

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

## 株価情報サンプル

**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、49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  
Stock Quote Sample

**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・オペレーション・センター

第1刷 2011.6

© Copyright IBM Corporation 2005, 2011.

---

## 目次

第 1 章 概要 . . . . .	1	メディエーションのエンドポイントの定義 . . . . .	24
第 2 章 概要 . . . . .	3	要求フローのビルド . . . . .	25
第 3 章 メディエーション設計 . . . . .	9	応答フローのビルド . . . . .	31
第 4 章 ユーザーによるビルド . . . . .	13	第 5 章 サンプルの実行 . . . . .	37
リソースのインポート . . . . .	13	ランタイム・コンポーネントのインストール . . . . .	37
StockQuoteService インターフェースの作成 . . . . .	15	メディエーション・フローをテストします . . . . .	39
一時データ用ビジネス・オブジェクトのセットアップ . . . . .	17	メディエーション・フローのデバッグ . . . . .	40
メディエーション・モジュールの作成 . . . . .	17	実行時のサービスの品質の変更 . . . . .	43
メディエーション・モジュールの組み立て . . . . .	19	特記事項 . . . . .	49
メディエーションの実装 . . . . .	24	ご利用条件 . . . . .	53



---

## 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。



---

## 第 2 章 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。

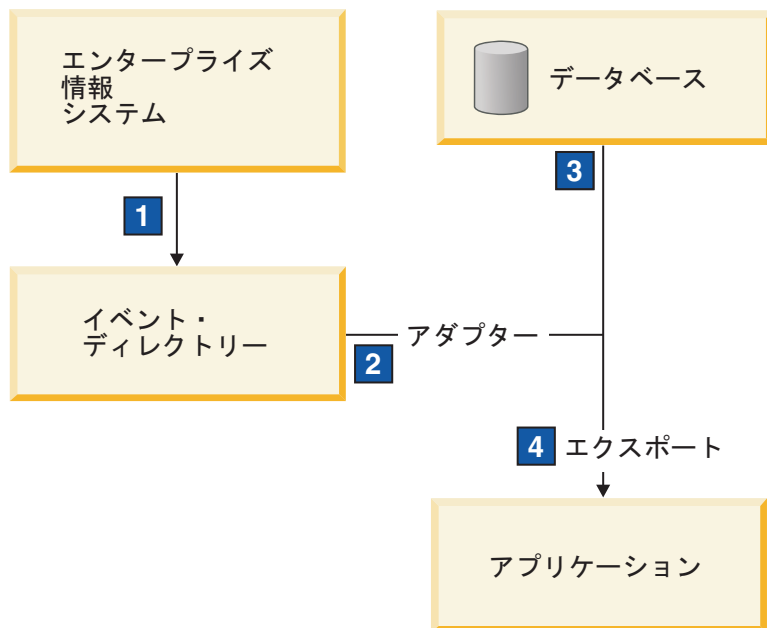
アウトバウンド処理 は、WebSphere Adapter for Flat Files が (アプリケーションのコンポーネントから) 要求を受信し、ファイル操作を実行し、該当する場合は結果を呼び出し元に返す操作モードです。操作の例としては、ファイルの作成、ファイルへの書き込み、特定のファイルが存在するかどうかの検査が含まれます。

## 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使います。

#### #### 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詳しくは、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。

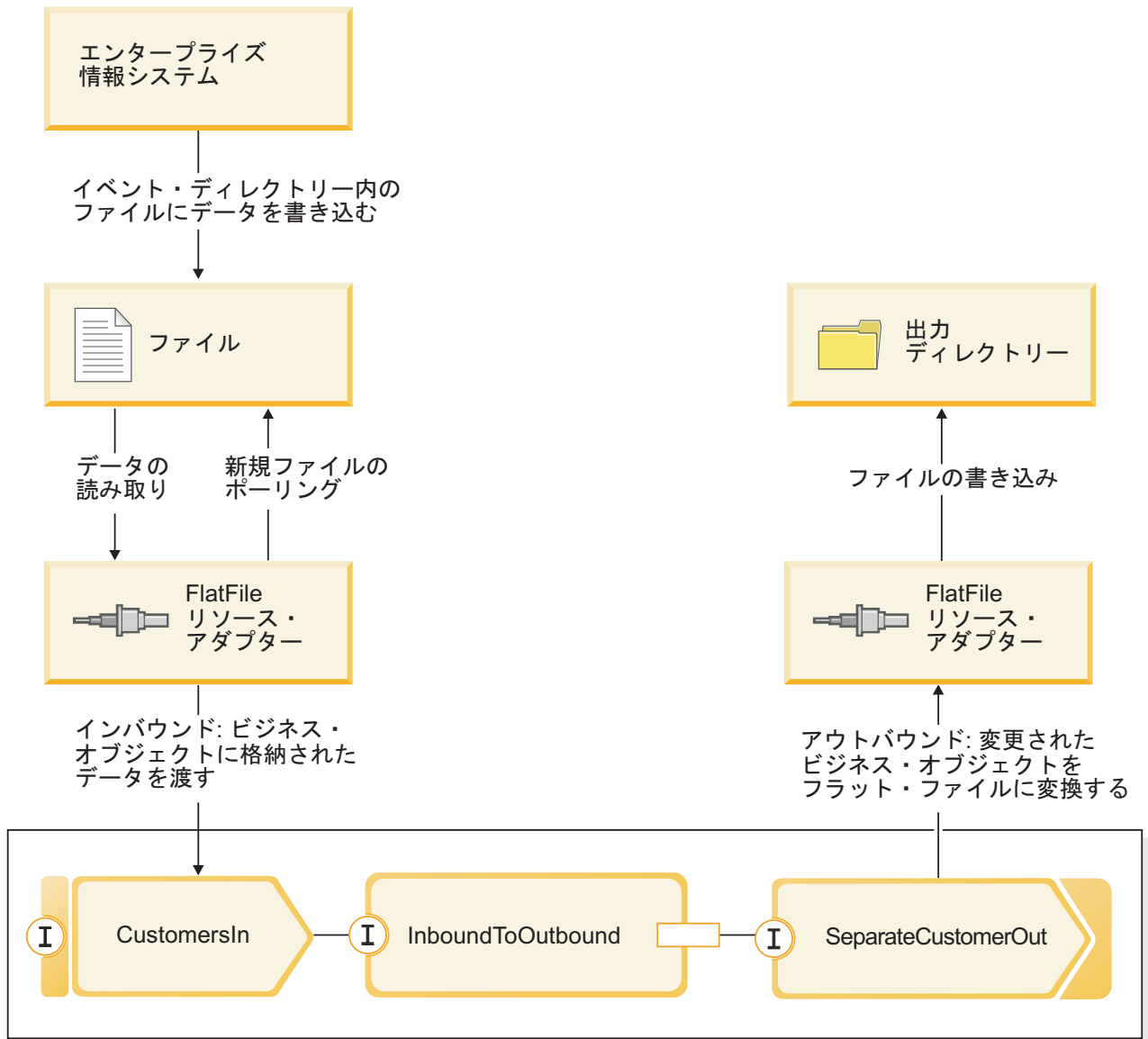
- 作成 - データを新規ファイルに保管するか、空のファイルを作成します。
- 追加 - 既存のファイルにデータを追加します。まだファイルが存在していなかった場合は、新規ファイルが作成されます。
- 上書き - 既存のファイルを新規データで上書きします。
- 削除 - 既存のファイルを削除します。
- 存在 - 特定のファイルが存在するかどうかを検査します。
- リスト - ディレクトリー内のファイルをリストします。
- 取得 - ファイルから内容を読み取ります。

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。

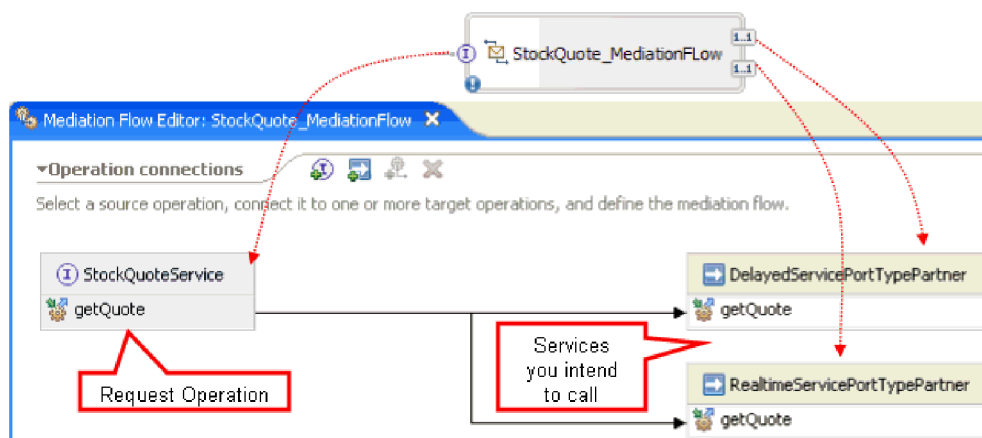


顧客レコードを含んでいるファイルがインバウンド・イベント・ディレクトリーに入っている場合、リソース・アダプターはそれらを取得し、顧客ビジネス・オブジェクトを作成し、モジュール内のエクスポートを呼び出します。このエクスポートはメディエーション・フローを開始します。メディエーション・フローはファイル・サービスを呼び出し、そのファイル・サービスは、既存のファイルの数に基づき、その名前を使用して、新しいファイル名を生成します。

## 第 3 章 メディエーション設計

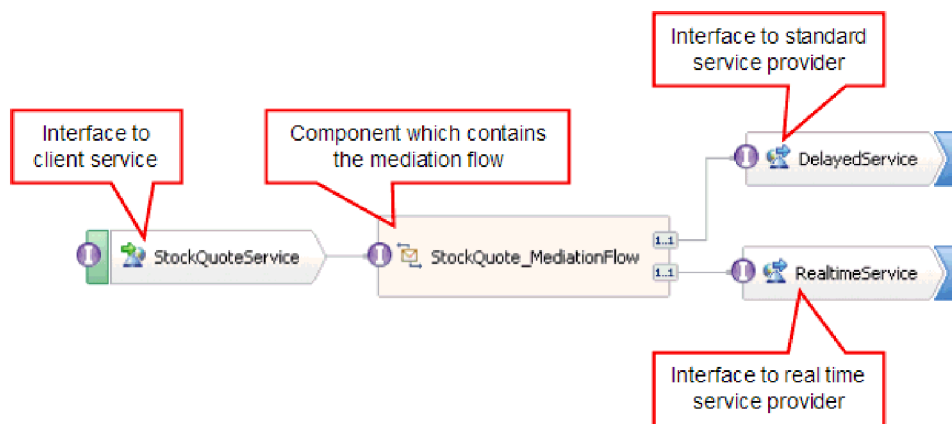
WebSphere ESB または IBM Process Server で実行されるメディエーション・サービスは、**StockQuote** という名前の単一のメディエーション・モジュールに含まれています。メディエーション・モジュールは、サービス呼び出し可能なインターフェースを提供するエクスポート、外部 Web サービス・プロバイダーへのインターフェースを提供するインポート、およびメディエーション・モジュールの実装を定義するメディエーション・フロー・コンポーネントから構成されます。

メディエーション・モジュール **StockQuote** は、アセンブリ・エディターに組み込まれ、メディエーション・フロー・コンポーネント **StockQuote\_MediationFlow** はメディエーション・フロー・エディターに作成されます。次の図は、アセンブリ・エディター内のインターフェースおよび参照と、メディエーション・フロー・エディター内のインターフェースおよび参照との関係を示しています。



### StockQuote メディエーション・モジュール

次の図は、組み立てられた **StockQuote** メディエーション・モジュールを示しています。



**StockQuote** メディエーション・モジュールは、以下のエレメントで構成されています。

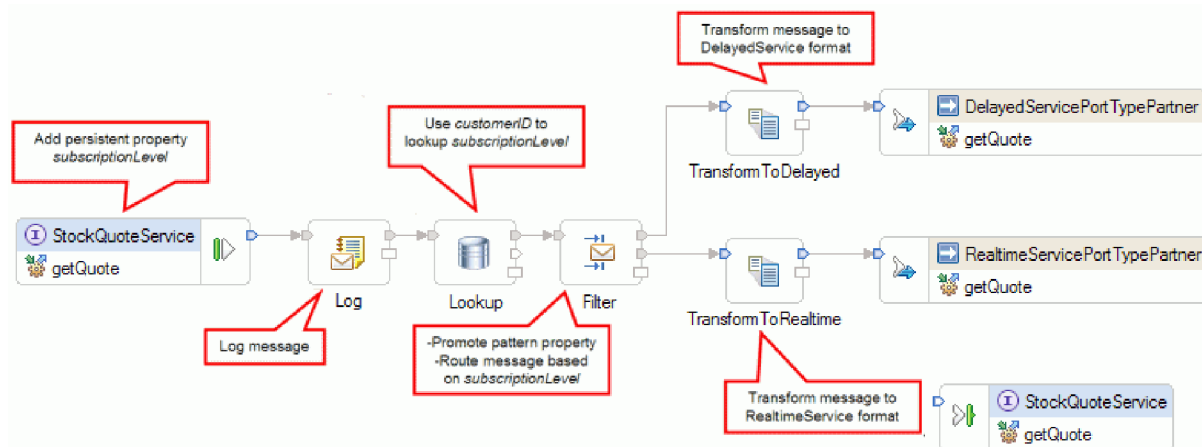
- **StockQuoteService** には、**StockQuoteService** という名前の WSDL インターフェースが用意されており、SOAP/JMS Web サービス・バインディングが使用されます。これにより、サーブレット・フロントエン

ドが JAX-RPC を使用してメディエーション・モジュールに接続できます。このサンプルでは、StockQuoteService インターフェースを作成し、WSDL ファイルを生成します。

- **StockQuote\_MediationFlow** にはメディエーション・フローが含まれています。このサンプルでは、StockQuote\_MediationFlow コンポーネントを作成および実装します。
- **RealtimeService** には、Web サービス・バインディング、およびリアルタイム (上級) サービスに適合するインターフェースが用意されています。このサンプルでは、WSDL ファイル RealtimeService.wsdl をインポートします。
- **DelayedService** には、Web サービス・バインディング、および遅延 (標準) サービスに適合するインターフェースが用意されています。このサンプルでは、WSDL ファイル DelayedService.wsdl をインポートします。

## StockQuote\_MediationFlow

下の図は、メッセージに適用されるメディエーション・ロジックを定義する要求フローを示しています。このフローは、StockQuote\_MediationFlow コンポーネントを介してターゲット・サービス・プロバイダーに流れます。

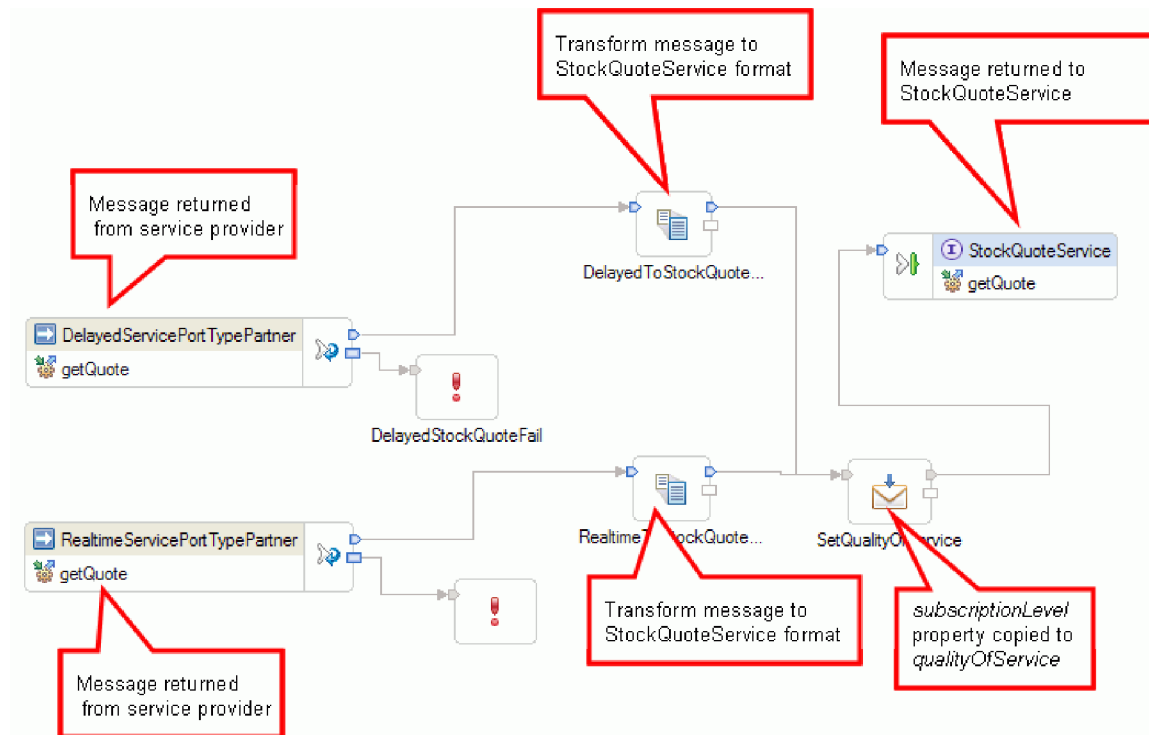


要求フローは、次の順序で左から右に実行されます。

1. メッセージの関連コンテキストにプロパティ subscriptionLevel が設定されているため、後で、応答フローでこのプロパティを使用できます。
2. 要求は、Log という名前のメッセージ・ロガー・メディエーション・プリミティブを使用して記録されます。
3. Lookup という名前のデータベース・ルックアップ・メディエーション・プリミティブは、メッセージ本体の customerID エレメントを使用し、指定の CustomerDatabase でこの情報をルックアップすることによって顧客が上級または標準のサービスを受ける資格があるかどうかを判別します。この情報は、後で使用できるように、メッセージの関連コンテキストの subscriptionLevel プロパティに追加されます。
4. 次に、関連コンテキストの subscriptionLevel 情報に基づき、Filter という名前のメッセージ・フィルターによって、要求がリアルタイム株価情報または遅延株価情報サービスのいずれかに送付されます。Filter のパターン・プロパティは、上級サービスが使用できない場合には株価情報要求を遅延サービスにリダイレクトするよう実行時に変更できるようにプロモートされます。
5. メッセージは、XSLT プリミティブ TransformToDelayed および TransformToRealtime によっていずれかのサービスへ向かう途中で変換され、そのサービスが予想するメッセージと一致するようになります。

6. 各サービスからの応答は、StockQuoteService によって要求されるフォーマットに一致するように、XSLT メディエーション・プリミティブ (DelayedToStockQuoteService および RealtimeToStockQuoteService) を介して渡されます。

下の図は、戻りメッセージに適用されるメディエーション・ロジックを定義する応答フローを示しています。このフローは、StockQuote\_MediationFlow コンポーネントを介してターゲット・サービス・プロバイダーからクライアントに流れます。メッセージ・エレメント・セッターは、subscriptionLevel の値を相関コンテキストからメッセージ内の qualityOfService プロパティにコピーする場合に使用されます。qualityOfService テキストは、リアルタイム・サービスから戻された応答には「上級」を示し、遅延サービスから戻された応答には「標準」を示します。qualityOfService テキストは、使用されたサービス・プロバイダーを示すためにクライアントに表示されます。







---

## 第 4 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

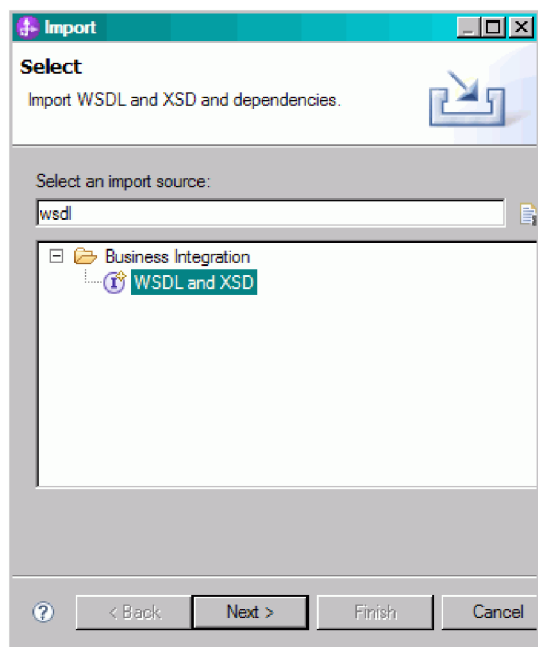
---

### リソースのインポート

ライブラリーを作成し、既製の WSDL ファイルをそのライブラリーにインポートします。これらのファイルは Web サービス・インターフェースであり、遅延およびリアルタイムのサービスへの接続に使用されません。

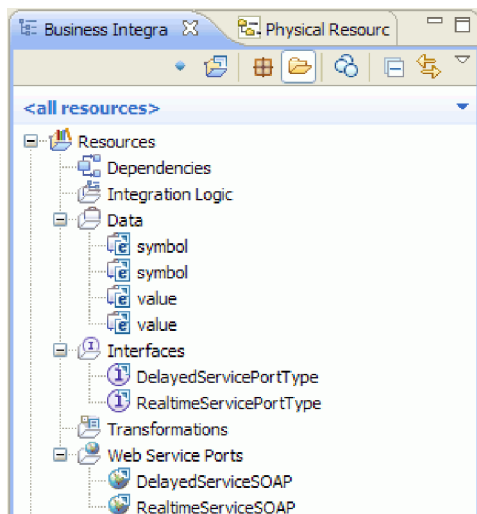
「**Resources**」ライブラリーを作成してそこに既製の WSDL ファイルをインポートするには、以下の手順を完了します。

1. 「ビジネス・インテグレーション」ビューで、「プロジェクト」の下の「**新規...**」をクリックします。プロジェクトのタイプとして「**ライブラリー**」を選択し、「**次へ**」をクリックします。「ライブラリー名」に **Resources** を指定します。「**完了**」をクリックします。「ビジネス・インテグレーション」ビューにプロジェクトが既にある場合は、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**」の順に選択します。ライブラリーに **Resources** という名前を付けます。「**完了**」をクリックします。
2. 「**Resources**」プロジェクトを右クリックし、「**インポート**」を選択します。「インポート・ソースの選択」テキスト・ボックスに **wsdl** と入力し、「**WSDL および XSD**」を選択します。「**次へ**」をクリックします。



3. wsdl をローカルにインポートするオプションを選択し、「次へ」をクリックします。
4. 「ソース・ディレクトリー」フィールドで、共有リソースをインストールしたディレクトリー (デフォルトのディレクトリーは Shared\_Resources) をブラウズします。 *Installed\_Shared\_Resources\_Directory/plugins/com.ibm.wbit.samples.content\_修飾子/artifacts/stockquote/wsdl* (ここで修飾子とは、プラグイン名に付加され、バージョン番号とタイム・スタンプを表す 7.0.0.v20091113\_0200 などのストリングです) にナビゲートします。「OK」をクリックします。
5. 「インポート」ウィザードで、WSDL ファイル「**DelayedService.wsdl**」と「**RealtimeService.wsdl**」の両方を選択します。
6. ターゲット・モジュールは「**Resources**」です。
7. 「完了」をクリックして、WSDL ファイルをインポートします。

インポートされた WSDL ファイルに基づいて、使用可能なポート・タイプとポートが、ナビゲーション・ツリーの「インターフェース」カテゴリーと「Web サービス・ポート」カテゴリーの下にそれぞれ作成されます。





## StockQuoteService インターフェースの作成

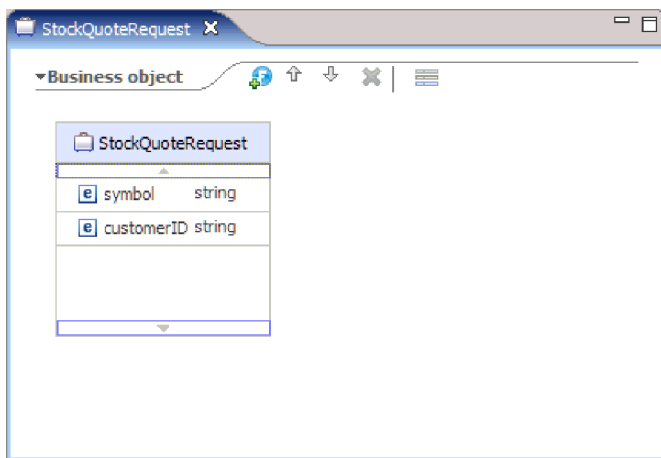
StockQuoteService インターフェースを作成します。これは、Web クライアントをメディエーション・モジュールに接続する場合に使用できるインターフェースです。


StockQuoteService インターフェースには `getQuote` 操作があります。`getQuote` 操作によって送受信されるデータは、ビジネス・オブジェクトに含まれます。リソース・ライブラリーにインターフェースおよびビジネス・オブジェクトを作成し、それらを今後、別のモジュールで使用できるようにします。

この `getQuote` 操作は、株価情報の要求を送信するのに使用されます。この操作は、`symbol` および `customerID` というフィールドを持つ `StockQuoteRequest` という名前のビジネス・オブジェクトとして要求データを送信します。また、`value` および `qualityOfService` というフィールドを持つ `StockQuoteResponse` という名前のビジネス・オブジェクトとして応答データを受信します。

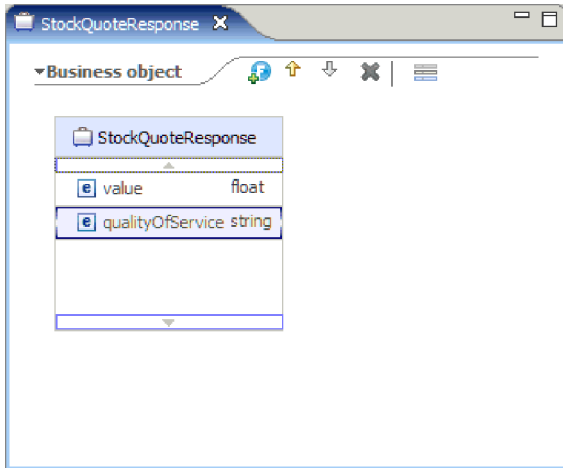
以下の手順に従って、まずビジネス・オブジェクトを、次にサービス・インターフェースを作成します。


1. 「ビジネス・インテグレーション」ビューで「**Resources**」を選択し、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。
2. 「新規ビジネス・オブジェクト」ウィザードで、「名前」フィールドに `StockQuoteRequest` と入力します。「完了」をクリックします。
3. この時点で開かれる「ビジネス・オブジェクト・エディター」で、「フィールドをビジネス・オブジェクトに追加します」ボタン  をクリックします。`field1` という名前のフィールドが作成されます。
4. 名前をクリックして入力モードに入り、`field1` を `symbol` に名前変更します。
5. 「フィールドをビジネス・オブジェクトに追加します」ボタン  をクリックして別のフィールドを追加します。フィールドの名前を `customerID` に変更します。新規ビジネス・オブジェクトを保存します。



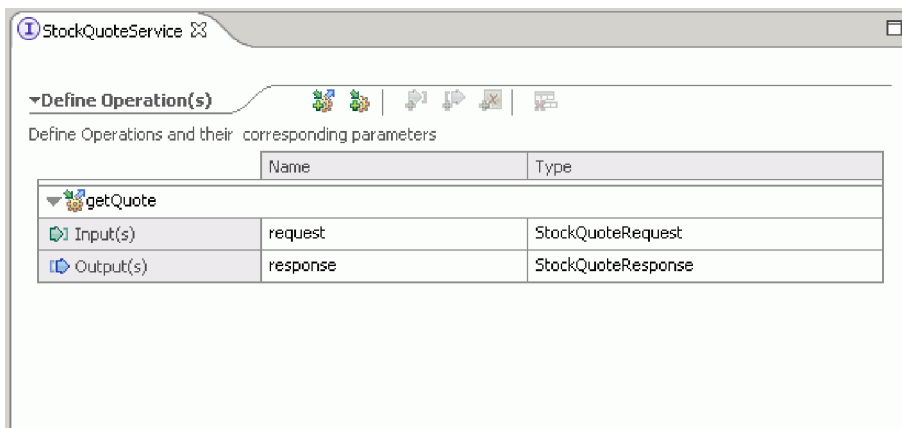
6. 「データ」を右クリックして、「新規」>「ビジネス・オブジェクト」を選択し、別のビジネス・オブジェクトを作成します。
7. 「新規ビジネス・オブジェクト」ウィザードで、`StockQuoteResponse` と入力します。「完了」をクリックします。
8. 「ビジネス・オブジェクト・エディター」で、「フィールドをビジネス・オブジェクトに追加します」ボタン  をクリックします。`field1` という名前の属性が作成されます。`field1` の名前を `value` に変更します。「ストリング」をクリックして、データ型リストから「浮動」を選択します。

- 別のフィールドを追加します。フィールドの名前を `qualityOfService` に変更します。新規ビジネス・オブジェクトを保存します。



- 「ビジネス・インテグレーション」ビューで「**Resources**」を選択し、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」ウィザードで、「名前」フィールドに `StockQuoteService` と入力します。「完了」をクリックします。
- インターフェース・エディターが開くので、そこで「**要求/応答操作の追加**」アイコンをクリックします。 . 入力と出力を 1 つずつ持つ操作 `operation1` が作成されます。 `operation1` を `getQuote` に名前変更します。
- `input1` を `request` に名前変更します。「`input1`」をクリックして選択し、`request` と入力します。
- `request` 入力の型（「**ストリング**」）をクリックし、表示される「データ型選択」ダイアログで「参照」を選択します。タイプとして「**StockQuoteRequest**」を選択し、「**OK**」をクリックします。
- `output1` を `response` に名前変更します。「`output1`」をクリックして選択し、`response` と入力します。
- 出力の型を **StockQuoteResponse** に変更します。インターフェースを保存します。

作成したインターフェースは、インターフェース・エディターに次のように表示されます。




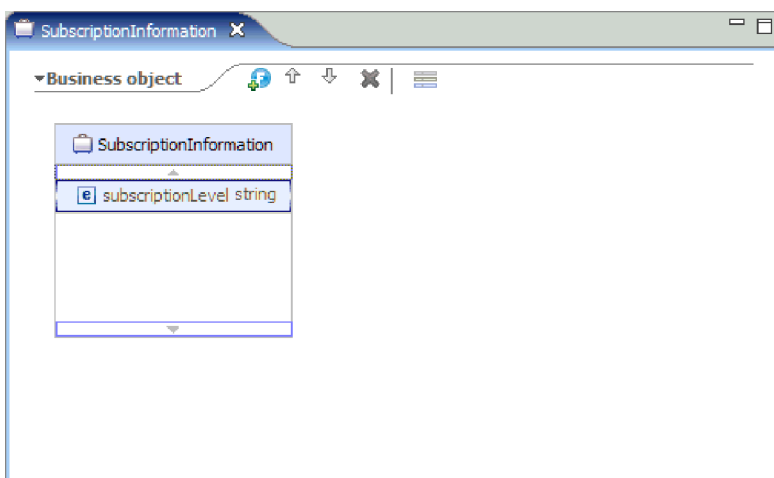
---

## 一時データ用ビジネス・オブジェクトのセットアップ

要求メッセージは、サブスクリプション・レベルの値に基づいて経路指定されます。この値は、プリミティブ間でメッセージとともに引き渡す必要があります。サブスクリプション・レベルを含むビジネス・オブジェクトを作成し、要求フローのビルド時にサブスクリプション・レベルの値を設定します。

SubscriptionInformation ビジネス・オブジェクトには subscriptionLevel フィールドが含まれます。ビジネス・オブジェクトを作成するには、以下の手順を完了します。

1. 「ビジネス・インテグレーション」ビューで「Resources」を選択します。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。「名前」フィールドで SubscriptionInformation と入力し、「完了」をクリックします。
2. ビジネス・オブジェクト・エディターで、「フィールドをビジネス・オブジェクトに追加します」ボタン  をクリックします。field1 という名前でストリング型の属性が作成されます。
3. field1 を subscriptionLevel に名前変更し、ストリング型のままとします。



要求フローを作成するとき、要求フローおよび応答フロー全体で subscriptionLevel プロパティが使用できるように、このビジネス・オブジェクトを入力ノードに追加します。開いているすべてのエディターを閉じます。

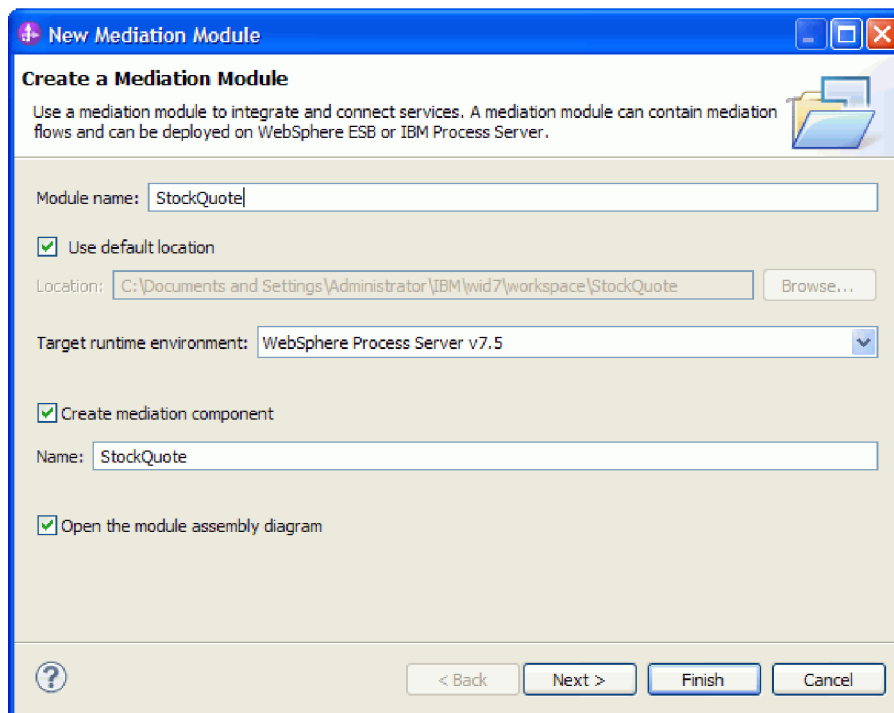
---

## メディエーション・モジュールの作成

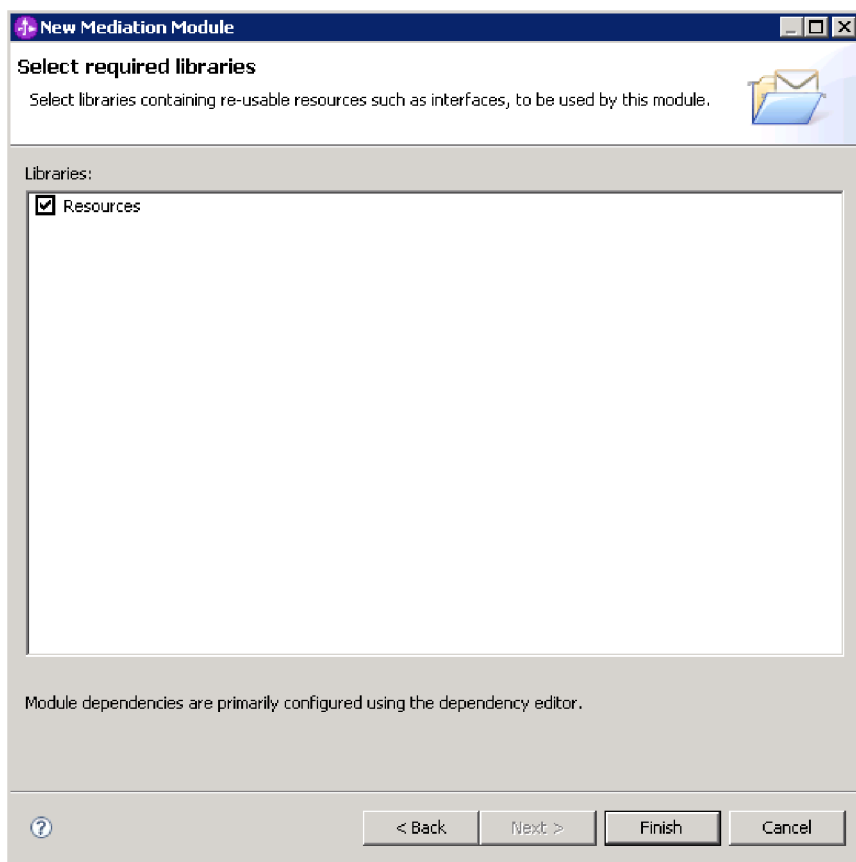
エクスポート、インポート、およびメディエーション・フロー・コンポーネントを含む StockQuote という名前のメディエーション・モジュールを作成します。

メディエーション・モジュールを作成するには、以下の手順を完了します。

1. 「ビジネス・インテグレーション」ビューで、右クリックしてコンテキスト・メニューを表示し、「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」ウィンドウが開きます。
2. 「モジュール名」フィールドに、StockQuote と入力します。
3. インストールしたサーバー・プロファイルに応じて、デフォルトのターゲット・ランタイム WebSphere ESB Server v 7.5 を保持するか、それを IBM Process Server v 7.5 に変更します。
4. 「メディエーション・コンポーネントの作成」ボックスはチェック・マークを付けたままにしておきます。



5. 「次へ」をクリックします。
6. 「必要なライブラリーの選択」ウィザードで「**Resources**」ライブラリーを選択し、「完了」をクリックします。これにより、メディエーション・モジュールでライブラリー内の成果物を使用できるようになります。

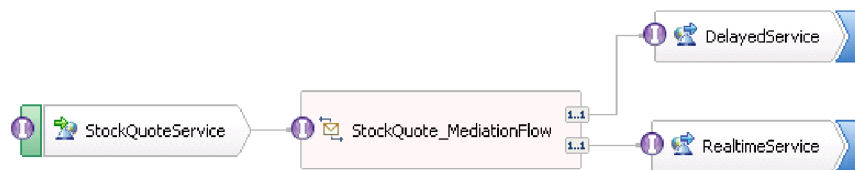



リソース・ライブラリーとの依存関係を持つ **StockQuote** という名前のメディエーション・モジュールが作成されます。**StockQuote** という名前のメディエーション・フロー・コンポーネントが、モジュールのアセンブリー・ダイアグラムに作成されます。

## メディエーション・モジュールの組み立て

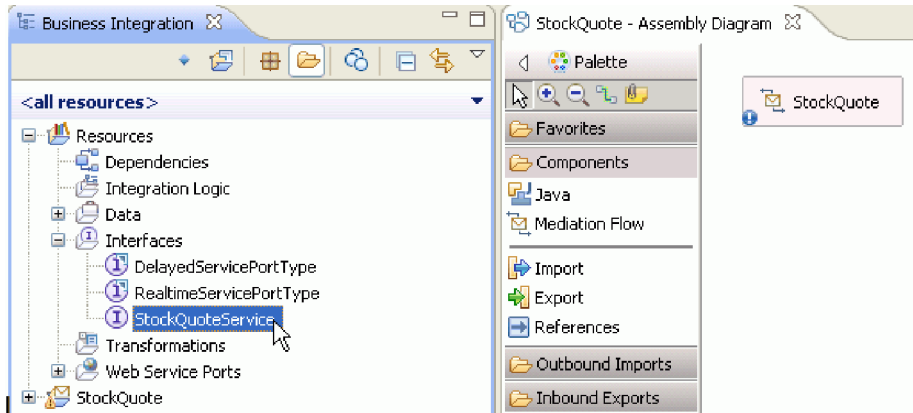
**StockQuote** サンプルのクライアント・サービスおよびプロバイダー・サービスを組み立てて、そのサービスをメディエーション・フロー・コンポーネントにワイヤリングします。

**StockQuote** メディエーション・モジュール・アセンブリー・ダイアグラムを作成するには、次の図に示されているように、以下の手順を完了します。

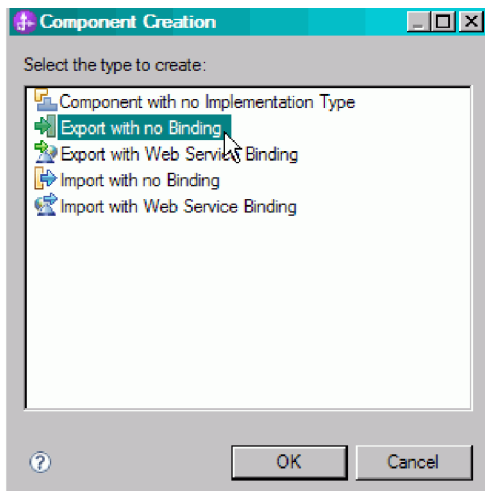


1. 「ビジネス・インテグレーション」ビューで、「**StockQuote**」モジュールを展開します。
2. アセンブリー・エディターを開くには、 「アセンブリー・ダイアグラム」をダブルクリックします。アセンブリー・エディターが開き、メディエーション・フロー・コンポーネント **StockQuote** が表示されます。

3. 「Resources」ライブラリーの「インターフェース」カテゴリで「StockQuoteService」を選択し、アセンブリー・エディターのキャンバスにドラッグします。

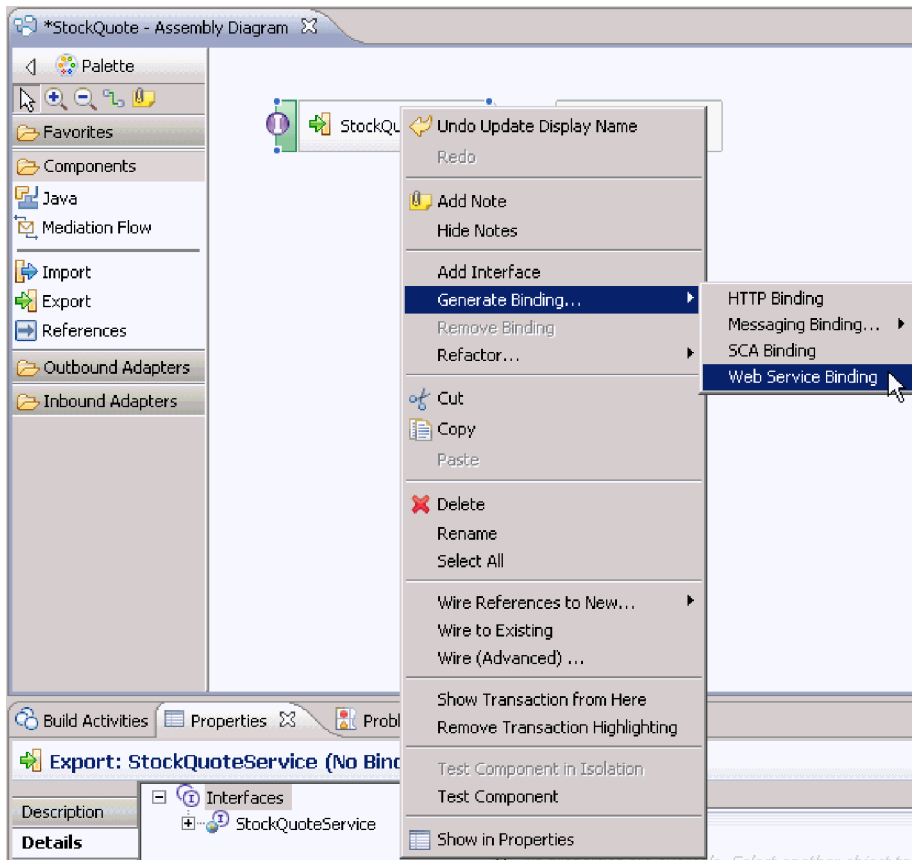


4. 「コンポーネントの作成」ダイアログが表示されるので、「バインディングのないエクスポート」を選択して作成します。



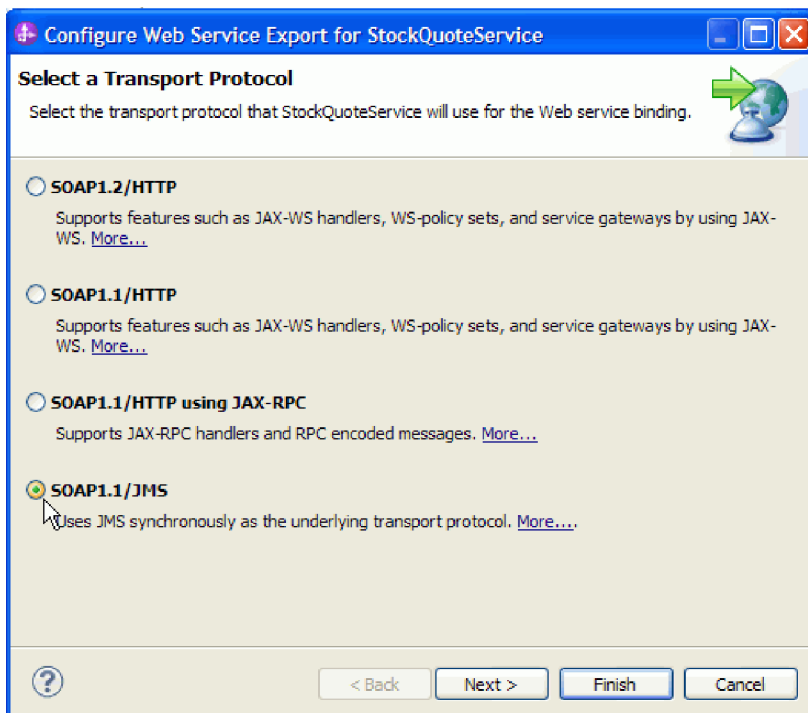
5. アセンブリー・エディターで「StockQuoteServiceExport1」をクリックして、その名前を強調表示します。StockQuoteService と入力して、エクスポートを名前変更します。この名前の変更は、エクスポート・アイコンの「プロパティ」ビューの「説明」タブに反映されます。
6. 「StockQuoteService」を右クリックし、「バインディングの生成...」>「Web サービス・バインディング」を選択します。





ここでは、トランスポート・プロトコルとして soap/jms を使用し、タイプ Web サービスのバインディングを生成します。

7. トランスポートとして「SOAP1.1/JMS」を選択し、「OK」をクリックします。

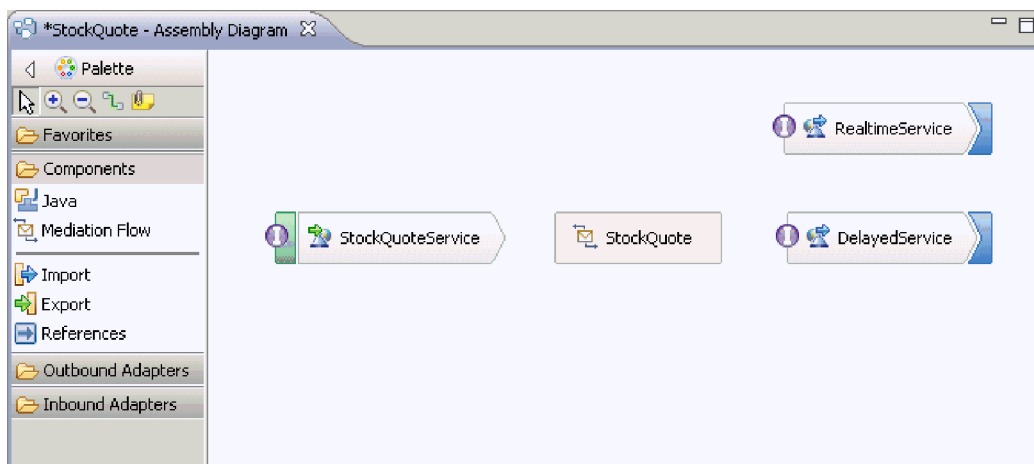


「Resources」ライブラリーの「Web サービス・ポート」カテゴリで、生成された WSDL ポートの StockQuoteService\_StockQuoteServiceJms ポートを探します。

- 「Resources」ライブラリーの「インターフェース」カテゴリで「DelayedServicePortType」を選択し、アセンブリ・エディターのキャンバスにドラッグします。「コンポーネントの作成」ダイアログで「Web サービス・バインディング付きインポート」の作成を選択し、「OK」をクリックします。
- 詳細ダイアログが表示されるので、デフォルトの設定値を保持して既存の Web サービス・ポートを使用します。「参照」をクリックします。

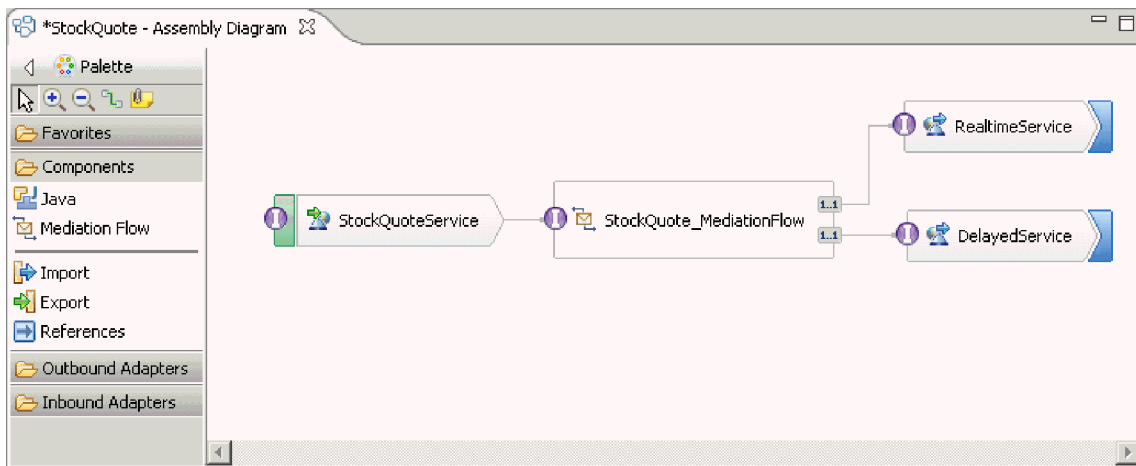


- 選択ウィザードで「DelayedServiceSOAP」をクリックします。「OK」をクリックします。
- トランスポート選択ウィンドウで、デフォルトのトランスポート SOAP 1.1/HTTP のままにし、「完了」をクリックします。
- 詳細ダイアログで「OK」をクリックします。
- インポートを選択して、名前を DelayedService に変更します。「プロパティ」ビューで「バインディング」タブをクリックし、バインディング情報を表示します。
- 「RealtimeServicePortType」をアセンブリ・エディターのキャンバスにドラッグし、「Web サービス・バインディング付きインポート」を作成します。
- ポートとして「RealtimeServiceSOAP」を指定し、トランスポート・プロトコルをデフォルトのままとします。
- インポートの名前を RealtimeService に変更します。

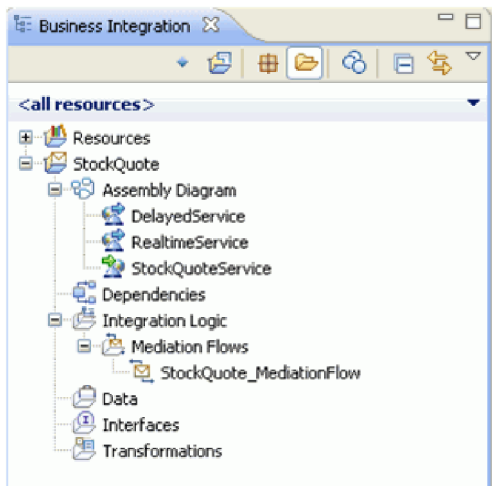


- 「StockQuote」をクリックします。これは、メディエーション・モジュールを使用して作成されたメディエーション・フロー・コンポーネントです。このコンポーネントを StockQuote\_MediationFlow に名前変更します。

18. マウスを「StockQuoteService」に移動します。オレンジ色のハンドルが表示されます。これがワイヤーのソース・ノードになります。
19. ワイヤーをターゲット・ノードである「StockQuote\_MediationFlow」にドラッグし、離します。
20. 「ワイヤーの追加」ダイアログで「OK」をクリックします。ソース・ノードのインターフェース StockQuoteService がターゲットに追加され、ワイヤーが作成されます。
21. ソース「StockQuote\_MediationFlow」からターゲット「RealtimeService」にワイヤーを作成し、「OK」をクリックします。マッチング参照 RealtimeServicePortTypePartner がソースに作成され、ワイヤーが作成されます。
22. ソース「StockQuote\_MediationFlow」からターゲット「DelayedService」にワイヤーを作成し、「OK」をクリックします。マッチング参照 DelayedServicePortTypePartner がソースに作成され、ワイヤーが作成されます。



23. StockQuote\_MediationFlow の実装を生成します。アセンブリー・エディターでコンポーネントを選択し、右クリックして「実装の生成」を選択します。StockQuote フォルダを選択して「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、ソース・インターフェースとターゲットの参照が表示されます。
24. 「株価 - アセンブリー・ダイアグラム」タブをクリックして、アセンブリー・エディターに戻ります。
25. 最初のメディエーション・コンポーネントは参照またはインターフェースを接続せずに作成されたため、これらを同期化する必要があります。これを実行するには、「StockQuote\_MediationFlow」を右クリックし、「インターフェースと参照の同期化...」>「実装へ」を選択します。「はい」をクリックします。
26. アセンブリー・ダイアグラムを保存します。「ビジネス・インテグレーション」ビューのモジュールの下で「StockQuote」アセンブリーを展開し、作成された成果物を確認します。



## メディエーションの実装

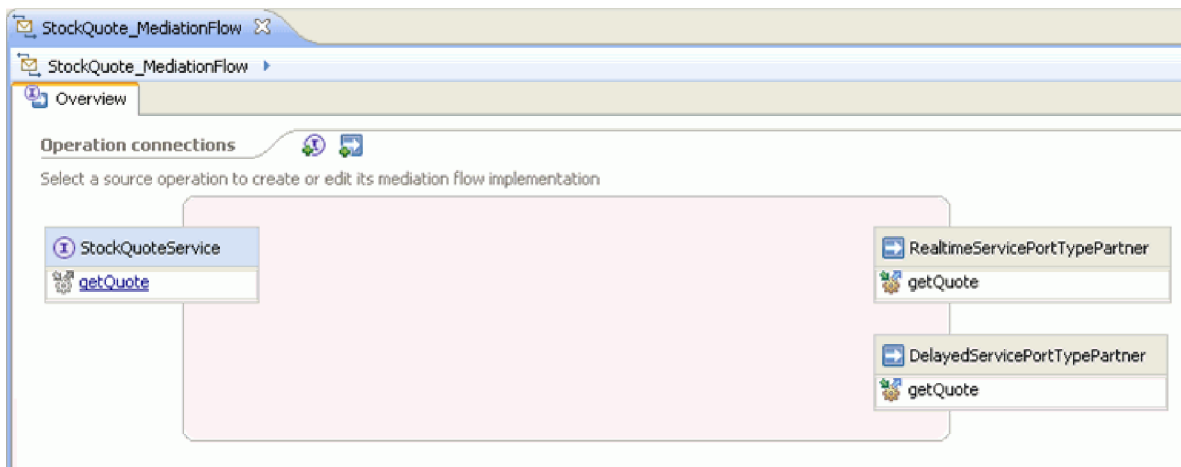
送受信操作を選択します。これらは、メディエーション・フローのエンドポイントになります。次に、エンドポイントの間にメディエーション・プリミティブを追加してメッセージをログ記録し、データベースから顧客のサブスクリプション・レベルを取得します。取得した値に基づいてメッセージを条件付きで経路指定し、受信操作のフォーマットに一致するようにメッセージを変換します。

以下のトピックの説明に従って、メディエーション・フローをビルドします。

## メディエーションのエンドポイントの定義

以下の手順を実行して、メディエーションのエンドポイントを定義します。

1. メディエーション・フロー・コンポーネントをダブルクリックしてメディエーション・フロー・エディターを開きます。



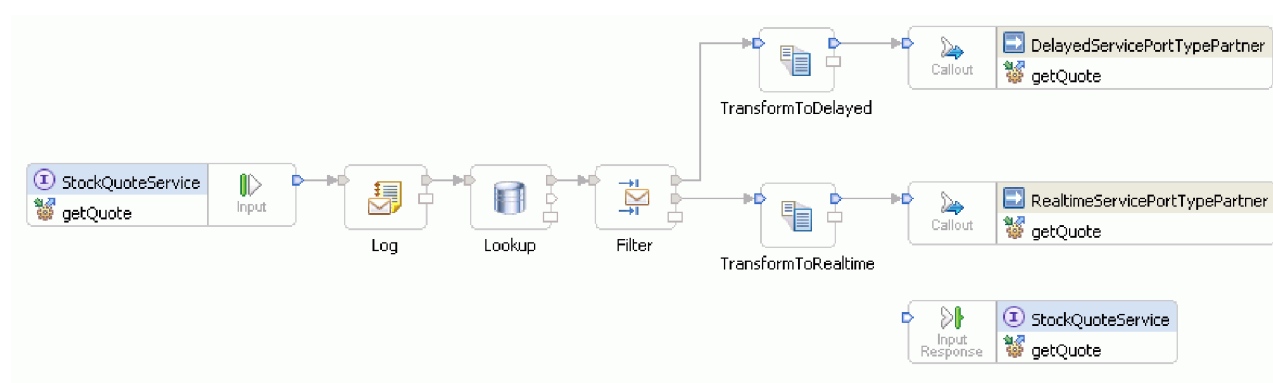
2. 「操作接続」セクションで、「StockQuoteService」インターフェースの「getQuote」操作をクリックします。
3. テンプレートの選択ダイアログで、「サービス統合」テンプレートを選択します。
4. サービスの選択ダイアログで、呼び出し対象のターゲット操作を選択します。「追加」をクリックします。

5. 「**DelayedServicePortTypePartner**」を選択し、「**getQuote**」操作を選択します。
6. 同様に、「**RealtimeServicePortTypePartner**」の「**getQuote**」操作を追加します。
7. 別個の応答フローを作成するために、デフォルトの設定値のままにします。
8. 「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。これで、要求フローのビルドに進むことができます。

## 要求フローのビルド

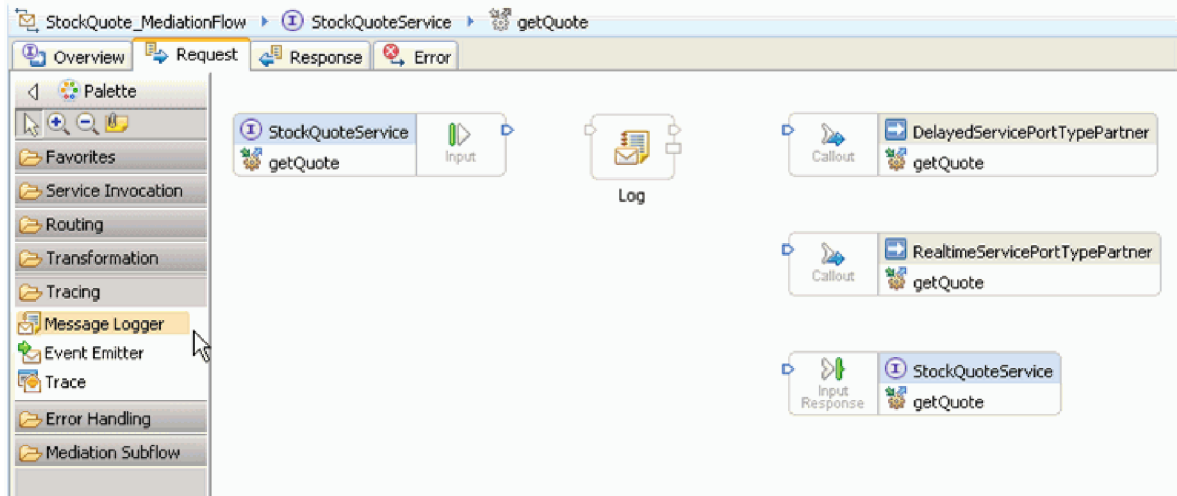
getQuote 操作の要求フローをビルドします。それには、メディエーション・プリミティブを作成し、そのプロパティを設定して、フローをワイヤリングします。

メディエーション・フロー・エディターで、要求フローは左から右に表示されます。左側に入力ノードが表示されます。これは、要求メッセージがフローに入る場所です。右側には、ターゲット操作ごとに 1 つ存在するコールアウト・ノードが 2 つあります。入力応答も表示されます。このノードは、フローの実行後にメッセージがクライアントに戻されるときに使用されます。このノードは、このサンプルでは使用しません。入力ノードとコールアウト・ノードの間にメディエーション・プリミティブを追加して、フローをワイヤリングします。完成した要求フローを下に示します。



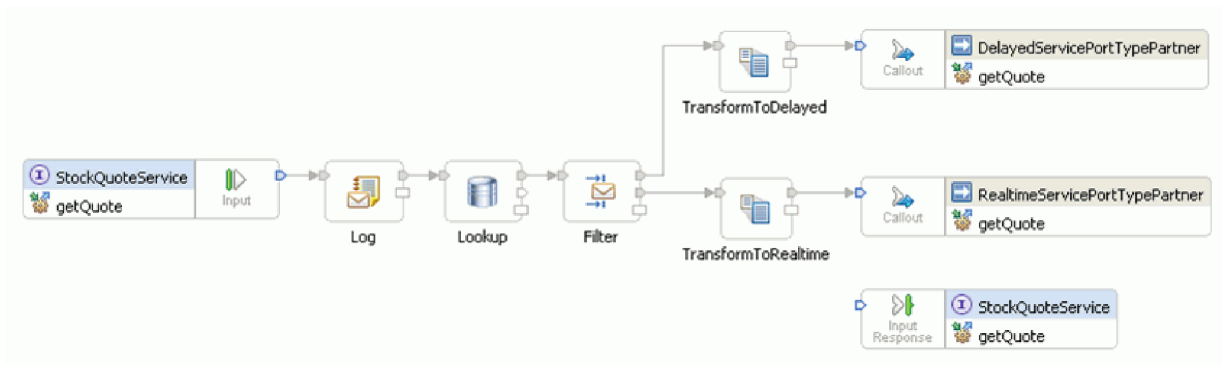
要求フローを作成するには、以下の手順を完了します。

1. メディエーション・フローが開くと、サービスの呼び出し方法を説明するヒントが表示されます。「サービス統合」テンプレートを使用してターゲット・サービスにコールアウトをすでに追加済みであるため、このヒントは閉じてかまいません。
2. 「トレース」パレット・カテゴリをクリックして、グループを展開します。
3. 「メッセージ・ロガー」プリミティブをクリックして要求フロー・キャンバスにドロップし、このプリミティブを Log に名前変更します。

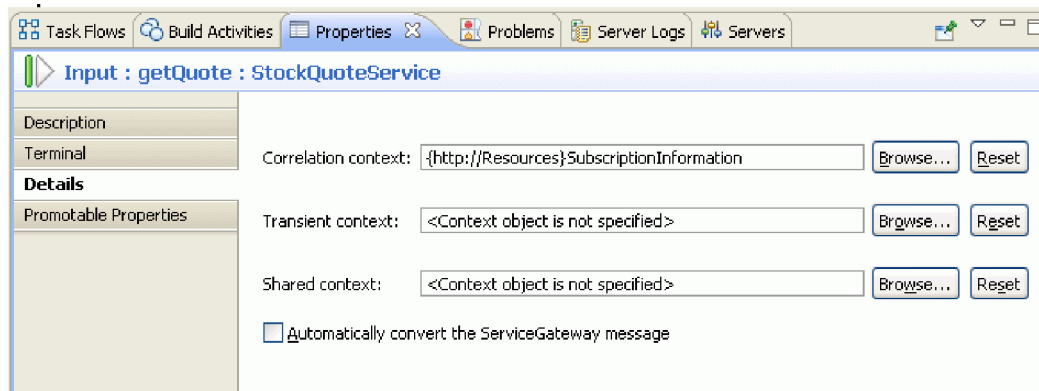


4. パレット内の「変換」フォルダーから「データベース・ルックアップ」プリミティブを選択して要求フロー・キャンバスにドロップし、それに Lookup という名前を付けます。
5. 「ルーティング」フォルダーから「メッセージ・フィルター」プリミティブを選択して要求フロー・キャンバスにドロップし、それに Filter という名前を付けます。
6. 「変換」フォルダーから「XSL 変換」プリミティブを選択して要求フロー・キャンバスにドロップし、それに TransformToDelayed という名前を付けます。
7. 別の「XSL 変換」プリミティブを選択して要求フロー・キャンバスにドロップし、それに TransformToRealtime という名前を付けます。
8. プリミティブをワイヤリングする前に、キャンバスを右クリックして、「自動レイアウト」オプションがオンになっていることを確認します。
9. 要求フロー・キャンバスでプリミティブをワイヤリングします。
  - 「getQuote : StockQuoteService」の出力ターミナルから「Log」の入力ターミナルへ
  - 「Log」の出力ターミナルから「Lookup」の入力ターミナルへ
  - 「Lookup」の出力ターミナルから「Filter」の入力ターミナルへ
  - 「Filter」のデフォルト・ターミナルから「TransformToDelayed」の入力ターミナルへ
  - 「Filter」の match1 ターミナルから「TransformToRealtime」の入力ターミナルへ
  - 「TransformToDelayed」の出力ターミナルから「getQuote : DelayedServicePortTypePartner」の入力ターミナルへ
  - 「TransformToRealtime」の出力ターミナルから「getQuote : RealtimeServicePortTypePartner」の入力ターミナルへ

ワイヤリングされた要求フローは、次のような外観になります。



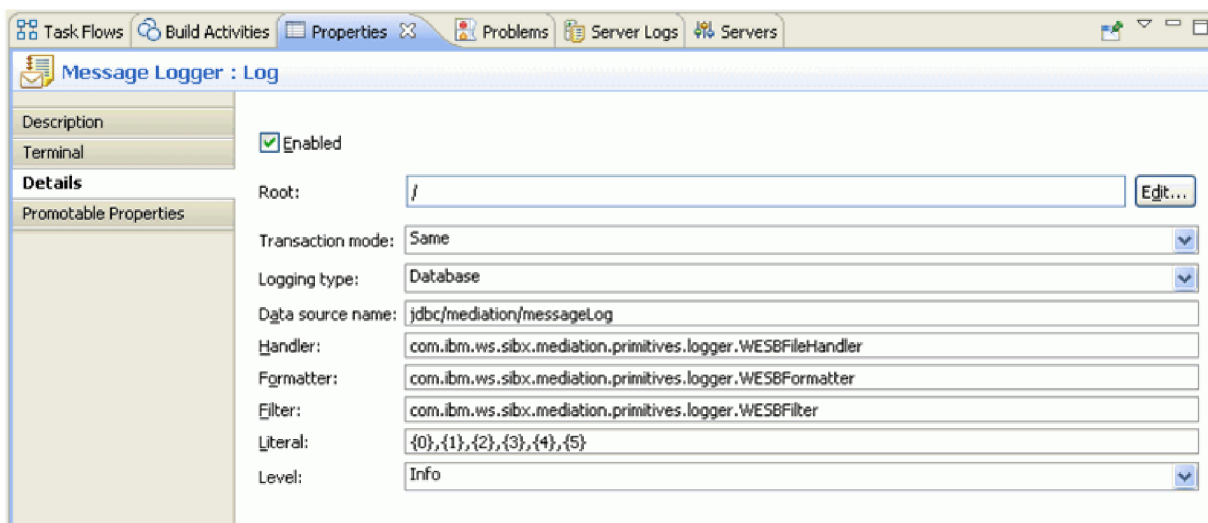
- ここで、前に作成したビジネス・オブジェクトを入力ノード「**getQuote : StockQuoteService**」の相関コンテキストに追加します。これにより、プロパティ **subscriptionLevel** がメッセージ・フローに維持されます。「**getQuote : StockQuoteService**」入力ノードをクリックし、「プロパティ」ビューの「詳細」タブに切り替えます。「相関コンテキスト」フィールドで「参照」をクリックします。「一致するデータ型」で「SubscriptionInformation」を選択して、「OK」をクリックします。ここで、「`{http://Resources}SubscriptionInformation`」が「相関コンテキスト」に表示されます。



- 要求フロー・キャンバスで「**Log**」をクリックして、「プロパティ」ビューでプリミティブのプロパティを参照します。プロパティを表示するには、「詳細」タブをクリックしてください。デフォルトのデータベースを使用して、メッセージをログに記録します。以下のプロパティが設定されていることを確認します。

表 1. メッセージ・ロガー・プロパティ

プロパティ	値
データ・ソース名 (Data source name)	jdbc/mediation/messageLog
ルート	/
トランザクション・モード	同じ



「ルート」には、ログに記録するメッセージの部分を指定します。「/」を指定するとメッセージ全体が記録され、「/body」と指定するとメッセージの本体のみが記録されます。

12. 要求フロー・キャンバスで「**Lookup**」をクリックして、「詳細」タブに次のプロパティ値を入力します。

表2. データベース・ルックアップ・プロパティ

プロパティ	値
データ・ソース名 (Data source name)	jdbc/sample/CustomerDatabase
テーブル	CUSTOMERTABLE
検索列	CUSTOMERID
検索場所	「編集...」をクリックします。XPath Expression Builderで次の XPath 式を入力するか、Simple XPath Builder を使用して作成します。/body/getQuote/request/customerID 「OK」をクリックします。

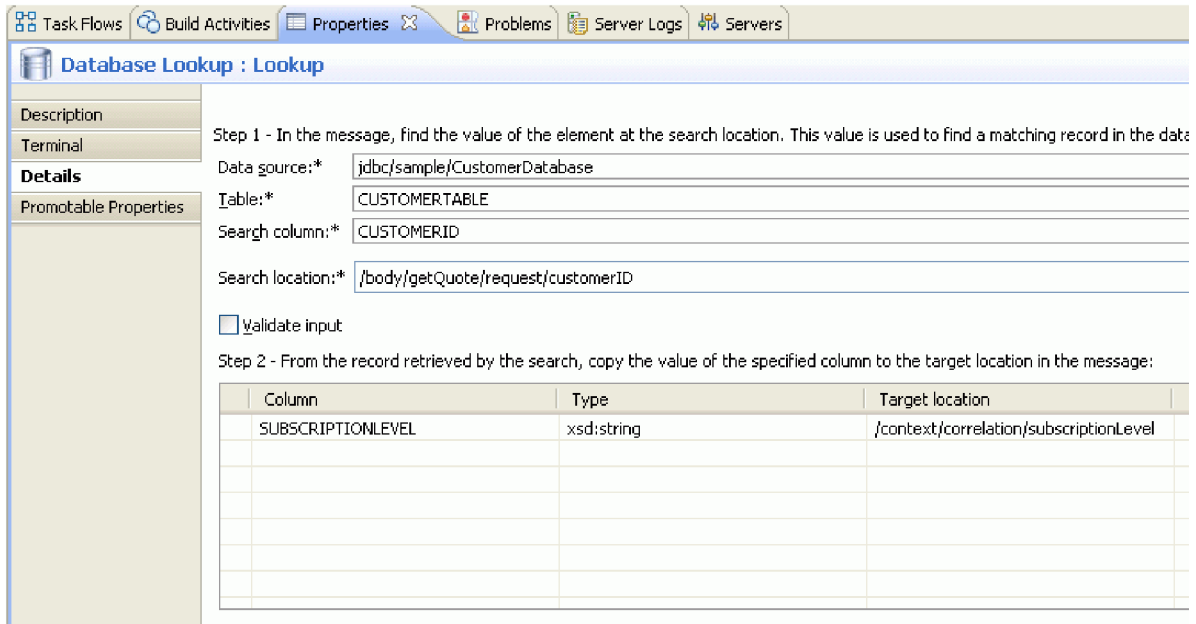
手順 2 のテーブルで、「追加...」をクリックし、以下の値を入力します。

表3. データベース・ルックアップ・テーブル・プロパティ

列	値
列	SUBSCRIPTIONLEVEL
タイプ	{http://www.w3.org/2001/XMLSchema}string
ターゲット・ロケーション	/context/correlation/subscriptionLevel

「入力の検証」のチェックマークは外しておきます。

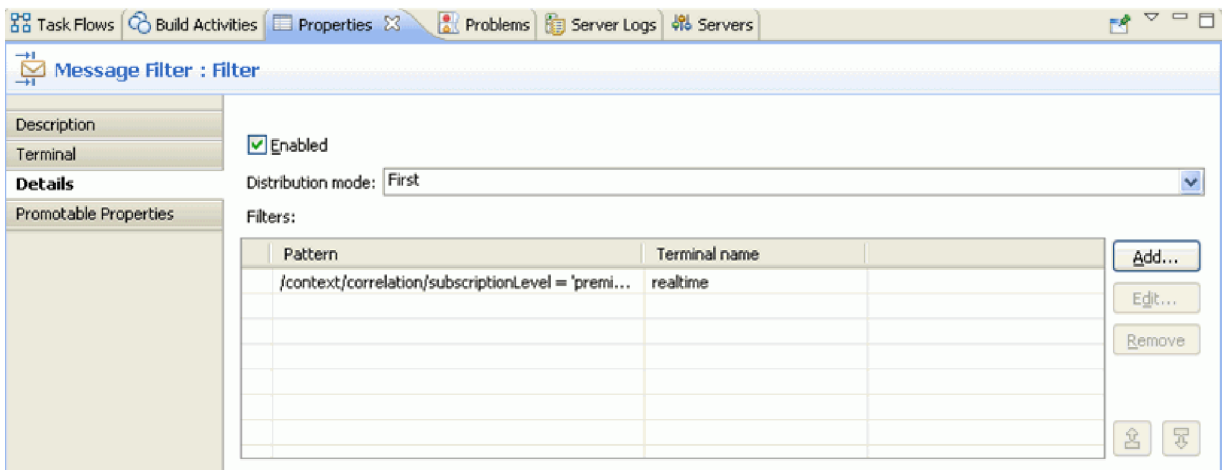




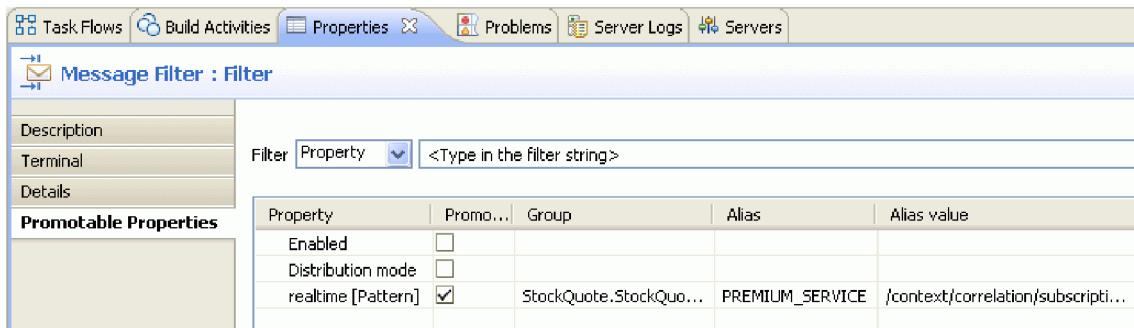
13. 要求フロー・キャンバスで「Filter」をクリックします。「ターミナル」ビューを以下の手順で選択します。
  - a. 「出力ターミナル」リストで **match1** をクリックします。 match1 ターミナルのプロパティが右に表示されます。
  - b. 「ターミナル名」フィールドで、名前を **realtime** に変更します。
14. デフォルトでは、メッセージは TransformToDelayed プリミティブに送信されます。 TransformToRealtime へのマッピングのパターンを設定する必要があります。「詳細」タブを選択します。「フィルター」テーブルで、「追加...」をクリックし、以下の値を入力します。

表4. 「フィルター」テーブル・プロパティ

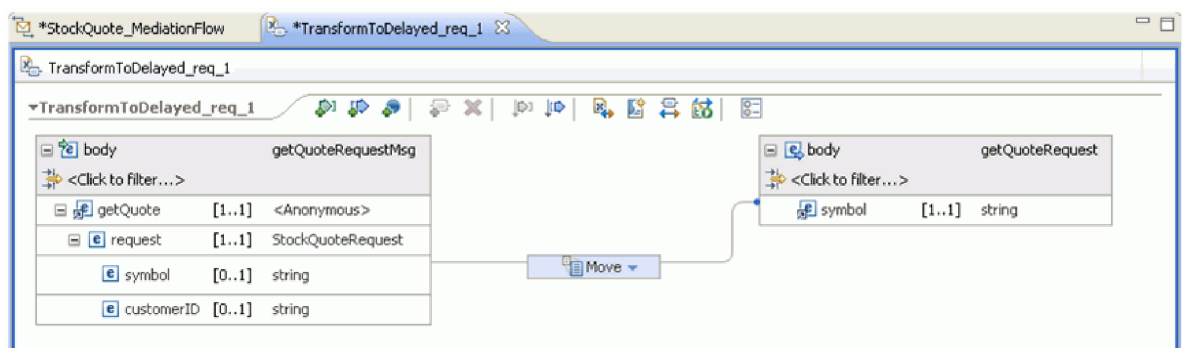
列	値
パターン	/context/correlation/subscriptionLevel = 'premium'
ターミナル名	realtime



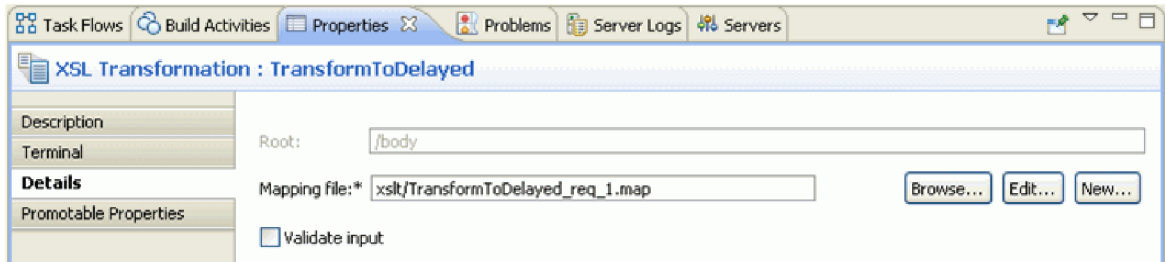
15. プロモートされたプロパティは、管理者が実行時に変更できます。パターン・プロパティを実行時に変更してサービスの品質を変更することができます。パターン・プロパティをプロモートするには、以下の手順に従います。
  - a. 「プロモート可能なプロパティ」タブをクリックします。
  - b. realtime [パターン] プロパティの「プロモート済み」チェック・ボックスをクリックします。
  - c. 別名「Filter.filters」をクリックします。PREMIUM\_SERVICE と入力して、その別名を名前変更します。



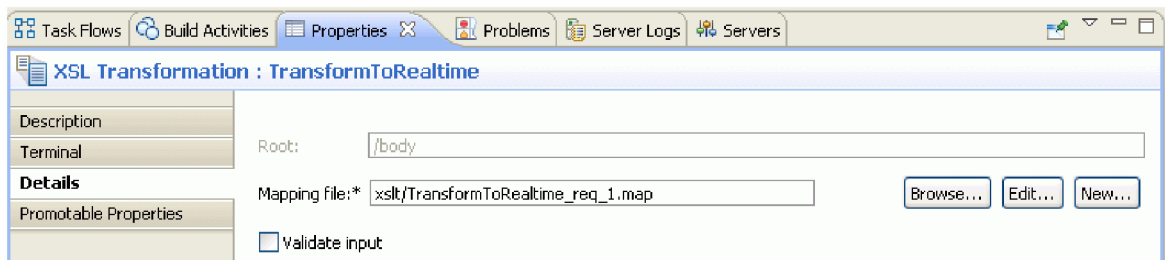
16. XSL 変換プリミティブ TransformToDelayed のプロパティを設定します。
  - a. 要求フロー・キャンバスで「**TransformToDelayed**」プリミティブを選択してダブルクリックします。
  - b. 「次へ」をクリックして、マップされるルート、入力、および出力のメッセージ・タイプを表示します。「完了」をクリックしてデフォルト設定を受け入れます。これにより、マップ・エディターが起動します。
  - c. 左側の入力オブジェクト側で、「**body**」 > 「**getQuote**」 > 「**request**」を展開します。右側の出力オブジェクト側で、「**body**」を展開します。左側の symbol をクリックし、それを右側の symbol の上までドラッグしてそれらをワイヤリングし、マッピングを作成します。



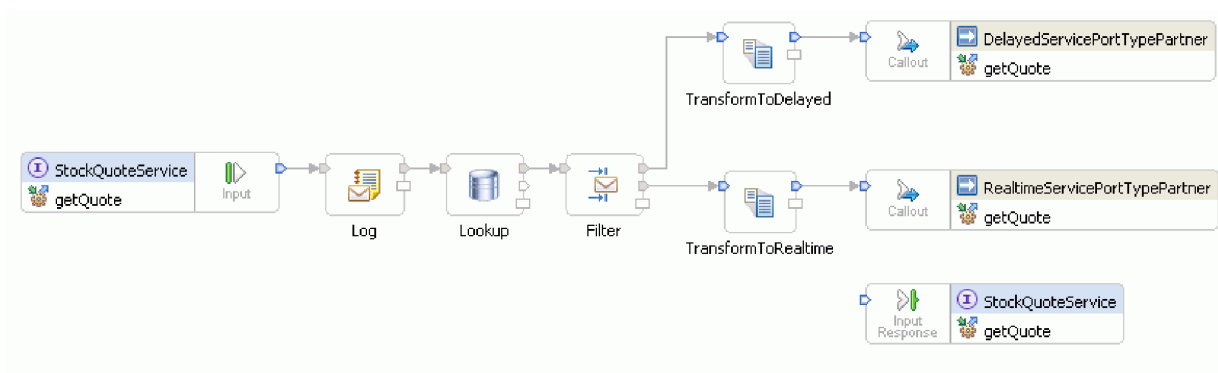
- d. 変更内容を保存し、マップ・エディターを閉じます。マッピング・ファイルが「プロパティ」ビューの「詳細」タブに表示されます。



17. 同様に、XSL 変換プリミティブ TransformToRealtime のプロパティを設定します。
  - a. 要求フロー・キャンバスで「**TransformToRealtime**」プリミティブを選択してダブルクリックします。
  - b. 「新規 XML マッピング」ウィザードが開きます。「次へ」をクリックして、マップされるルート、入力、および出力のメッセージ・タイプを表示します。「完了」をクリックしてデフォルトを受け入れます。これにより、マップ・エディターが開きます。
  - c. 左側で、「body」>「getQuote」>「request」を展開します。右側で、「body」を展開します。左側の *symbol* をクリックし、それを右側の *symbol* の上までドラッグしてマッピングを作成します。
  - d. 変更内容を保存し、マップ・エディターを閉じます。マッピング・ファイルが「プロパティ」ビューの「詳細」タブに表示されます。



18. 要求フローを保存します。

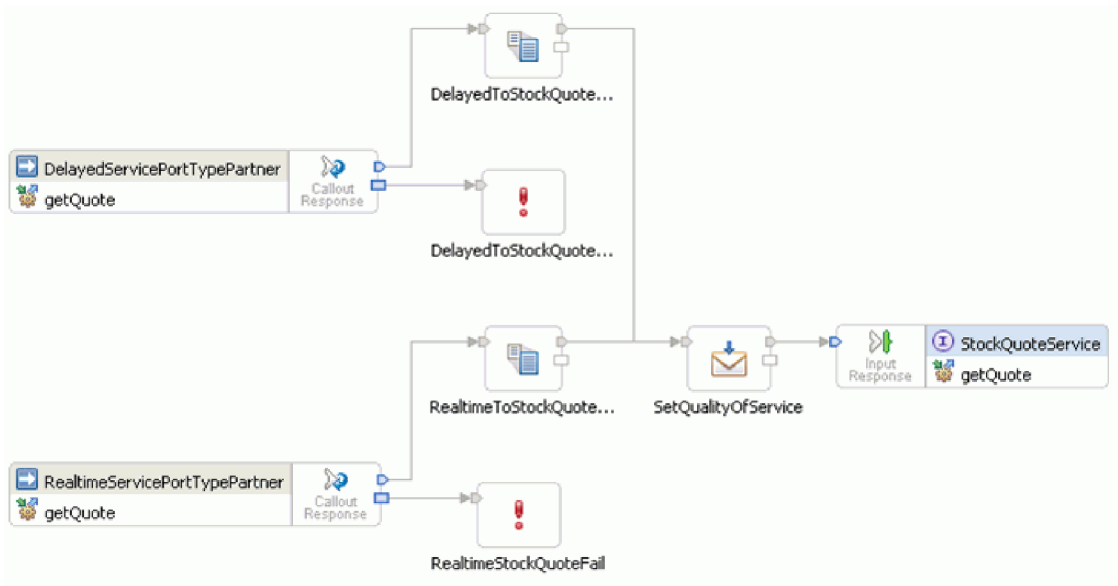


## 応答フローのビルド

getQuote 操作の応答フローをビルドします。それには、メディエーション・プリミティブを作成し、そのプロパティを設定して、フローをワイヤリングします。

応答フローは、エディターで左から右に表されます。左側には、ターゲット操作ごとに 1 つ存在するコールアウト応答ノードがあります。これは、返されるメッセージが応答フローに入る場所です。右側には、ソ

ース操作に返されるメッセージを示す入力応答ノードがあります。ノード間にメディエーション・プリミティブを追加してフローをワイヤリングします。完成した応答フローを下に示します。

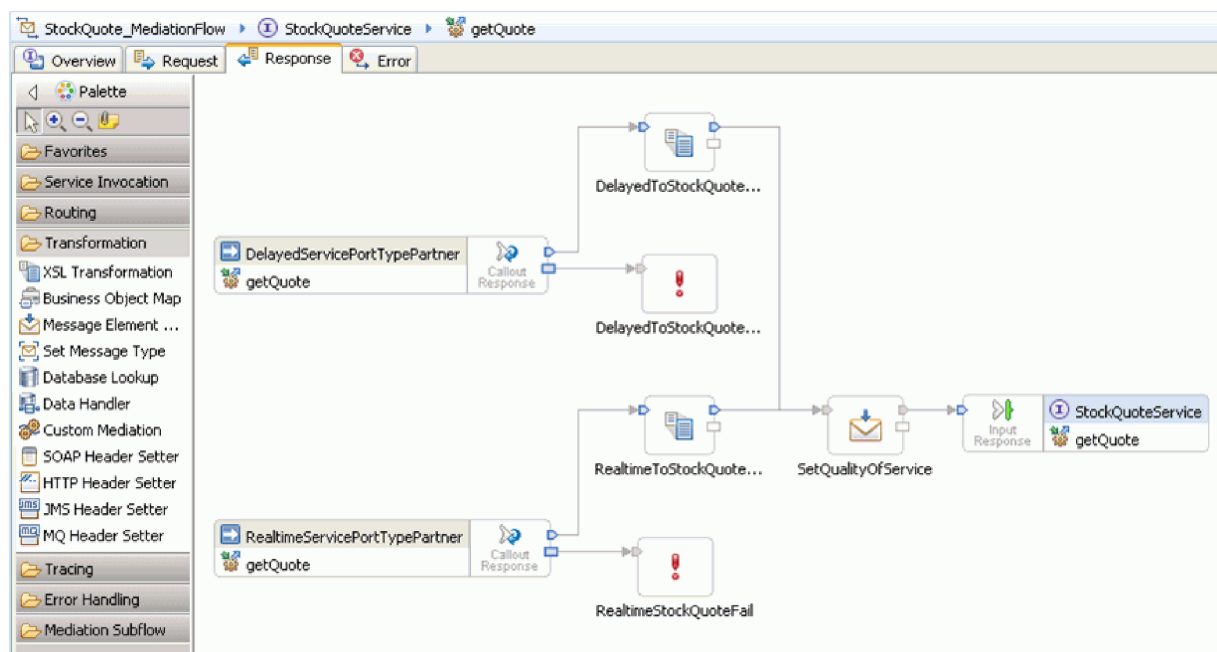


応答フローを作成するには、以下の手順を完了します。

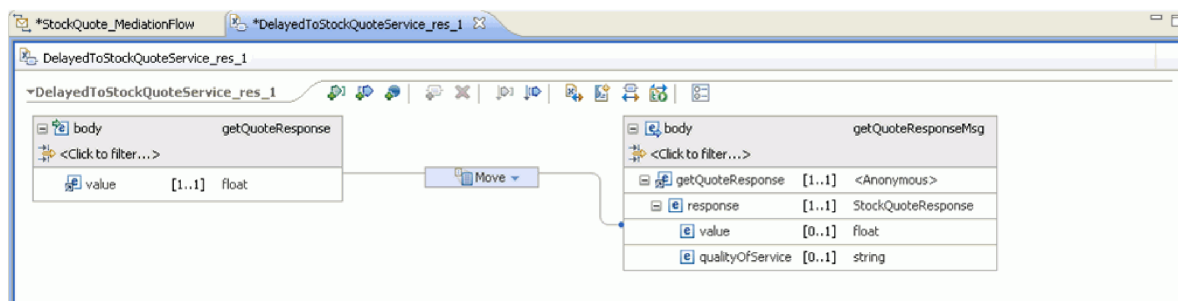
1. 「応答」タブをクリックして、応答フローを表示します。
2. パレット内の「変換」フォルダーから「XSL 変換」プリミティブを選択し、それを応答フロー・キャンバスにドロップして、DelayedToStockQuoteService に名前変更します。
3. 別の「XSL 変換」プリミティブを選択し、それを応答フロー・キャンバスにドロップして、RealtimeToStockQuoteService に名前変更します。
4. 「変換」フォルダーから「メッセージ・エレメント・セッター」プリミティブを選択し、それをキャンバスにドロップして、SetQualityOfService に名前変更します。
5. 「エラー処理」フォルダーから「障害」プリミティブを選択し、それをキャンバスにドロップして、DelayedStockQuoteFail に名前変更します。さらに別の「障害」プリミティブをキャンバスに追加して、RealtimeStockQuoteFail に名前変更します。
6. プリミティブをワイヤリングする前に、キャンバスを右クリックして、「自動レイアウト」オプションがオンになっていることを確認します。
7. 次のように、プリミティブをワイヤリングします。
  - 「getQuote : DelayedServicePortTypePartner」の出力ターミナルから「DelayedToStockQuoteService」の入力ターミナルへ
  - 「getQuote : RealtimeServicePortTypePartner」の出力ターミナルから「RealtimeToStockQuoteService」の入力ターミナルへ
  - 「DelayedToStockQuoteService」の出力ターミナルから「SetQualityOfService」の入力ターミナルへ
  - 「RealtimeToStockQuoteService」の出力ターミナルから「SetQualityOfService」の入力ターミナルへ
  - 「SetQualityOfService」の出力ターミナルから「getQuote : StockQuoteService」の入力ターミナルへ
  - 「getQuote : DelayedServicePortTypePartner」の障害ターミナルから「DelayedStockQuoteFail」の入力ターミナルへ

- 「getQuote : RealtimeServicePortTypePartner」の障害ターミナルから「RealtimeStockQuoteFail」の入カターミナルへ

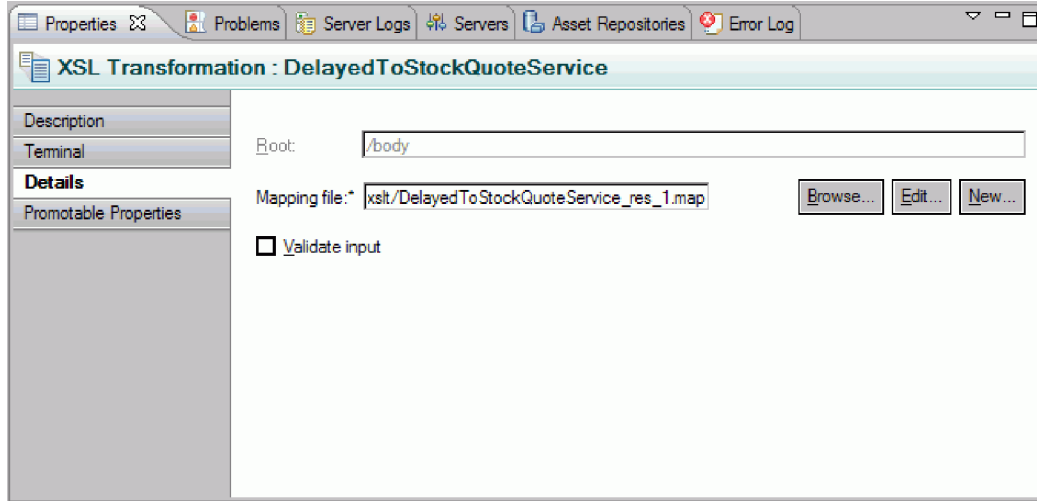
ワイヤリングされた応答フローは、次のような外観になります。



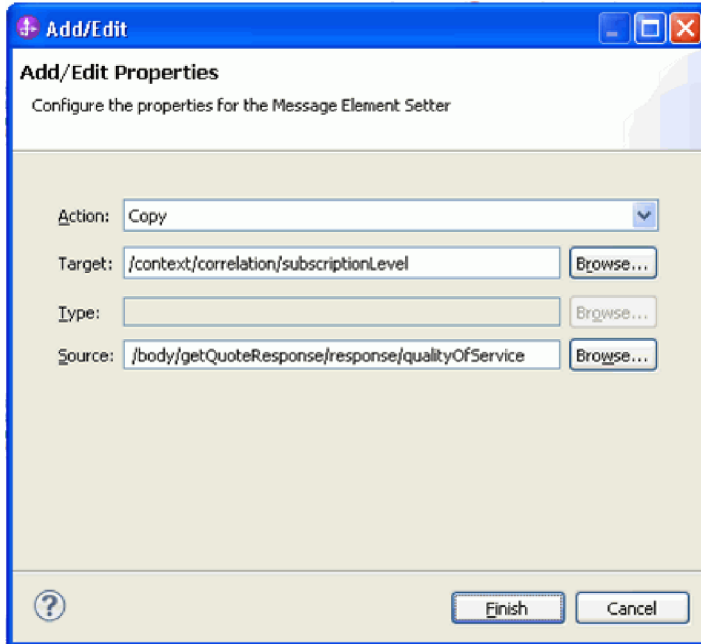
8. XSLT プリミティブ `DelayedToStockQuoteService` のプロパティを設定します。
  - a. 応答フロー・キャンバスで「**DelayedToStockQuoteService**」プリミティブを選択してダブルクリックします。
  - b. 「新規 XML マッピング」ウィザードが開きます。「次へ」をクリックして、マップされるルート、入力、および出力のメッセージ・タイプを表示します。デフォルトを受け入れて「完了」をクリックします。
  - c. マップ・エディターの入力オブジェクト・セクション (左側) で、「**body**」を展開します。出力オブジェクト・セクション (右側) で、「**body**」 > 「**getQuoteResponse**」 > 「**response**」を展開します。
  - d. 左側の値をクリックし、それを右側の値の上までドラッグしてそれらをワイヤリングし、マッピングを作成します。
  - e. 変更内容を保存し、マップ・エディターを閉じます。



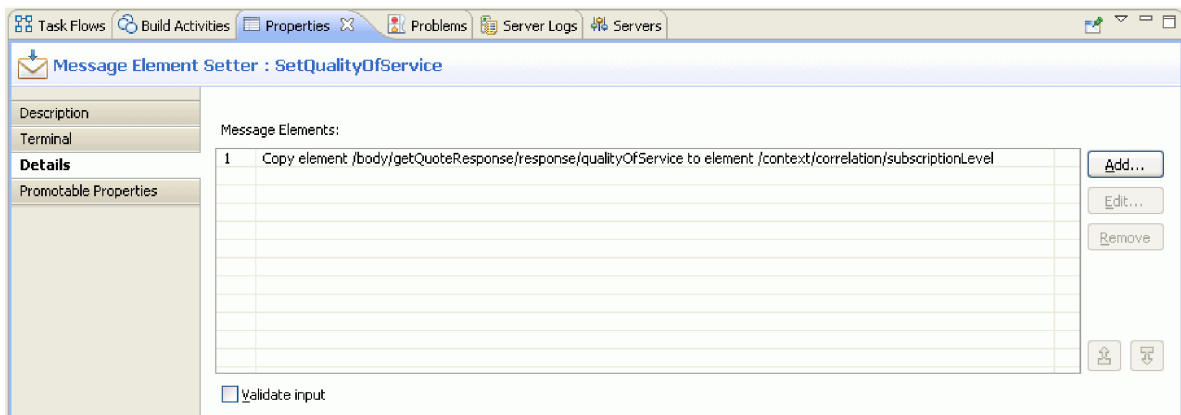
マッピング・ファイルが「プロパティ」ビューの「詳細」タブに表示されます。



9. 同様に、XSLT プリミティブ **RealtimeToStockQuoteService** のプロパティを設定します。
  - a. 応答フロー・キャンバスで「**RealtimeToStockQuoteService**」プリミティブを選択してダブルクリックします。
  - b. 「新規 XML マッピング」ウィザードが開きます。「次へ」をクリックして、マップされるルート、入力、および出力のメッセージ・タイプを表示します。デフォルトを受け入れて「完了」をクリックします。
  - c. マップ・エディターの入力オブジェクト・セクション (左側) で、「**body**」を展開します。出力オブジェクト・セクション (右側) で、「**body**」 > 「**getResponse**」 > 「**response**」を展開します。
  - d. 左側の値をクリックし、それを右側の値の上までドラッグしてそれらをワイヤリングし、マッピングを作成します。
  - e. 変更内容を保存し、マップ・エディターを閉じます。マッピング・ファイルと関連 XSL スタイル・シートが、「プロパティ」ビューの「詳細」タブに表示されます。
10. メッセージ・エレメント・セッター・プリミティブ **SetQualityOfService** のプロパティを設定します。
  - a. 応答フロー・キャンバスで「**SetQualityOfService**」プリミティブを選択します。「プロパティ」ビューの「詳細」タブに切り替えます。
  - b. 「追加...」をクリックして「プロパティの追加/編集」ウィザードを開始します。
  - c. 「アクション」ドロップダウンから「コピー」を選択します。「ターゲット」で「参照」を選択すると、XPath Expression Builder が起動されます。
  - d. データ型ビューアーで、「**ServiceMessageObject**」 > 「**body**」 > 「**getResponse**」 > 「**response : StockQuoteResponse**」を展開し、「**qualityOfService**」を選択します。「式」フィールドに XPath 式が表示されます。「OK」をクリックします。
  - e. 「ソース」フィールドで「参照」を選択します。データ型ビューアーで、「**ServiceMessageObject**」 > 「**context**」 > 「**correlation**」を展開し、「**subscriptionLevel**」を選択します。「式」フィールドに XPath 式が表示されます。「OK」をクリックします。「追加/編集」ウィンドウで、「完了」をクリックします。

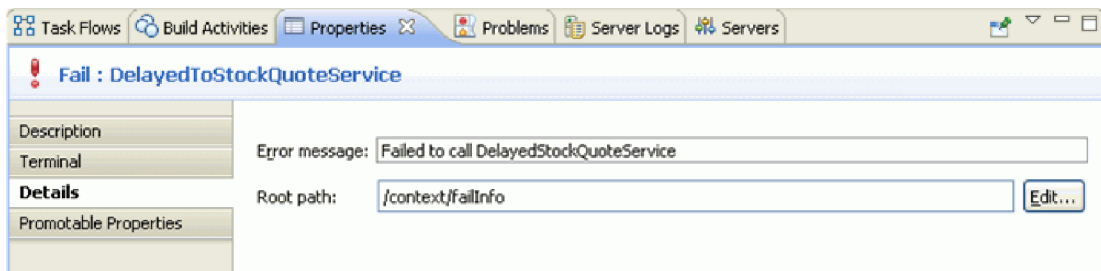


テーブルの「ターゲット」、「型」、および「値」の列が、図に示すように、1 行目に取り込まれます。これらを後で編集するには、行を選択し、「編集...」をクリックします。



11. 2 つの障害プリミティブを以下のように設定します。

- a. 「**DelayedStockQuoteFail**」を右クリックして「プロパティを表示」を選択します。「詳細」をクリックし、「エラー・メッセージ」フィールドに Failed to call DelayedStockQuoteService と入力します。



- b. 「**RealtimeStockQuoteFail**」で、「エラー・メッセージ」フィールドに Failed to call RealtimeStockQuoteService と入力します。

12. **Ctrl+S** を押してフローを保存します。



## 第 5 章 サンプルの実行

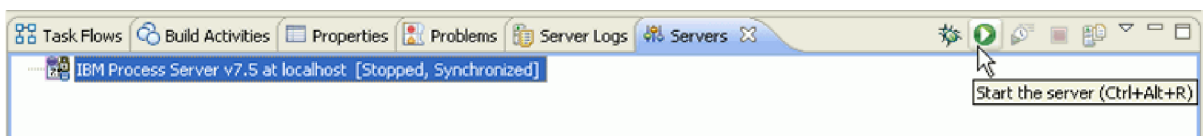
サンプルのビルドまたはインポート後、統合テスト・クライアントまたはデバッガーで StockQuote アプリケーションをテストできます。

株価情報サンプルには、複数のランタイム・コンポーネントが含まれています。それは、テスト・クライアントで指定したシンボルについてランダムに株価情報を生成するリアルタイム Web サービスと遅延 Web サービス、および顧客データがプリロードされた Derby データベースです。サンプルをテストする前に、スクリプトを実行してランタイム・コンポーネントをインストールする必要があります。スクリプトは一度のみ実行する必要があります。次に、サンプルを実行するには、サーバーに StockQuoteApp プロジェクトを追加して、テスト・クライアントを起動します。

### ランタイム・コンポーネントのインストール

顧客データベースとバックエンドをサーバーにインストールするスクリプトを実行するには、以下の手順を完了します。

1. 「サーバー」ビューに切り替えます。サーバー・プロファイルが 1 つしかインストールされていない場合 (IBM Process Server または WebSphere ESB Server)、サーバーを右クリックし、「始動」を選択してサーバーを始動します。両方のサーバー・プロファイルがインストールされている場合は、いずれか 1 つを選択します。



注: 「サーバー」ビューにサーバーが表示されない場合は、WebSphere テスト環境をインストールしていない可能性があります。製品ランチパッドを実行して、WebSphere テスト環境をインストールしてください。

2. ここで、このサンプルで使用するランタイム・コンポーネントをインストールします。サーバーが正常に始動したら、コマンド・プロンプトを開き、`Install Shared Resources Directory/plugins/com.ibm.wbit.samples.content/artifacts/stockquote/bin/` に切り替えます。注: ご使用のインストール環境によっては、複数のバージョンの `com.ibm.wbit.samples.content` がある場合があります。常に最高位バージョン番号のもの、つまり最新のものを選択してください。
3. 以下のいずれかのコマンドを実行します。
  - IBM Process Server を始動した場合: `Installation Directory/runtimes/bi_version_number/bin/wsadmin -f wid-install.jacl -profileName wps_profile_name -username username -password password`
  - WebSphere ESB Server を始動した場合: `Installation Directory/runtimes/bi_version_number/bin/wsadmin -f wid-install.jacl -profileName esb_profile_name -username username -password password`

コマンド・プロンプトは再度必要となるため、開いたままにしておきます。

#### トラブルシューティングのヒント

- インストール・ディレクトリーにスペースが含まれている場合は、コマンドの実行時にエラーが表示されます。例えば、インストール・ディレクトリーが `d:\Program Files` の場合、次のエラーが表示されます。

'D:\Program' は、内部コマンドまたは外部コマンド、操作可能なプログラムまたはバッチ・ファイルとして認識されていません。

次のように、コマンドを二重引用符で囲んでください。

```
"D:\Program Files\IBM\WID7_WTE\runtimes\bi_v7\bin\wsadmin"
-f wid-install.jacl -profileName qwps -username admin -password admin
```

- デフォルトのユーザーは「admin」で、デフォルトのパスワードも「admin」です。
  - デフォルトの IBM Process Server プロファイル名は qwps です。
  - デフォルトの WebSphere ESB Server プロファイル名は qesb です。
  - IBM Integration Designer にサーバーのプロファイル名を表示するには、「サーバー」ビューに切り替えてサーバーを右クリックし、「サーバー・プロファイルの管理」を選択します。
4. スクリプトが正常に終了すると、インストールされたサーバーのポート番号が表示されます。以下の図に示すように、ポート番号は 9080 です。

```

C:\> wsadmin server repository.
ADMA5053I: The library references for the installed optional package are created.
ADMA5005I: The application $Q$Sample is configured in the WebSphere Application S
erver repository.
ADMA5001I: The application binaries are saved in D:\Program Files\IBM\WID61\pf\w
ps\wstemp\Script11851898e2b\workspace\cells\widCell\applications\Q$Sample.ear\Q$
Sample.ear
ADMA5005I: The application $Q$Sample is configured in the WebSphere Application S
erver repository.
SECJ0400I: Successfully updated the application $Q$Sample with the appContextIDFor
Security information.
ADMA5011I: The cleanup of the temp directory for application $Q$Sample is complet
e.
ADMA5013I: Application $Q$Sample installed successfully.

SAMP INFO: The install command for ESBSample completed successfully.

-----
Save
-----

Summary
-----

+++++
Port number is 9080.
+++++

The install command succeeded for the following Samples:

    ESBSample

All commands ran successfully.
  
```

図のとおりポート番号が 9080 の場合は、以下の手順をスキップできます。

5. ポートが 9080 でない場合は、株価情報アプリケーションがデータベースを検索するポートが正しいことを確認する必要があります。これはインポートに影響します。ポートを変更するには、以下の処理を行います。
  - a. 株価情報モジュールのアセンブリー・ダイアグラムを開きます。
  - b. DelayedService インポートを選択して、「プロパティ」ビューで「バインディング」をクリックします。
  - c. 「アドレス」テキスト・ボックスには、`http://localhost:9080/DelayedService/services/DelayedServiceSOAP` などの値が挿入されています。

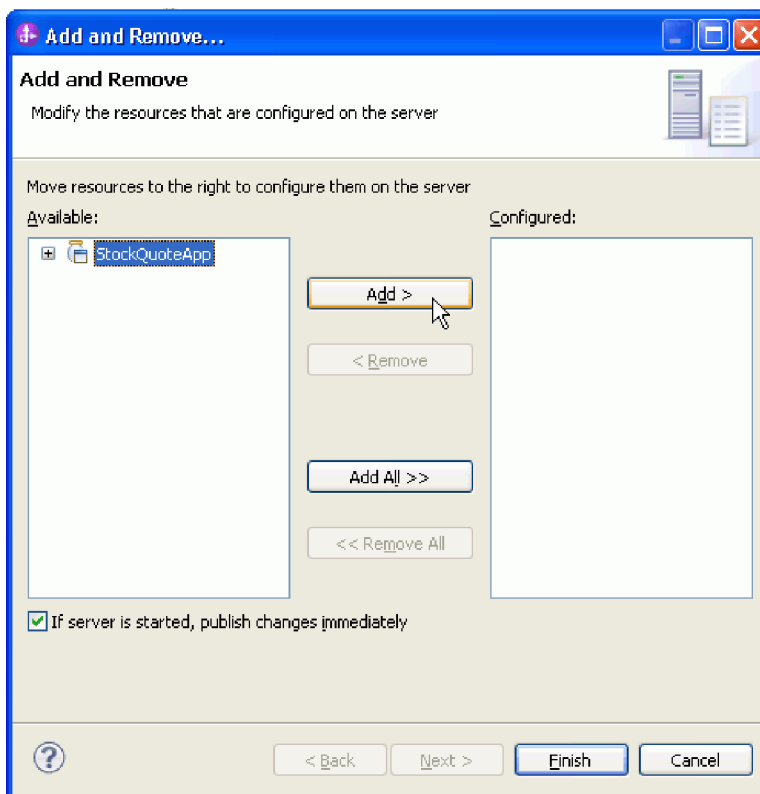
- d. 「9080」をスクリプトの実行時に表示されたポート番号で上書きします。
- e. RealtimeService インポートでも前の手順を実行し、その後モジュールを保存します。
6. スクリプトが正常にインストールされたら、サーバーを右クリックし、「再始動」を選択して再始動します。

---

## メディエーション・フローをテストします。

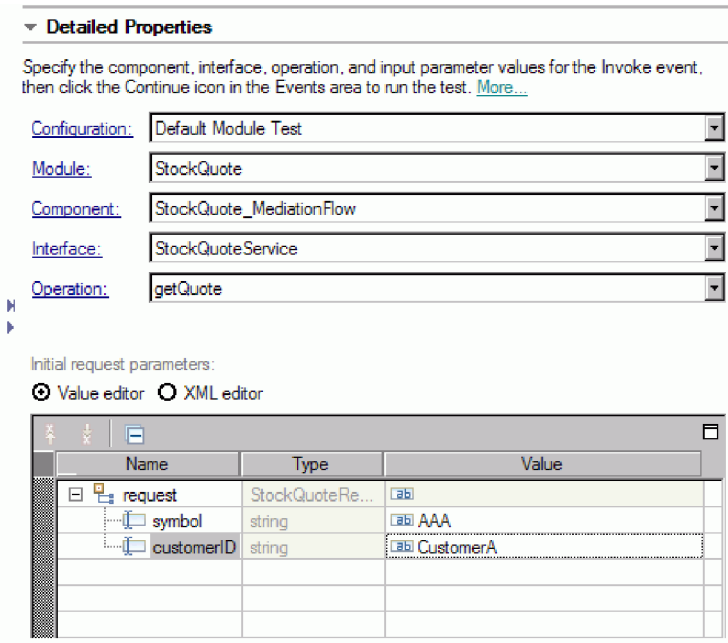
スクリプトを実行したら、稼働中のサーバーに StockQuoteApp を追加する必要があります。その後、テスト・クライアントでメディエーション・フローをテストできます。

1. サーバーに StockQuoteApp プロジェクトを追加するには、サーバーを右クリックして、「プロジェクトの追加と削除」を選択します。「使用可能プロジェクト」リストから「StockQuoteApp」プロジェクトを選択し、「追加>」をクリックしてそれを「構成済みプロジェクト」リストに加えます。「完了」をクリックし、サーバーが公開を終了するまで待機します。



2. StockQuote メディエーション・モジュールのアセンブリ・ダイアグラムを開きます。「StockQuote\_MediationFlow」コンポーネントを右クリックして、「コンポーネントのテスト」を選択します。統合テスト・クライアントの「イベント」ページが開きます。
3. 「イベント」ページでは、テストするモジュール、コンポーネント、インターフェース、および操作を選択できます。このサンプルでは、「詳細プロパティ」が以下であることを確認してください。
  - 構成: デフォルト・モジュール・テスト
  - モジュール: StockQuote
  - コンポーネント: StockQuote\_MediationFlow
  - インターフェース: StockQuoteService
  - 操作: getQuote

- 「初期要求パラメーター」テーブルで、「値」列のセルをダブルクリックして情報を入力します。「symbol」行の値セルをダブルクリックして **AAA** と入力します。同じように、「customerID」に **CustomerA** と入力します。



▼ Detailed Properties

Specify the component, interface, operation, and input parameter values for the Invoke event, then click the Continue icon in the Events area to run the test. [More...](#)

Configuration: Default Module Test

Module: StockQuote

Component: StockQuote\_MediationFlow


Interface: StockQuoteService

Operation: getQuote

Initial request parameters:

Value editor  XML editor

Name	Type	Value
request	StockQuoteRe...	[abi]
symbol	string	[abi] AAA
customerID	string	[abi] CustomerA

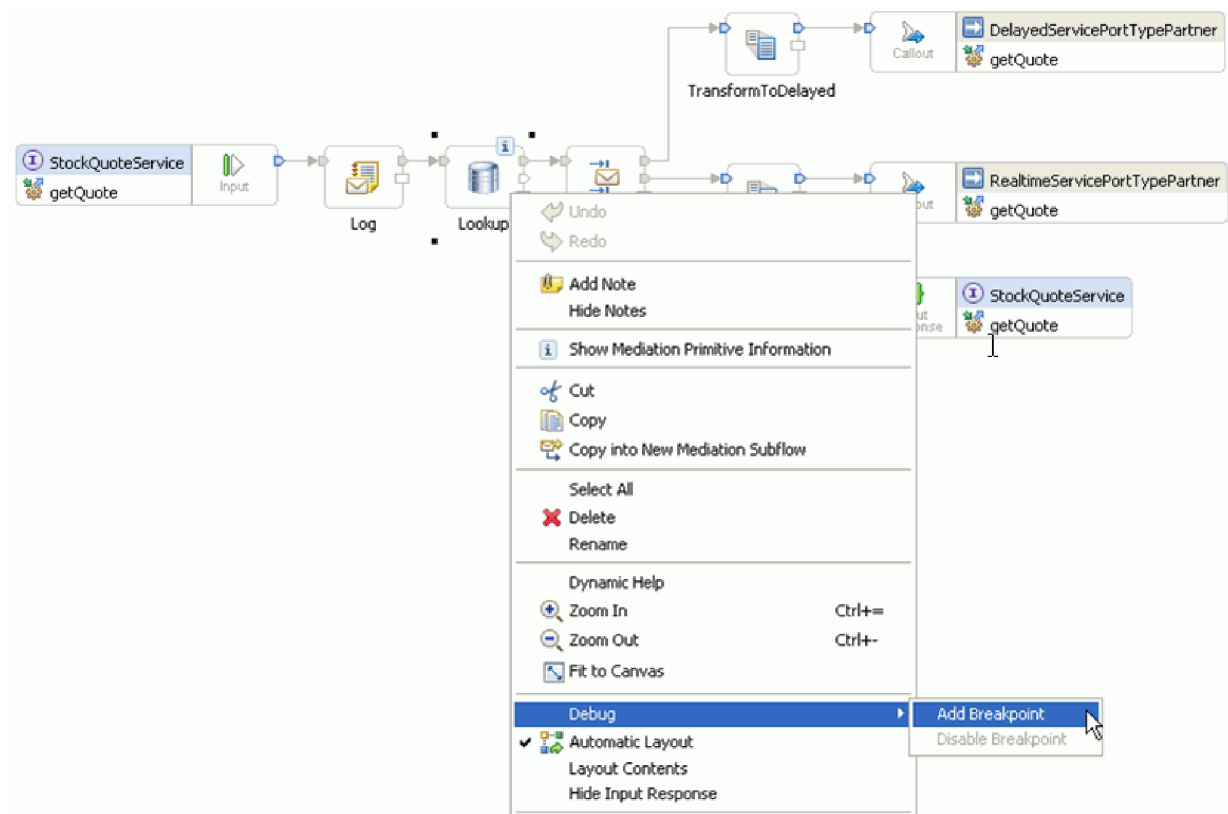
- 次に、「続行」ボタン  を押して getQuote 操作を呼び出します。「デプロイメント・ロケーション」ダイアログが開きます。
- サーバーを選択して「完了」をクリックします。IBM Process Server または WebSphere Enterprise Service Bus のいずれかでメディエーション・フローを実行できます。サーバーのユーザー ID およびパスワードを入力します。デフォルトでは admin/admin になります。
- qualityOfService の結果の値と戻りパラメーターの値を見ることができます。
- アプリケーションが正常に実行しない場合は、以下のことを試してください。
  - サービスで実行時例外が発生している場合は、Web サービス・アプリケーションが開始されていることを確認します。「サーバー」ビューでサーバーを選択し、「**Run Administration Console (管理コンソールの実行)**」を右クリックで選択してログインします。「**アプリケーション**」>「**エンタープライズ・アプリケーション**」の下の実行中のアプリケーションのリストで、SQSample アプリケーションの状況を確認し、必要ならばそのアプリケーションを開始します。
  - モジュール StockQuote が見つからないというエラーが表示された場合は、上記の説明に従って、StockQuoteApp アプリケーションが実行されていることを確認します。

## メディエーション・フローのデバッグ

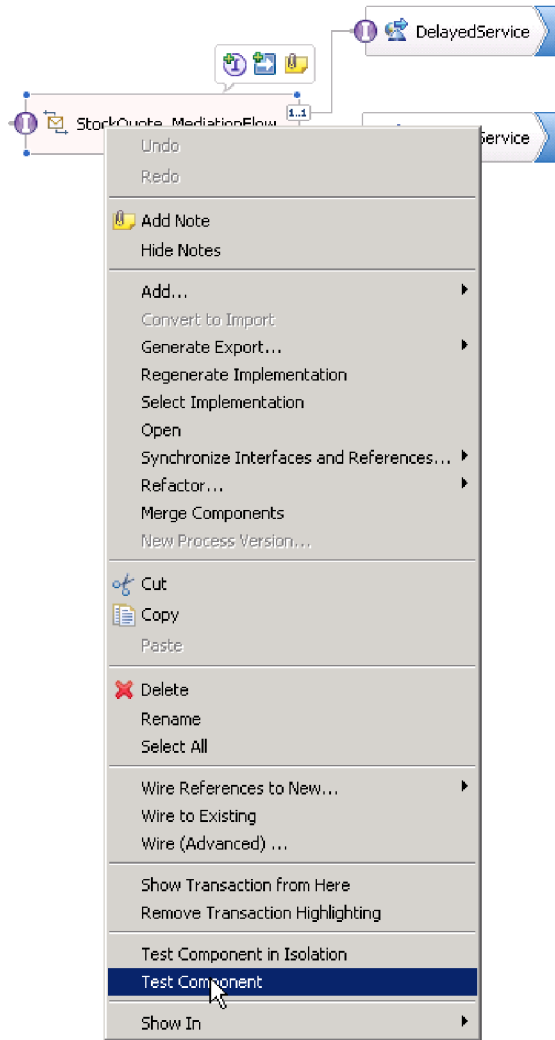
統合デバッガーを使用して、単体テスト環境で StockQuote\_MediationFlow コンポーネントをデバッグします。

以下の手順では、ランタイム・コンポーネントが既にインストールされていることが前提となっています。これらのコンポーネントのインストール方法については、37 ページの『ランタイム・コンポーネントのインストール』を参照してください。

1. メディエーション・プリミティブにブレークポイントを追加するには、要求または応答フロー・キャンバスでプリミティブ・ノードを右クリックし、「デバッグ」>「ブレークポイントの追加」を選択します。ブレークポイントが追加されたことを示す小さな青色のアイコンが、ノードの左上隅に追加されます。



2. サーバーを選択して右クリックし、「Restart in Debug」を選択してデバッグ・モードでサーバーを再始動します。サーバーが既に始動している場合は、サーバーを停止してからデバッグ・モードで始動し直す必要があることに注意してください。
3. サーバーが始動したら、StockQuote メディエーション・モジュールのアセンブリー・ダイアグラムを開きます。「StockQuote\_MediationFlow」コンポーネントを右クリックして、「コンポーネントのテスト」を選択します。



単体テスト環境が開きます。

4. 単体テスト環境では、テストするモジュール、コンポーネント、インターフェース、および操作を選択できます。このサンプルでは、「詳細プロパティ」が以下であることを確認してください。
  - a. 構成: デフォルト・モジュール・テスト
  - b. モジュール: **StockQuote**
  - c. コンポーネント: **StockQuote\_MediationFlow**
  - d. インターフェース: **StockQuoteService**
  - e. 操作: **getQuote**
5. 「初期要求パラメーター」テーブルで、「値」列のセルをダブルクリックして情報を入力します。「symbol」行の値セルをダブルクリックして **AAA** と入力します。同じように、「customerID」に **CustomerA** と入力します。

**▼ Detailed Properties**

Specify the component, interface, operation, and input parameter values for the Invoke event, then click the Continue icon in the Events area to run the test. [More...](#)

Configuration:

Module:

Component:



Interface:

Operation:

Initial request parameters:

Value editor  XML editor

Name	Type	Value
request	StockQuoteRe...	[abi]
symbol	string	[abi] AAA
customerID	string	[abi] CustomerA

- 次に、「**続行**」ボタン  を押します。「デプロイメント・ロケーション」ダイアログが開きます。
- 先ほどデバッグ・モードで始動したサーバーを選択し、「**完了**」をクリックします。サーバーのユーザー ID およびパスワードを入力します。デフォルトでは admin/admin になります。フローがブレークポイントに到達すると、デバッグ・パースペクティブを開くよう促すプロンプトが出されます。「**はい**」をクリックしてパースペクティブを開きます。
- このビューで、メッセージ・エレメントの値、取得されたパス、および到達したブレークポイントを参照することができます。フローを最後まで、または次のブレークポイントまで続行するには、「**再開**」ボタン  をクリックします。

## 実行時のサービスの品質の変更

サンプルのビルド時に、実行時にプロパティの値を変更できるようにフィルター・パターン・プロパティをプロモートしました。ここでは、単体テスト環境の管理コンソールでフィルター・パターン・プロパティの値を変更し、要求が別のサービスに送信されるようにします。結果はテスト・クライアント・エミュレーターに表示されます。

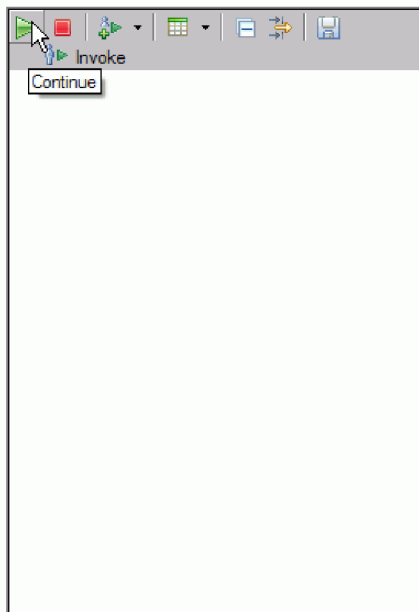
このタスクで説明されているアクションを実行するには、サーバー・スクリプトを実行してランタイム・コンポーネントをインストールする必要があります。手順については、37 ページの『ランタイム・コンポーネントのインストール』を参照してください。

StockQuote アセンブリー・ダイアグラムを開き、以下の手順に従って、プロモートされた Filter プリミティブのプロパティをテストします。

- 「StockQuote\_MediationFlow」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。
- 「イベント」ページで以下に示す初期要求パラメーターを入力して「**続行**」をクリックします。
  - symbol について、AAA と入力します。
  - customerID について、CustomerB と入力します。

## Events

This area displays the events in a test trace. Select an event to display its properties in the General Properties and Detailed Properties sections. [More...](#)



## Detailed Properties

Specify the component, interface, operation, and input parameter values for the Invoke event, then click the Continue icon in the Events area to run the test. [More...](#)

Configuration:

Module:

Component:

Interface:

Operation:

Initial request parameters:

Value editor  XML editor

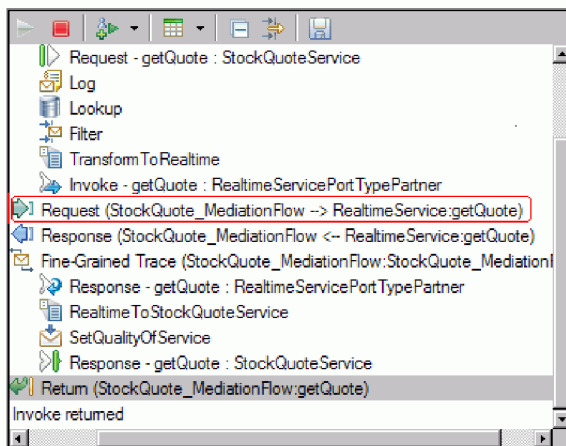
Name	Type	Value
request	StockQuoteService	[abi]
symbol	string	[abi] AAA
customerID	string	[abi] CustomerB

3. 単体テスト環境用のサーバーを選択します。「完了」をクリックし、サーバーにログインするためのユーザー名およびパスワードを入力します。デフォルトでは admin/admin になります。

エミュレーターに表示された結果から、呼び出されるサービスが RealtimeService であることがわかります。qualityOfService スtringの値を見て、この顧客のサービス・レベルが上級であることも確認できます。

## Events

This area displays the events in a test trace. Select an event to display its properties in the General Properties and Detailed Properties sections. [More...](#)



## General Properties

### Detailed Properties

Module: [StockQuote](#)

Component: [StockQuote\\_MediationFlow](#)

Interface: [StockQuoteService](#)

Operation: [getQuote](#)

Return parameters:

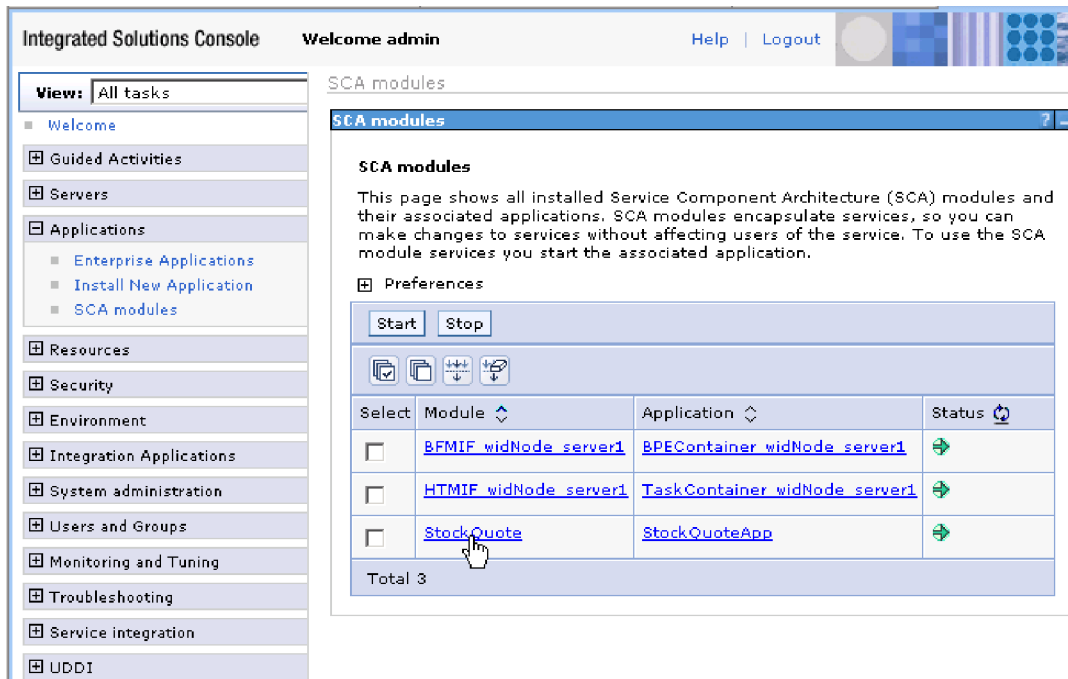
Value Editor  XML Source

Name	Type	Value
response	StockQuoteRes...	[abi]
value	float	[abi] 27.792974
qualityOfService	string	[abi] premium

4. 「サーバー」ビューに切り替えます。サーバーを右クリックして、「管理」>「管理コンソールの実行 (Run administrative console)」を選択します。ログイン・ウィンドウで、ユーザー ID とパスワード (デフォルトは admin/admin) を入力します。「ログイン」をクリックします。
5. 管理コンソールで、「アプリケーション」を展開して「SCA モジュール」をクリックします。



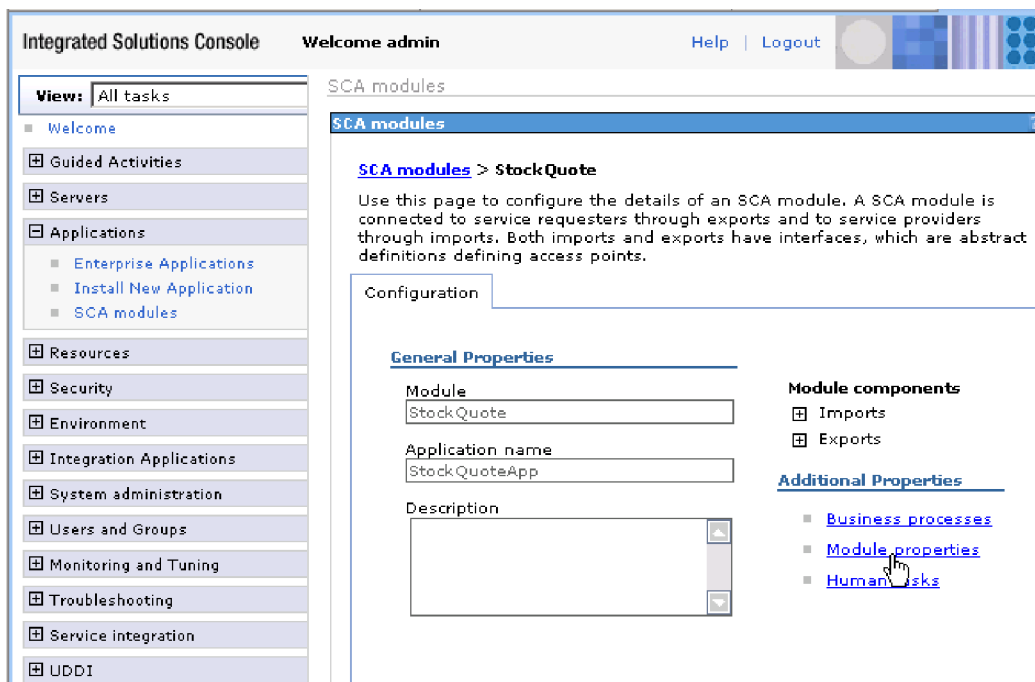
6. アプリケーションのリストで、**StockQuote** をクリックします。



The screenshot shows the 'Integrated Solutions Console' interface. On the left is a navigation tree with categories like 'Applications', 'Resources', and 'Security'. The main content area is titled 'SCA modules' and contains a table of installed modules. A mouse cursor is pointing at the 'StockQuote' entry in the table.

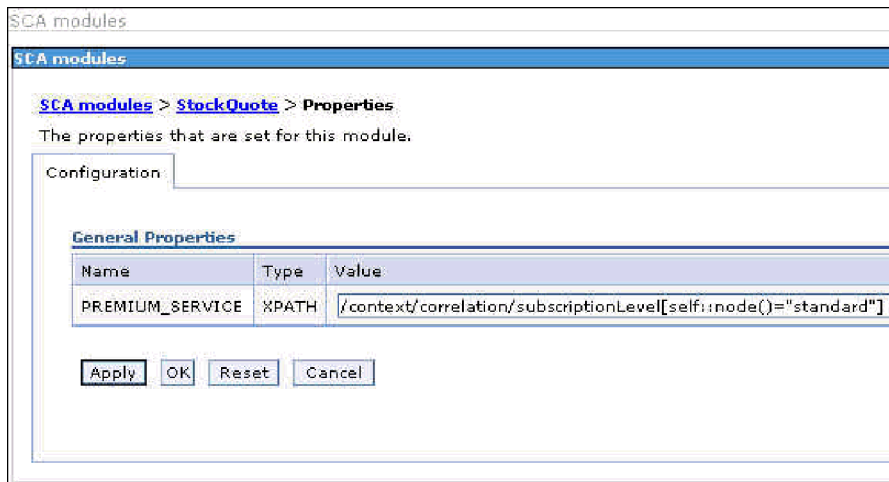
Select	Module	Application	Status
<input type="checkbox"/>	<a href="#">BFMIF_widNode_server1</a>	<a href="#">BPEContainer_widNode_server1</a>	
<input type="checkbox"/>	<a href="#">HTMIF_widNode_server1</a>	<a href="#">TaskContainer_widNode_server1</a>	
<input type="checkbox"/>	<a href="#">StockQuote</a>	<a href="#">StockQuoteApp</a>	

7. 「モジュール・プロパティ」をクリックします。

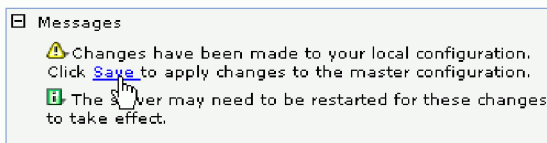


The screenshot shows the 'SCA modules > StockQuote' configuration page. It includes a 'Configuration' tab and a 'General Properties' section with fields for 'Module', 'Application name', and 'Description'. On the right, there are sections for 'Module components' (Imports, Exports) and 'Additional Properties' with links to 'Business processes', 'Module properties', and 'Human tasks'. A mouse cursor is pointing at the 'Module properties' link.

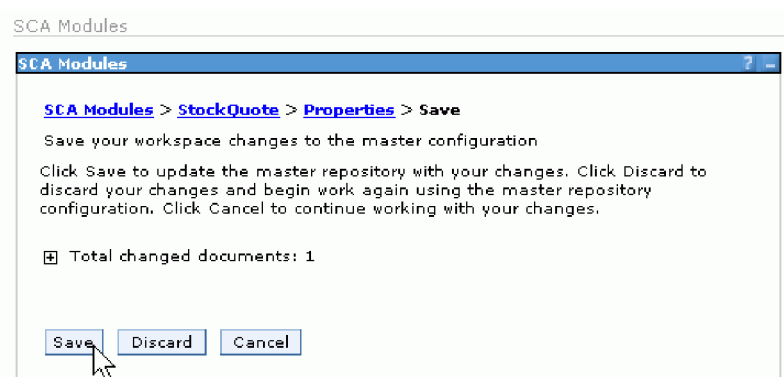
8. 前にプロモートしたプロパティが表示されます。別名は PREMIUM\_SERVICE です。PREMIUM\_SERVICE の値フィールドをクリックして、「上級」を「標準」に変更します。「適用」をクリックします。




9. メッセージ・ウィンドウで、「保存」をクリックします。



10. 「SCA モジュール」ウィンドウで、「保存」ボタンをクリックします。



11. テスト・クライアントの「イベント」ページに切り替えます。ページ上部の左側にある「呼び出し」 をクリックします。

12. シンボルの値は AAA のままにして、customerID の値に上級サービスの顧客を意味する CustomerB を入力します。「続行」をクリックします。

「イベント」エリアに表示された結果から、呼び出されるサービスが DelayedService であることがわかります。

## Events

Invoke (StockQuote\_MediationFlow:getQuote)  
  Invoke started  
  Invoke returned  
Invoke (StockQuote\_MediationFlow:getQuote)  
  Invoke started  
  Invoke (StockQuote\_MediationFlow:getQuote)  
    Request (StockQuote\_MediationFlow --> DelayedService:getQuo...)  
    Response (StockQuote\_MediationFlow <-- DelayedService:getQu...)  
    Return (StockQuote\_MediationFlow:getQuote)  
  Invoke returned

## General Properties

### Detailed Properties

Module: [StockQuote](#)  
Component: [StockQuote\\_MediationFlow](#)  
Interface: [StockQuoteService](#)  
Operation: [getQuote](#)

Return parameters:

Name	T...	Value
response	Stoc...	✓
value	float	✓ 30.64931
qualityOfService	string	✓ premium



---

## 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 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、[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。

Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Intel Centrino、Intel Centrino ロゴ、Celeron、Intel Xeon、Intel SpeedStep、Itanium、および 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.