼び出すことにより呼び出されるものなどがあります。これらの組み込みインポート・バインディングを超えて、提供されたアダプターを使用して J2EE コネクター・アーキテクチャ (J2C) 標準によって外部サービス呼び出して、ファイルへの書き込みや E メール送信などを行うことができます。これらのアダプターのいくつかによってアクセス可能なサービスは、Enterprise Integration Service (EIS) と呼ばれています。

**オプション:** パレットで、「アウトバウンド・アダプター」カテゴリと「アウトバウンド・インポート」カテゴリを展開して、使用可能な外部サービスを確認します。

- Web サービスを呼び出すほかに、このメディエーションを公開して、「Hello World パート 2」に備えて他のモジュールから呼び出せるようにしたい場合があります。これは、サポートの必要な唯一のクライアントなので、SCA エクスポート・バインディングを使用します。(他のモジュールは、モジュール内の一致する SCA インポート・コンポーネントを使用して、SCA エクスポート・コンポーネントを呼び出すことができます。) 以下に示すように、アセンブリー・ダイアグラム・キャンバスで、「HelloWorldMediation」コンポーネントを右クリックし、「エクスポートの生成」>「SCA バインディング」を選択します。



このアクションにより、**HelloWorldMediation** コンポーネントにワイヤリングされた、SCA バインディング付きの新しい **HelloWorldMediationExport** エクスポート・コンポーネントが作成されます。これで、コンポーネントに、このモジュールの外部からアクセスできます。



注: SCA は、コンポーネントを公開して、モジュールの外部から呼び出せるようにする唯一の方法です。「エクスポートの生成」カスケード・メニューに表示されるように、その他に HTTP、JMS、MQ、および Web サービスがあります。インポートと同様に、これらは、すべてネイティブにサポートされる組み込みバインディングですが、さらに J2C 標準を使用してモジュールを呼び出すためのアダプターが提供されています。

**オプション:** パレットで、「インバウンド・アダプター」カテゴリと「インバウンド・エクスポート」カテゴリを展開して、モジュールを呼び出すためにサポートされている他の方法 (E メール受信や、フラット・ファイルに表示される内容など) を確認します。

7. **Ctrl+S** を押して、アセンブリー・ダイアグラムでの作業を保管します。

---

## メディエーション・フロー実装の作成

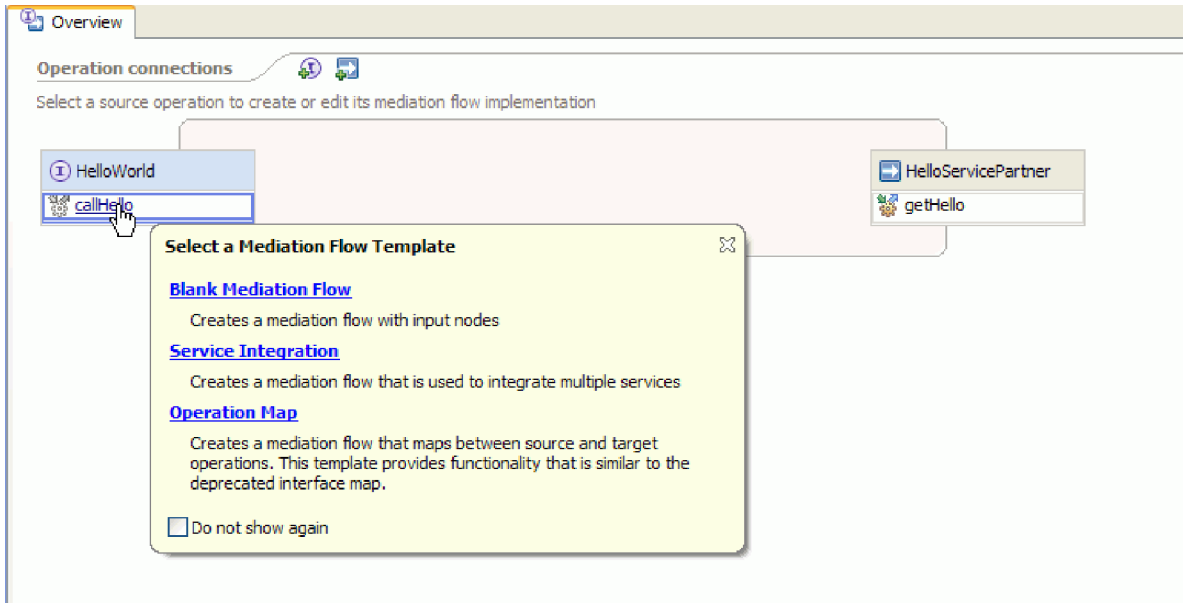
最後に、HelloWorldMediation コンポーネントの実装を作成します。このコンポーネントはメディエーション・フローであり、メディエーション・モジュールの作成時に作成されました。ただし、パレットからメディエーション・フローをドラッグして作成することもできます。コンポーネントがこのタイプであるため、メディエーション・フロー・エディターを使用して実装します。

メディエーション・フロー実装を作成するには、以下の手順を実行します。

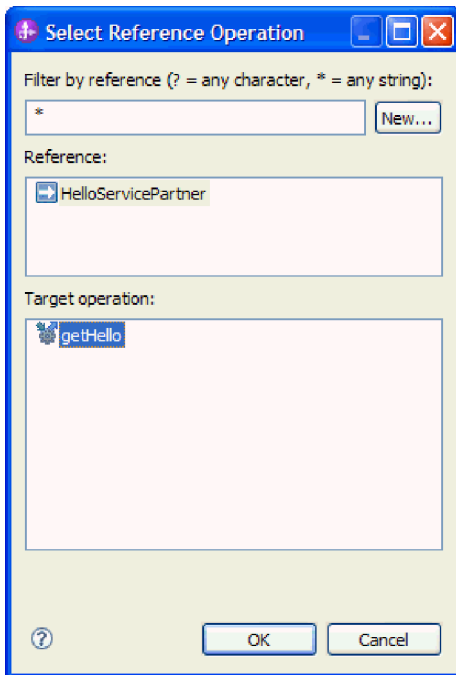
1. アセンブリー・エディターで、「**HelloWorldMediation**」コンポーネントをダブルクリックして「開く」ウィンドウで「はい」をクリックし、「実装の生成」ウィンドウで「OK」をクリックします。メディエーション・フロー・エディターが開きます。

注: 開発アクティビティーのある時点で、開発に関する判断を行うのに役立つ「ヒント」ダイアログ・ボックスが開く場合があります。このサンプルでは、「ヒント」ダイアログ・ボックスを閉じ、サンプルの説明に従ってください。

2. メディエーション・フロー・エディターの上部で、左側に **callHello**、右側に **getHello** と表示されています。これは、それぞれ、このコンポーネントのインターフェースからの単一の操作、およびこのコンポーネントの参照からの単一の操作となります。(各コンポーネントに対して複数のインターフェースおよび参照を持つこと、および各インターフェースと参照に対して複数の操作を持つことも有効なことに注意してください。しかし、これは *Hello World* です。) **callHello** インターフェース操作を選択します。操作間でシンプル・マップを実行するメディエーション・フローを作成するのかどうかを選択する必要があります。ただし、必要な変更はメディエーション・フロー・エディターで行うことができるため、この選択は固定されたものではありません。このサンプルでは、操作間のシンプル・マップを実行するため、以下の画面キャプチャーに示されている「**操作マップ**」リンクを選択する必要があります。

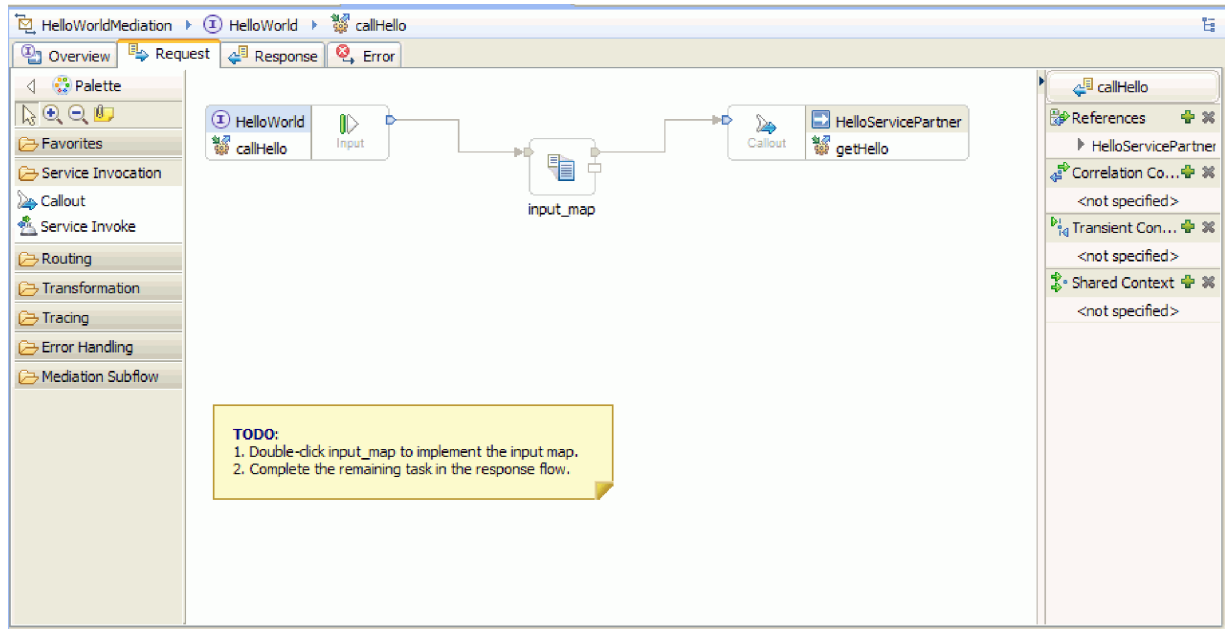


- 「参照操作の選択」ダイアログ・ボックスが開いたら、**getHello** 操作（下図参照）を選択して、「OK」をクリックします。



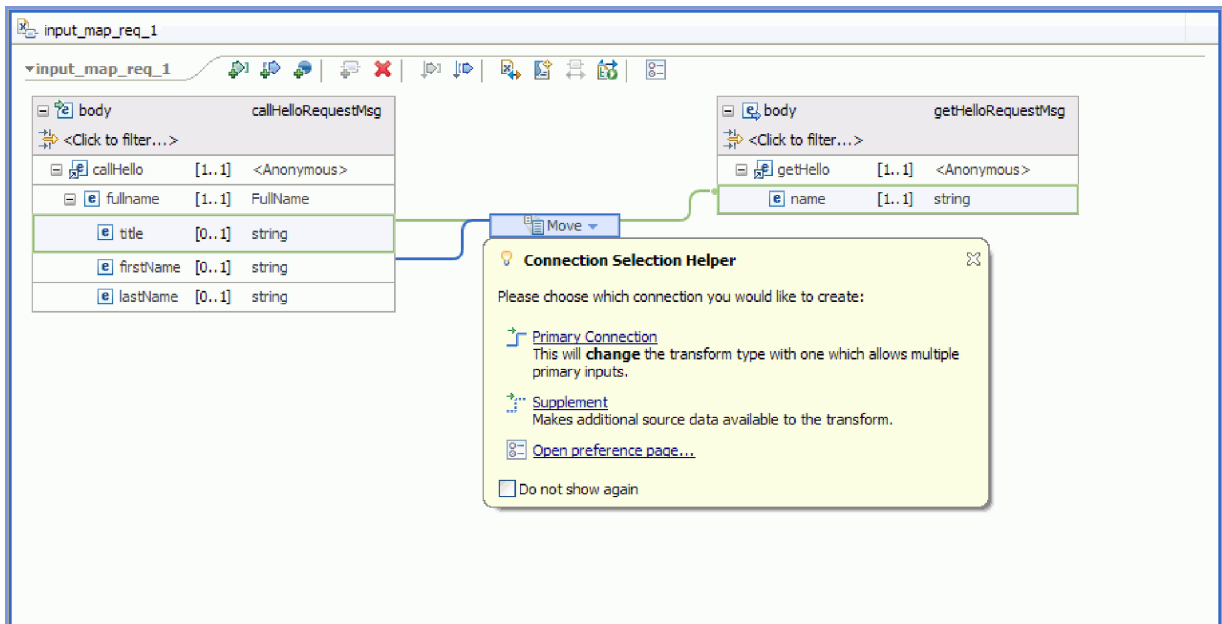
ここでは、このコンポーネントに対する **callHello** 操作の実装の一部として、**getHello** 操作を呼び出すことが示されています。

- 次に、**callHello** 操作に対するフローの実装を完了します。これは要求/応答操作であるため、要求に対して 1 つのフロー、応答に対してもう 1 つのフローが用意されますが、まず要求から開始します。以下に示すように、キャンバスの上部で **callHello** の「要求」タブをクリックします。

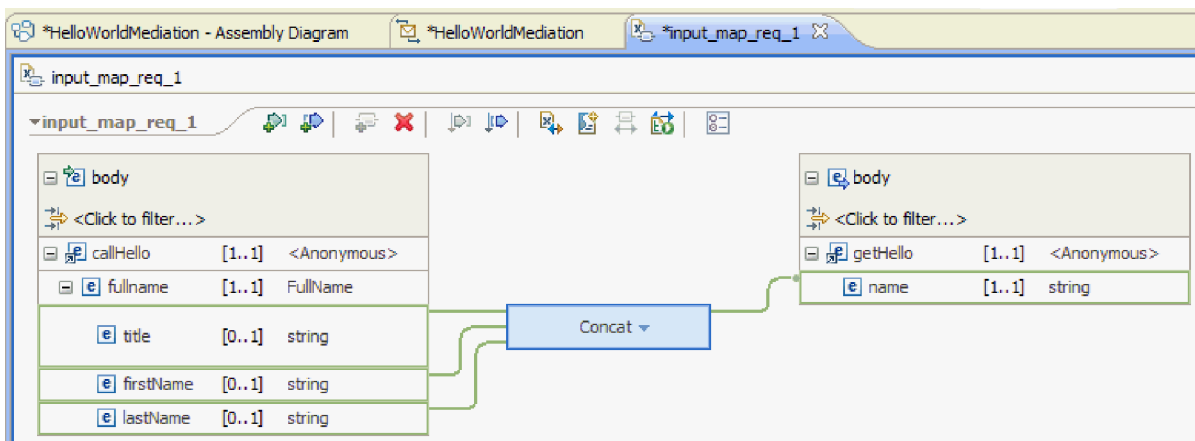


フロー領域で、左側に入力ノードが表示されており、callHello 操作の呼び出しにより要求フローに到達する制御を表しています。次いで、フローは、getHello 操作と callHello 操作の入力ビジネス・オブジェクト間をマッピングする input\_map XSL 変換を呼び出します。次に、フローは getHello 参照操作（この場合は外部 Web サービス）を呼び出します。要求のフロー領域内に、フローを完全に実装するために実行する必要があるタスクが書かれた付せんがあります。

5. 「input\_map」プリミティブをダブルクリックして、そのマップを作成します。「新規 XML マップ」ウィンドウが開きます。「終了」をクリックします。XML マップ・エディターが開きます。
6. XML マップ・エディターで、左右のツリーを完全に展開します。
7. 左側の「title」を右側の「name」にワイヤリングします。これで、「Move」タイプの変換が作成されます。
8. 左側の「firstName」を中央の「Move」変換にワイヤリングします。「接続選択ヘルパー」が表示されます。

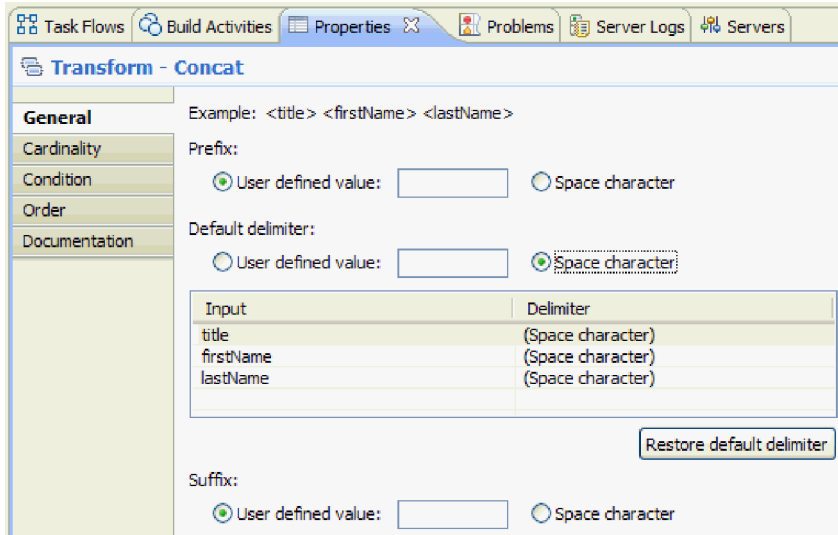


9. 「1 次接続」を選択します。これで、操作が「Concat」操作に変更されます。
10. 左側の「lastName」を中央の「Concat」操作にワイヤリングします。「接続選択ヘルパー」が再び表示されます。「1 次接続」を選択します。以下に示すように、これで 3 本の着信のワイヤーと 1 本の発信のワイヤーが表示されます。

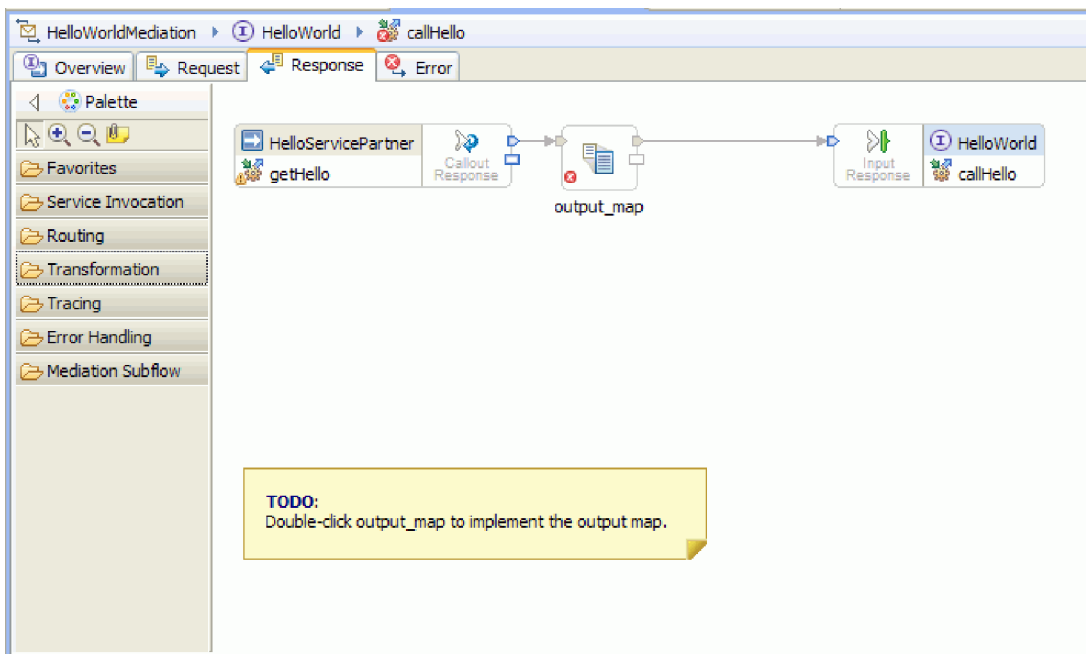


11. 「Concat」操作を選択し、「プロパティ」ビューに移動して「一般」タブをクリックします。
12. フィールドが連結されている場合は、タイトル、名、および姓の間にスペースを入れることができます。「プロパティ」ビューの「一般」タブの区切り文字設定を使用して、これらのスペースを作成します。デフォルトの区切り文字には、「スペース文字 (Space character)」を選択してください。区切り文字は、下図に示すように、表内に「(スペース文字) ((Space character))」として表示されます。

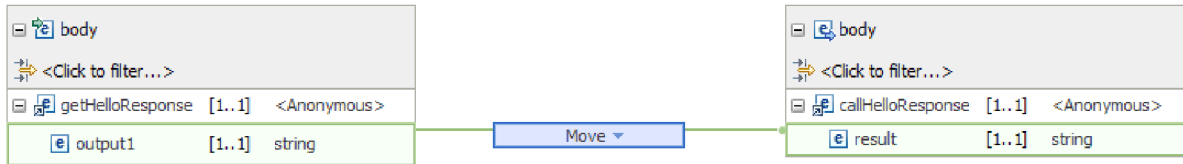




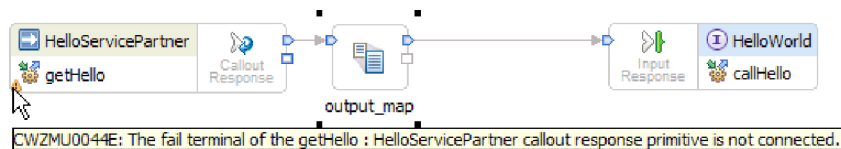
13. XML マップ・エディターを保管して閉じます。
14. メディエーション・フロー・エディターを保管します。
15. ここで、メディエーション応答 フローの実装を完了する必要があります。これにより、要求フローで呼び出した Web サービスからの応答を処理して、このコンポーネントの呼び出し元への応答に変換します。以下に示すように、フロー・エディターの上で「**Response**」タブを選択します。



16. 「**output\_map**」プリミティブをダブルクリックします。「新規 XML マップ」ウィザードで「終了」をクリックして、新規マップを作成します。
17. 以下の図に示すように、着信の「**output1**」フィールドを発信の「**result**」フィールドにマップします。

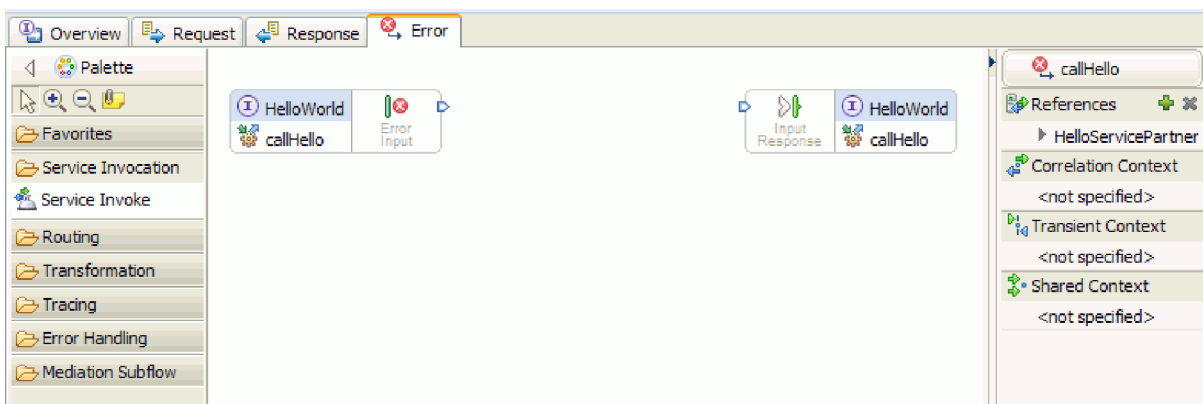


18. XML マップ・エディターを保管して閉じます。
19. メディエーション・フロー・エディターを保管します。
20. 最後に、以下の図に示すように、コールアウト応答ノードの障害ターミナルにワイヤリングしていないことを示す警告を除去します。

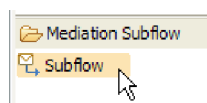


この警告は、Web サービスの呼び出しが失敗する状況に対応していないために発生します。

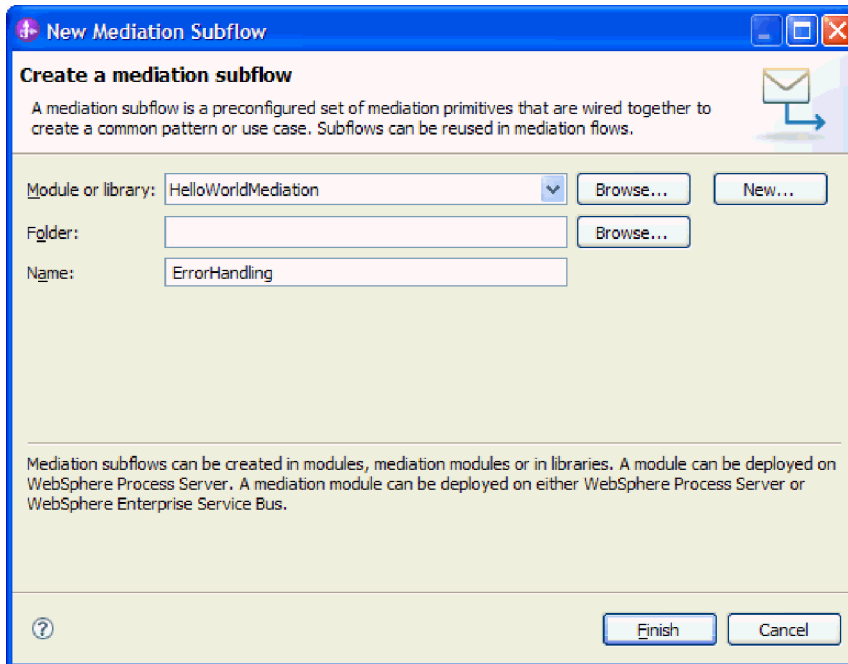
21. 要求フローまたは応答フロー時に発生する可能性のある未処理エラーを取り込むには、エラー・フローを実装します。以下に示すように、フロー・エディターの上にある「エラー」タブをクリックします。



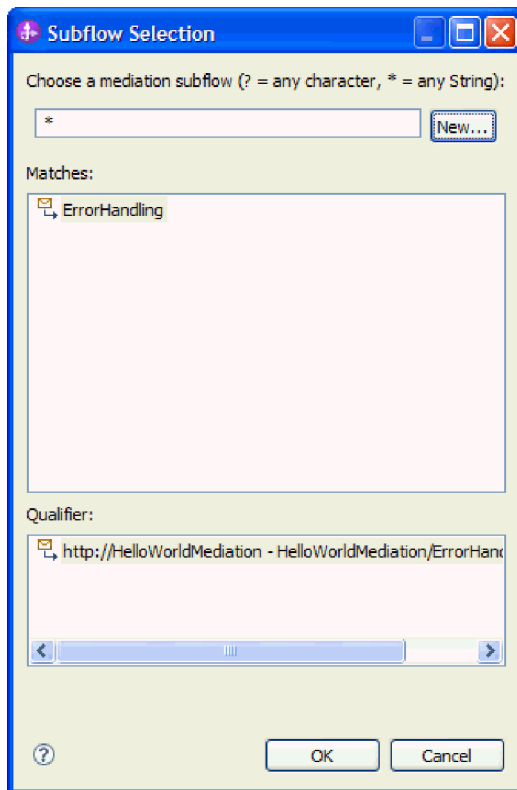
22. エラー処理をカプセル化し、より簡単にロギングを追加するには、以下のステップを実行してサブフローを追加してください。
  - a. 下図に示すように、パレットで「メディエーション・サブフロー」をクリックして、「サブフロー」をクリックします。



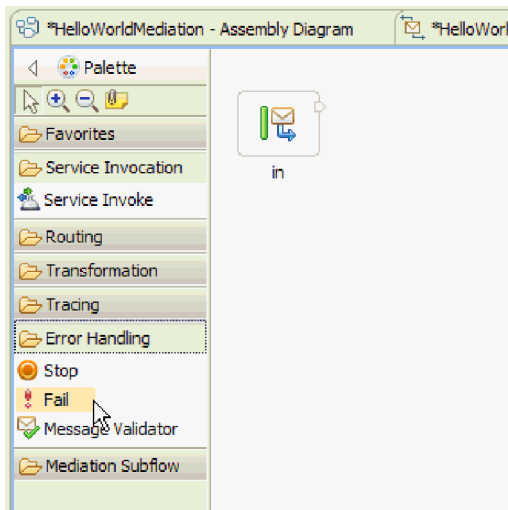
- b. 「サブフロー」をパレットからキャンバスにドラッグします。「サブフローの選択」ダイアログ・ボックスが開きます。
  - c. 「新規」ボタンをクリックします。「新規メディエーション・サブフロー」ウィザードが開きます。
  - d. 下図に示すように、「名前」フィールドに `ErrorHandling` と入力します。



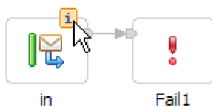
- e. 「終了」をクリックします。
- f. 下図に示すように、「サブフローの選択」ダイアログ・ボックスで **ErrorHandling** を選択します。



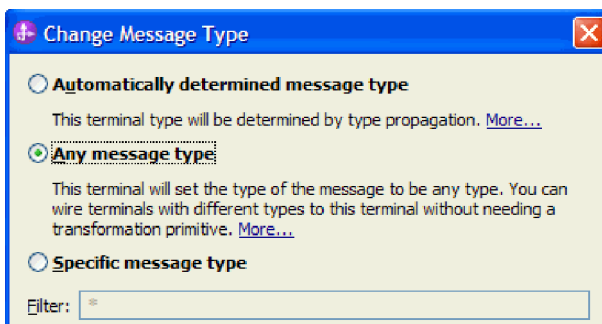
- g. 「OK」をクリックして、「サブフローの選択」ダイアログ・ボックスを閉じます。ErrorHandling サブフローが開きます。
- h. 下図に示すように、新規サブフローのパレットで「エラー処理」フォルダーをクリックして、「障害」プリミティブをクリックします。



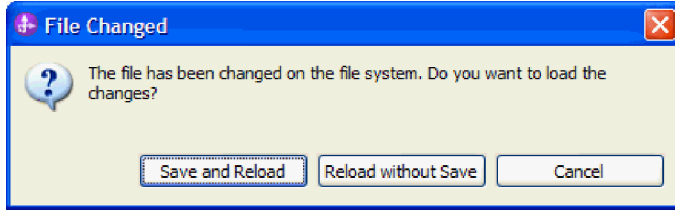
- i. 「障害」プリミティブをパレットからキャンバスにドラッグします。
- j. 下図に示すように、「in」ターミナルの右端を「障害」プリミティブにワイヤリングします。



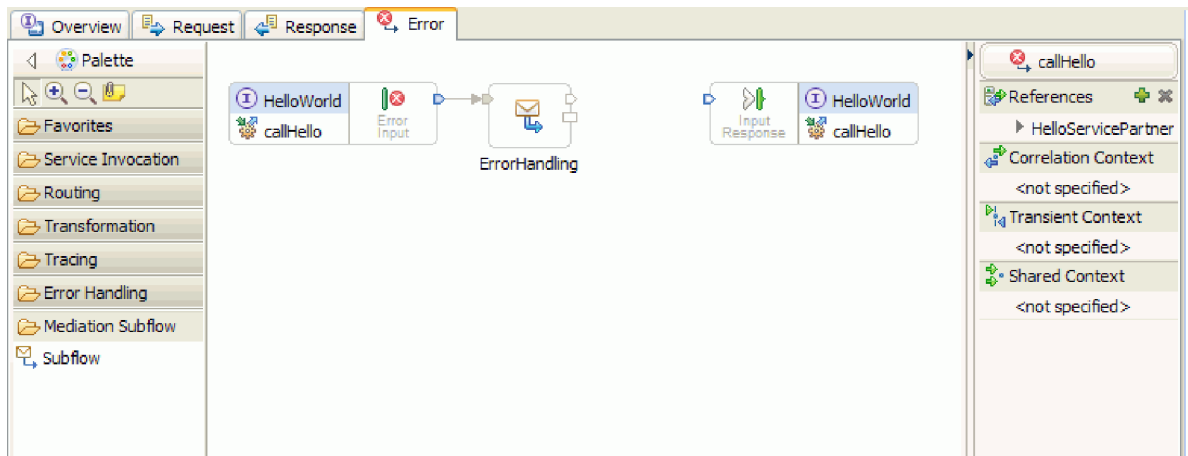
- k. フローから出るメッセージはないため、「out」ターミナルを右クリックして「削除」を選択します。
- l. ここで、入力メッセージ・タイプを「**ANY メッセージ・タイプ**」に変更する必要があります。「in」ターミナルの上にカーソルを移動して、ターミナルの上端に表示される「i」アイコンをクリックします。「in」ダイアログ・ボックスが開きます。
- m. このダイアログ・ボックスで「サービス・メッセージ・オブジェクトの詳細」を展開して、「変更」をクリックします。「メッセージ・タイプの変更」ダイアログ・ボックスが開きます。
- n. 下図に示すように、「**ANY メッセージ・タイプ**」ラジオ・ボタンを選択します。



- o. 「OK」をクリックして「メッセージ・タイプの変更」ダイアログ・ボックスを閉じてから、「in」ダイアログ・ボックスを閉じます。
- p. **Ctrl-S** を押してサブフローを保存してから、サブフローを閉じます。以下に示すように、「ファイルが変更されました」ダイアログ・ボックスが表示されます。



- q. 「保存して再ロード」を選択します。これで、HelloWorldMediation フロー内の ErrorHandling サブフローに対して行った変更が再ロードされます。
- r. 「エラー」タブに切り替えます。「エラー入力」の出力ターミナルを ErrorHandling サブフローにワイヤリングします。



- s. 「入力応答」は、サービス要求元にメッセージを送り返すために使用します。ここでは「入力応答」を使用しないため、入力応答を選択して非表示にできます。キャンバスで右クリックして、「入力応答の非表示」を選択します。
23. これで要求フローと応答フローの TODO が完了したので、「要求」タブと「応答」タブの両方で、キャンバス上の TODO ダイアログ・ボックスを右クリックして「削除」を選択し、TODO ダイアログ・ボックスを削除します。
24. メディエーション・フロー・エディターとアセンブリー・エディターを保管して閉じます。

おめでとうございます。これで、オーサリング手順が完了しました。今度はテストを行います。



---

## 第 3 章 サンプルの実行

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。また、アダプターのインポートに示すように、WebSphere Adapter for Flat Files もインポートする必要があります。

サンプルを実行するには、以下の手順に従います。

- 必要であれば、このサンプルの PDF 版を印刷します。既にサンプルをインポートしてあり、それを実行したい場合もあります。PDF 版を見ると、サンプルがどのようにして作成されたかが分かります。
- 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。

---

### モジュールのデプロイ

ランタイムでアプリケーションをテストできるよう、モジュールをサーバーにデプロイします。

アプリケーションがインバウンド処理モードで暗黙的に開始されたために、テストはモジュールがデプロイされるとすぐに始まります。

モジュールをサーバーにデプロイするには、以下の手順を実行します。

1. 「サーバー」をクリックして「サーバー」ビューを開きます。
2. 「IBM Process Server」を右クリックし、コンテキスト・メニューから「開始」を選択します。
3. サーバーが「開始済み」ステートに入った後、「IBM Process Server」を右クリックします。コンテキスト・メニューから、「追加と削除」を選択します。「使用可能プロジェクト」から、「SeparateCustomersApp」を選択します。「追加」をクリックします。「完了」をクリックします。
4. 前述のように、アプリケーションがインバウンド処理モードであるために、実行時のモジュールのテストはこの時点で開始されます。予期される出力については、モジュールのテストを参照してください。

---

### モジュールのテスト


次のタスクでは、デプロイしたばかりのモジュールの実行とテストを行います。統合テスト・クライアントを使用して、サンプル・データを提供し、その結果を表示することで、モジュールをテストします。

モジュールをテストするには、以下の手順を実行します。

1. 「ビジネス・インテグレーション」ビューで、「HelloWorldMediation」を展開して、「アセンブリー・ダイアグラム」をダブルクリックします。アセンブリー・ダイアグラムが開きます。
2. エクスポート・コンポーネント **HelloWorldMediationExport** を右クリックして、「コンポーネントのテスト」を選択します。統合テスト・クライアントが開きます。
3. エクスポート・コンポーネントに渡されるパラメーターのサンプル値を入力します。これを行うには、値エディター・テーブルの「値」列にあるセルを編集することによって、ビジネス・オブジェクトのフ

フィールドごとに値を指定します。値エディター・テーブル (テスト・クライアントの右下隅に配置) の「値」列で、セル (セルを入力してから) をダブルクリックして編集モードに入り、次に「title」に Mr、「firstName」に Phil、「lastName」に Bar と入力します。**ヒント:** 入力した後、下矢印をクリックするか、**Enter** キーを押します。以下の図に、値エディターを示します。

Name	Type	Value
fullname	FullName	✓
title	string	✓ Mr
firstName	string	✓ Phil
lastName	string	✓ Bar

4. テスト・クライアント内のイベント・リストの上部で、「**続行**」アイコン  をクリックします。「デプロイメント・ロケーション」ダイアログ・ボックスが開きます。
5. 「デプロイメント・ロケーション」ダイアログ・ボックスに複数のサーバーがリストされている場合、元の IBM Process Server サーバー以外のサーバーを選択すると、テストの結果が例外になる可能性があります。これは、選択したサーバーの HTTP ポート番号が、HelloServiceImport バインディングで指定されたデフォルト・ポート番号 (9080) と一致しない場合があるためです。目的のサーバーのポート番号を確認し、(必要であれば) その番号と一致するようにインポート・バインディングのポート番号を変更するには、以下の手順を実行します。

- a. ファイル・システムで、次のフォルダーに移動します (*installDir* は IBM Integration Designer の WebSphere テスト環境のインストール・パスで、*serverProfile* はサーバー・プロファイルの名前です)。

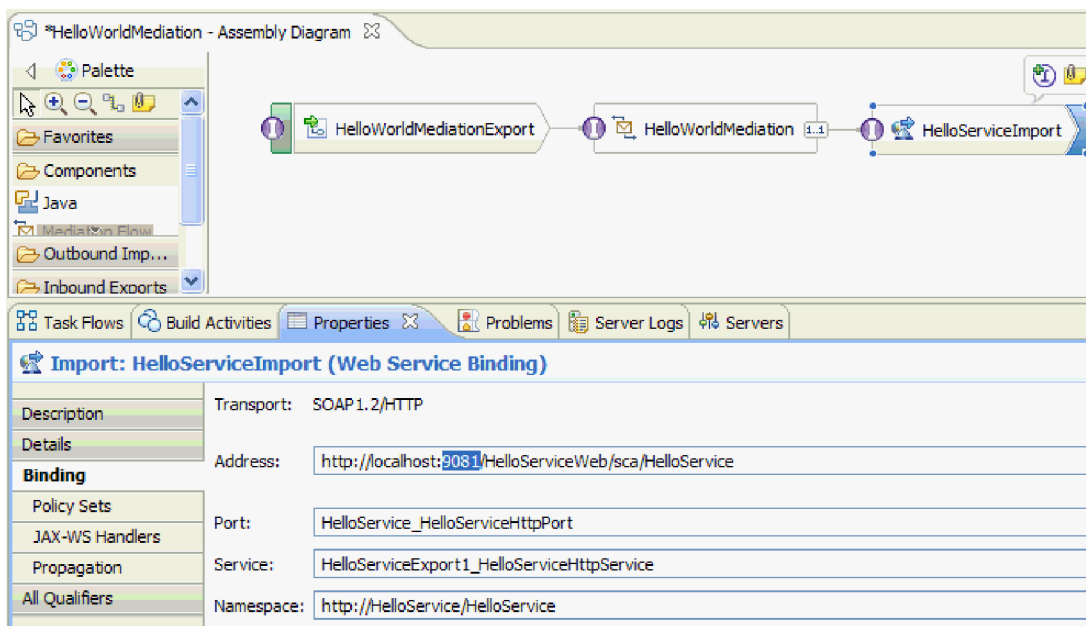
```
installDir\runtimes\bi_v7\profiles\serverProfile\logs
```

以下に例を挙げます。

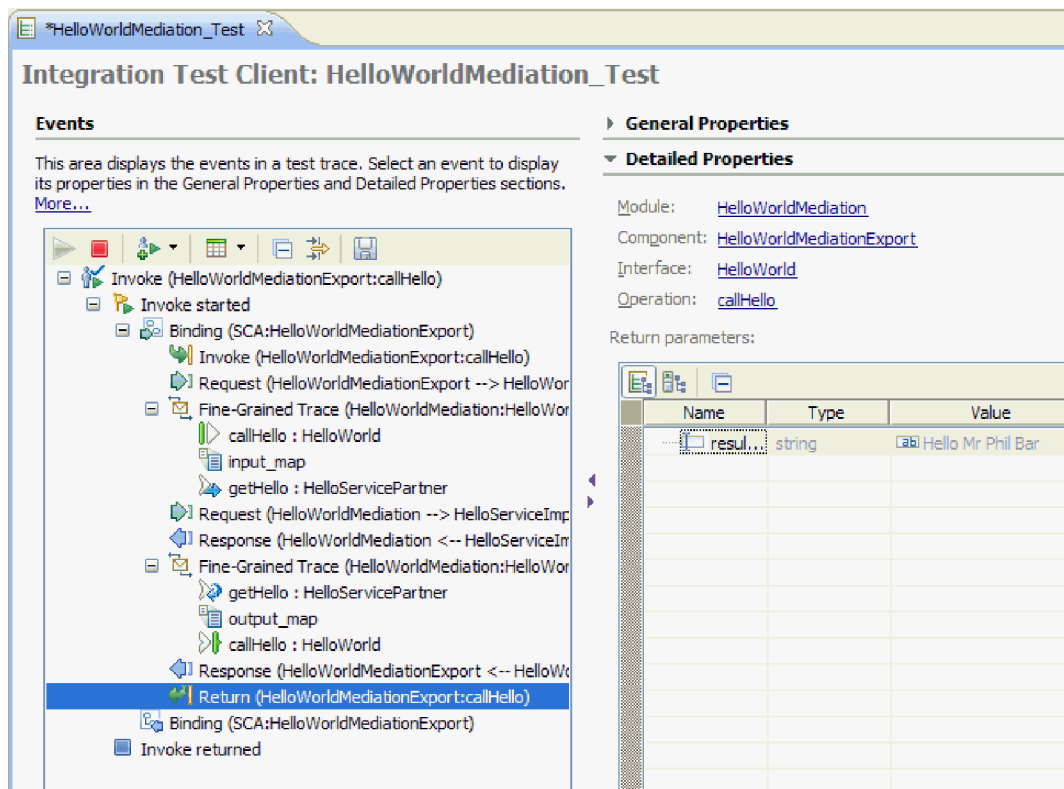
```
C:\Program Files\IBM\WID7_WTE\runtimes\bi_v7\profiles\qwps\logs
```

- b. テキスト・エディターでファイル **AboutThisProfile.txt** を開きます。
- c. ファイル内で、**HTTP** トランスポート・ポート番号を見つけます。HTTP トランスポートのポート番号が 9080 ではない場合は、以下の手順を実行して、インポート・バインディングのポート番号を、HTTP トランスポート・ポートのポート番号と一致するように変更します。
- d. AboutThisProfile.txt ファイルを閉じます。
- e. テスト・クライアントで、「**キャンセル**」を押して「デプロイメント・ロケーション」ダイアログ・ボックスを閉じます。
- f. テスト・クライアントを閉じ、変更点を保存するかどうかを尋ねるプロンプトが表示されたら「**いいえ**」をクリックします。
- g. 「ビジネス・インテグレーション」ビューで、「**HelloWorldMediation**」メディアエーション・モジュールを展開して、「**アセンブリー・ダイアグラム**」をダブルクリックします。アセンブリー・エディターでアセンブリー・ダイアグラムが開きます。
- h. アセンブリー・ダイアグラムで、「**HelloServiceImport**」インポートを選択します。
- i. 「**プロパティ**」タブをクリックし、「**バインディング**」タブをクリックします。「バインディング」ペインが開きます。
- j. 「バインディング」ペインの「**アドレス**」フィールドで、次の図に示すように、インポート・バインディングのポート番号を HTTP トランスポート・ポートのポート番号と一致するように変更します。

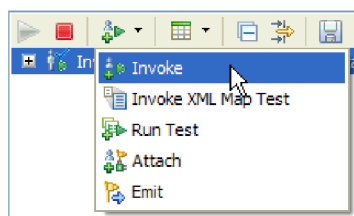




- k. **Ctrl+S** を押して変更内容を保管し、アセンブリー・エディターを閉じます。
1. 手順 1 から始めて、このトピックでの指示を繰り返し実行します。
6. 「デプロイメント・ロケーション」ダイアログ・ボックスで、正しいサーバーが選択されていることを確認し、「終了」をクリックします。「ユーザー・ログイン」ウィンドウが開きます。
7. インストール中に、サーバーのデフォルトのユーザー ID とパスワードを変更しなかった場合、「**OK**」をクリックします。変更した場合には、インストール中に指定したユーザー ID とパスワードを入力して「**OK**」をクリックします。サーバー上で実行されるテスト・クライアント・コードが開始され、必要な場合には、変更されたモジュールが公開され、さらにテストが実行されます。「イベント」リストにイベントが表示され、アセンブリー・ダイアグラム内のコンポーネントを通して進行する実行が示され、またメディエーションの要求と応答のフロー内のプリミティブを通して進行する実行の詳細イベントが示されます。以下に示すように、結果としてストリング「Hello Mr Phil Bar」が返されます。



8. オプション: テストを続行できます。以下に示すように、「イベント」リストの上のツールバー内で、3番目のアイコンの横にある小さな下矢印アイコンを選択して、「呼び出し」を選択します。



イベント・リストに新しい「呼び出し」イベントが表示され、また「初期要求パラメーター」値エディターに、そのテスト用のオリジナル入力データが表示されます。Bar を BarAgain に変更し、この場合も「続行」ボタンをクリックして、テストを再実行します。

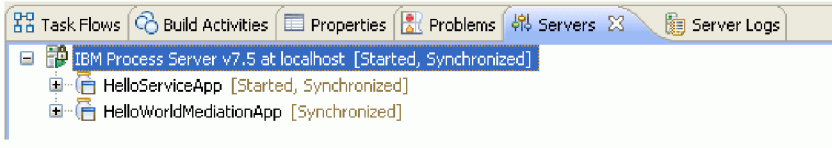
9. 「ファイル」>「すべて閉じる」を使用して、開いているすべてのエディターを閉じます。テスト・クライアント・セッションの保管を求めるプロンプトが出されたら、「いいえ」をクリックします。

## サーバーからのモジュールの除去

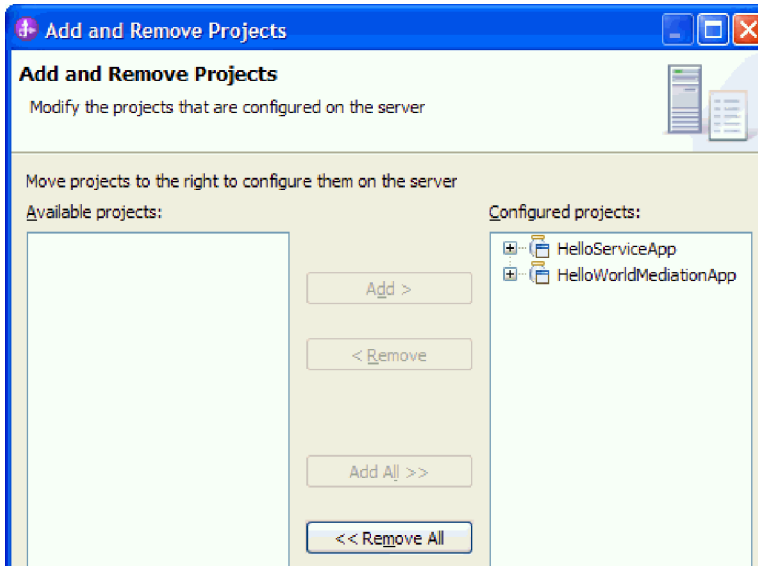
一般に、モジュールのテストが済んだら、サーバーからモジュールを除去します。これにより、サーバーにデプロイしたモジュールのみを対象としてテストの準備をすることになるため、サーバーの負荷が減り、パフォーマンスが向上します。

サーバーからモジュールを除去するには、以下の手順を実行します。

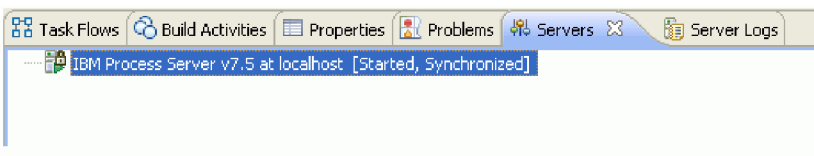
1. 「サーバー」タブをクリックします。以下の図に示すように、「サーバー」ビューが開きます。



2. 「サーバー」ビューで、「**IBM Process Server**」を右クリックして、「**プロジェクトの追加および除去**」を選択します。次の図に示すように、「プロジェクトの追加および除去」ダイアログ・ボックスが開きます。



3. 「すべて除去」をクリックします。「構成済みプロジェクト」リストから、アプリケーションが除去されます。
4. 「完了」をクリックします。サーバーからプロジェクトが除去されることを通知するダイアログ・ボックスが開いた場合、「**OK**」をクリックします。以下に示すように、「サーバー」ビュー内のサーバーの下にアプリケーションが表示されなくなります。



おめでとうございます。「Hello World パート 1: 入門」サンプルが完了しました。



---

## 第 4 章 インポート

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 7.5**」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。

〒242-8502

神奈川県大和市下鶴間1623番14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。

*Intellectual Property Dept. for WebSphere Software*

*IBM Corporation*

*3600 Steeles Ave. East*

*Markham, Ontario*

*Canada L3R 9Z7*

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

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

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

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っていません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

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

表示されている IBM の価格は IBM が小売り価格として提示しているもので、現行価格であり、通知なしに変更されるものです。卸価格は、異なる場合があります。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

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. \_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 (提供されている場合)、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](http://ibm.com) は、世界の多くの国で登録された International Business Machines Corp. 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、[www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml) をご覧ください。

Adobe、Adobe ロゴ、PostScript、PostScript ロゴは、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。

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。

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

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Intel Centrino、Intel Centrino ロゴ、Celeron、Intel Xeon、Intel SpeedStep、Itanium、Pentium は Intel Corporation または子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。

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。



---

## ご利用条件

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, 2011. All Rights Reserved.