

# IBM WebSphere Process Server 或 IBM WebSphere Enterprise Service Bus 中使用的 JMS 上 SOAP 的 Web 服務，第一部分：使用 SIBus JMS 提供者

級別：中級

[Fenglian Xu \(xufengli@uk.ibm.com\)](mailto:xufengli@uk.ibm.com)，IBM 軟體工程師

[Simon Holdsworth \(Simon\\_Holdsworth@uk.ibm.com\)](mailto:Simon_Holdsworth@uk.ibm.com)，IBM 資深技術人員

2008 年 5 月 22 日

這篇由兩部分組成的文章說明，如何在 IBM® WebSphere® Process Server 與 IBM WebSphere Enterprise Service Bus 中使用的 Java™ Message Service (JMS) 上的 SOAP。學習如何藉由 IBM WebSphere Integration Developer 工具，來設定與使用依預設配置的 JMS 上的 SOAP，還有如何透過配置啟用 IBM WebSphere MQ JMS 提供者。在本文即系列文章的第一部分中，您將建立與呼叫使用 JMS 上的 SOAP Web 服務以及端對端應用程式範例，包含完全的應用程式建立、建置、部署與測試程序。在整個過程當中，您將會經歷包含點對點與發佈/訂閱傳訊在內的實務練習。在此系列的第二篇文章中，您將配置使用 JMS 上的 SOAP 通訊協定的 Web 服務，以啟用 WebSphere MQ 做為 JMS 提供者，並且容許透過 WebSphere MQ 佇列傳輸 SOAP 訊息。

## 前言

一般而言，HTTP 是一種用來遞送 Web 服務訊息的不可靠通訊協定。因此，當您需要比較高的可靠性時，您可能需要使用替代的通訊協定。而現在有越來越多人使用現有或撰寫全新的 JMS 應用程式，來處理 SOAP 訊息安全載荷。WebSphere Process Server 與 WebSphere Enterprise Service Bus 皆支援使用 JMS 來遞送 Web 服務訊息，以提高可靠性並且整合使用 SOAP 通訊協定的 JMS 應用程式。

此支援是以 Web 服務連結的形式提供，您可以套用至 WebSphere Process Server 與 WebSphere Enterprise Service Bus 的匯出（服務提供點）與匯入（服務呼叫點）模組。您可以配置連結，使用 HTTP 或 JMS 來傳輸 SOAP 訊息。當使用 Web 服務連結匯出連線至元件時，執行時期會自動處理 Web 服務通訊協定，並且將 SOAP 訊息轉換成元件所需的格式。同樣地，當元件連線至使用 Web 服務連結匯入時，執行時期會自動處理 Web 服務的呼叫。

當選取 JMS 上的 SOAP 供 Web 服務連結使用時，WebSphere Application Server 預設的 JMS 提供者（也稱為 Service Integration Bus (SIBus) JMS 提供者）將會用來建立必要資源，以提供或呼叫 Web 服務。WebSphere Integration Developer 可讓整合開發者在無須撰寫複雜程式碼的情況下，輕鬆建置與呼叫使用 JMS 上的 SOAP 的 Web 服務。使用 SIBus JMS 提供者的 WebSphere Integration Developer 會自動產生 Web 服務資源「Java 命名和目錄介面 (JNDI)」名稱。

## 軟體需求

您需要在機器上安裝下列軟體，才能繼續追隨本文：

- WebSphere Integration Developer 6.02 或 6.1 版
- WebSphere Process Server 6.02 或 6.1 版，或者 WebSphere Enterprise Service Bus 6.02 版或 6.1 版

範例會使用 6.02 與 6.1 版分別進行開發與測試。WebSphere Integration Developer 6.1 版提供[可下載的專案交換檔](#)。

## 傳訊模型

JMS 應用程式採用兩種方式之一進行互動：

- **點對點**：當存在兩個彼此直接互動的應用程式時，其傳訊模型稱為點對點，而所使用的資源為**佇列**。
- **發佈/訂閱**：當存在一個職司生產資訊的應用程式集以及另一個職司使用該資訊的應用程式集時，其傳訊模型為發佈/訂閱，而所使用的 JMS 資源為**主題**。

範例應用程式將會示範點對點與發佈/訂閱傳訊模型，以便您瞭解在 WebSphere Integration Developer 中建置使用 JMS 上的 SOAP Web 服務時，需要哪些 JMS JNDI 資源。

自動產生的 JNDI 名稱與其對應資源是用於點對點傳訊模型。若要使用發佈/訂閱傳訊模型，就必須在部署描述子和管理主控台中，同時配置 JNDI 名稱與其對應資源。

## 點對點傳訊模型的應用情境

此情境將示範由使用 JMS 上的 SOAP Web 服務連結之模組所提供的 Web 服務，以及使用相同連結之另一個模組所呼叫的 Web 服務。在此情況下，在傳送者與接收者使用相同 JMS 目的地之處，將會使用點對點傳訊模型。有一個 JMS 目的地會用於要求訊息，而另一個則用於回應訊息，如圖 1 所示。

圖 1：使用 JMS 上的 SOAP 連結的點對點實務練習範例

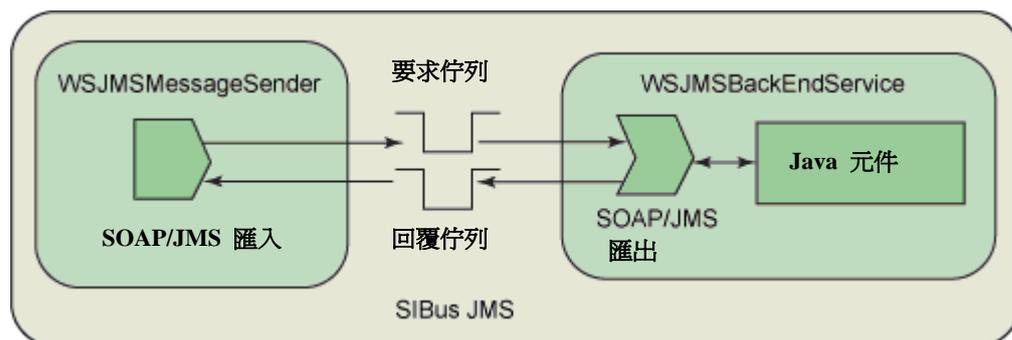


圖 1 顯示採用點對點傳訊模型的 JMS 上的 SOAP 使用範例。Web 服務應用程式 WSJMSBackEndService 包括使用 Web 服務 JMS 上的 SOAP 連結匯出與 Java 元件。該匯出會控制在接收與傳送 JMS 上的 SOAP 訊息時的 Web 服務處理，並且透過其介面呼叫 Java 元件。Java 元件提供後端服務實作；它不會察覺正在使用 JMS 上的 SOAP 通訊協定來呼叫它。

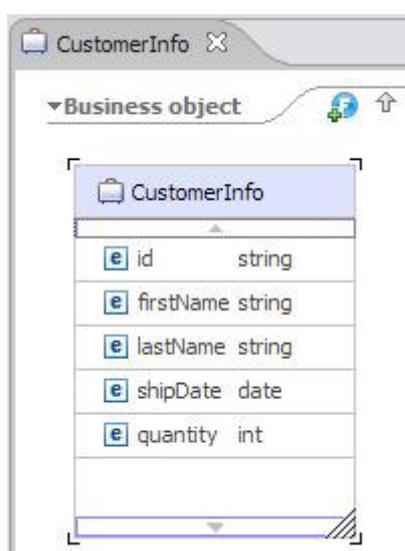
WSJMSMessageSender 應用程式具有使用 Web 服務 JMS 上的 SOAP 連結匯入，可藉由將訊息放置在要求目的地，以及從回覆目的地讀取訊息來呼叫 WSJMSBackEndService。要求訊息指出在必要時使用 JMSReplyTo 頁首欄位的回應目的地。

## 為點對點傳訊建立使用 JMS 上的 SOAP Web 服務

首先建立含共用介面的檔案庫，供本文中的應用程式使用。此檔案庫有一個名為 `CustomerService` 的介面，其中內含三項作業。請遵循這些步驟，在 `WebSphere Integration Developer` 中建立介面與商業物件 (BO)：

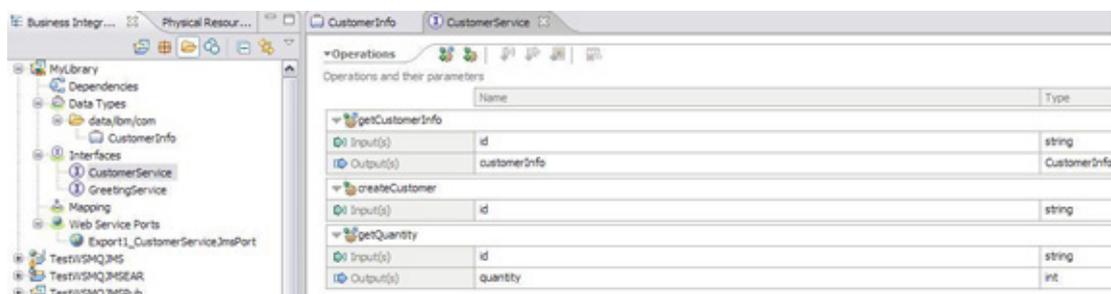
1. 若要在商業整合視景中建立檔案庫，請選取 **New > Library** 以啟動精靈，然後鍵入 `MyLibrary` 做為名稱。按一下 **Finish**。
2. 若要在 `MyLibrary` 中建立 BO，請選取 **Data Type > New > Business Object**。在圖 2 所示的欄位中輸入資料。

圖 2：具有五個欄位的 BO 範例



1. 若要建立供 `MyLibrary` 呼叫的介面，按一下滑鼠右鍵 **Interface > New**，並且在精靈中輸入 `CustomerService` 做為名稱。按一下 **Finish**。
2. 在 `CustomerService` 中建立三項作業，如圖 3 所示（請參閱圖 3 的放大版）。

圖 3：含三項作業的介面範例



在範例應用程式中，`CustomerService` 介面用於 `Service Component Architecture (SCA)` 元件。

現在請遵循接下來的步驟，以便在 `WebSphere Integration Developer` 中建立使用 `Java` 元件實作之 `JMS`

上的 SOAP 的 Web 服務：

1. 要建立新模組，請按一下 **File > New Module**，並且在精靈中輸入 `WSJMSBackEndService` 做為模組名稱。按一下 **Finish**。
2. 建立 JMS 上的 SOAP 匯出，將 `Export` 圖示拖曳至  組合圖中，並且產生含預設值的 Web 服務 JMS 上的 SOAP 連結。
3. 建立名為 `HandleRequest` 的 Java 元件，其中含介面 `CustomerService`。
4. 連線 `BackEndSOAPJMSEXP` 與 `HandleRequest` 元件，如圖 4 所示。

圖 4：WSJMSBackEndService 應用程式



在 `HandleRequest` 元件上用滑鼠右鍵按一下以產生實作。清單 1 顯示實作程式碼的範例。

#### 清單 1：HandleRequest 實作的 Snippet

```
public DataObject getCustomerInfo(String id) {
    ServiceManager serviceManager = ServiceManager.INSTANCE;
    BOFactory bof = (BOFactory) serviceManager
        .locateService("com/ibm/websphere/bo/BOFactory");

    if( bof == null ) System.out.println("bof == null");

    DataObject myData = bof.create("http://MyLibrary/data/ibm/com",
        "CustomerInfo");
    System.out.println("Creating a CustomerInfo object with soap/jms -----");
    myData.setString("id", id);
    myData.setString("firstName", "Lisa");
    myData.setString("lastName", "Frank");
    Date shipDate = new Date();
    myData.setDate("shipDate", shipDate);
    myData.setInt("quantity", 40);

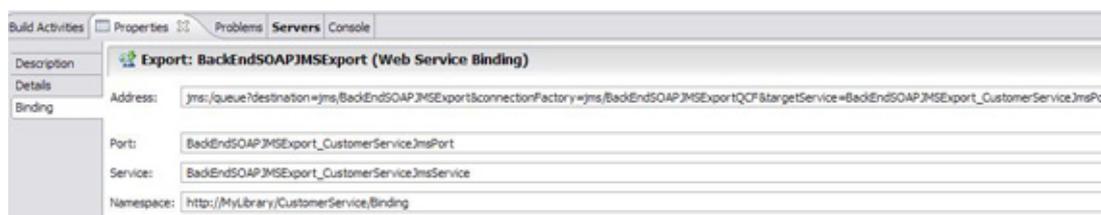
    System.out.println("the created customerInfo is returned:");
    printCustomerInfo(myData);
}
```

```
return myData;
}
```

`printCustomerInfo(myData)` 是可印出 `myData` 物件值的專用函數。

若要檢視匯出連結內容，可從組合圖選取 **BackEndSOAPJMSExport**，如圖 5 所示。

圖 5：JMS 上的 SOAP Web 服務匯出連結內容



位址內容包括以下 JMS 資源：

- 訊息類型
- 目的地
- 連線 Factory

當 WebSphere Integration Developer 中產生 JMS 上的 SOAP 連結時，這些資源的值就會自動產生並且儲存於 `MyLibrary` 下的 `BackEndSOAPJMSExport_CustimerServiceJmsPort.wsdl` 檔。目的地是用來接收要求訊息；連線 Factory 則是用來連接 JMS 提供者（在此為 SIBus JMS 提供者）以及存取目的地。現在您可以儲存及建立 `WSJMSBackEndService` 專案：

1. 將 **WSJMSBackEndService** 專案匯出至 `.ear` 檔，建立 `WSJMSBackEndService.ear` 並且從管理主控台執行安裝。
2. 從管理主控台安裝 **WSJMSBackEndService.ear** 檔，在每個步驟中採用預設值。您可能需要從管理主控台手動啟動已安裝的應用程式。

當 Web 服務 JMS 上的 SOAP 連結用於匯出時，它會使用訊息驅動 Bean (MDB) 來接收要求訊息。此 MDB 與所需資源將會在安裝期間，使用前述的 JMS 資源與 SIBus JMS 提供者加以建立。所建立的資源包括圖 6 所示的 JMS 啟動規格（請參閱圖 6 的放大版），以及供圖 7 中要求與回應訊息使用的連線 Factory（請參閱圖 7 的放大版）。

圖 6：為匯出 MDB 所建立的啟動規格

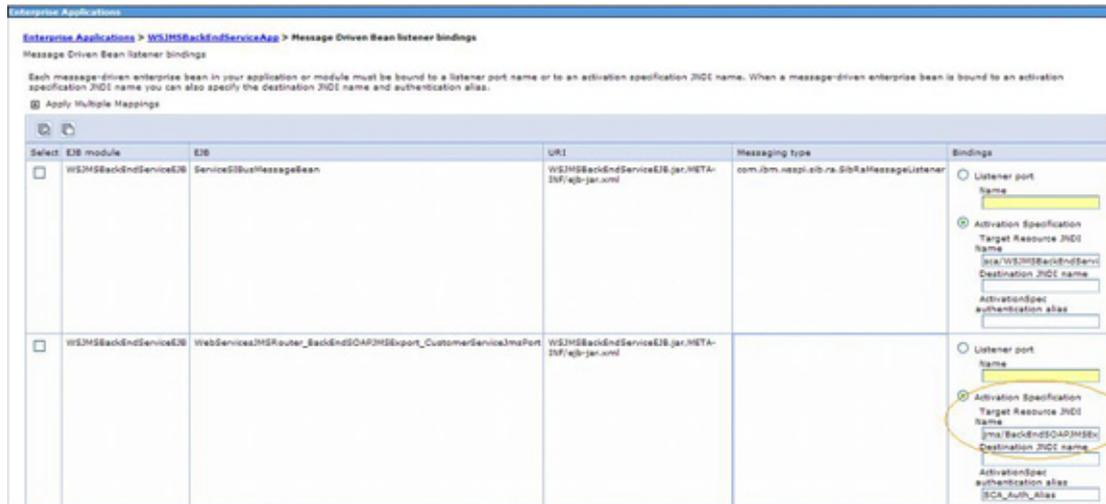
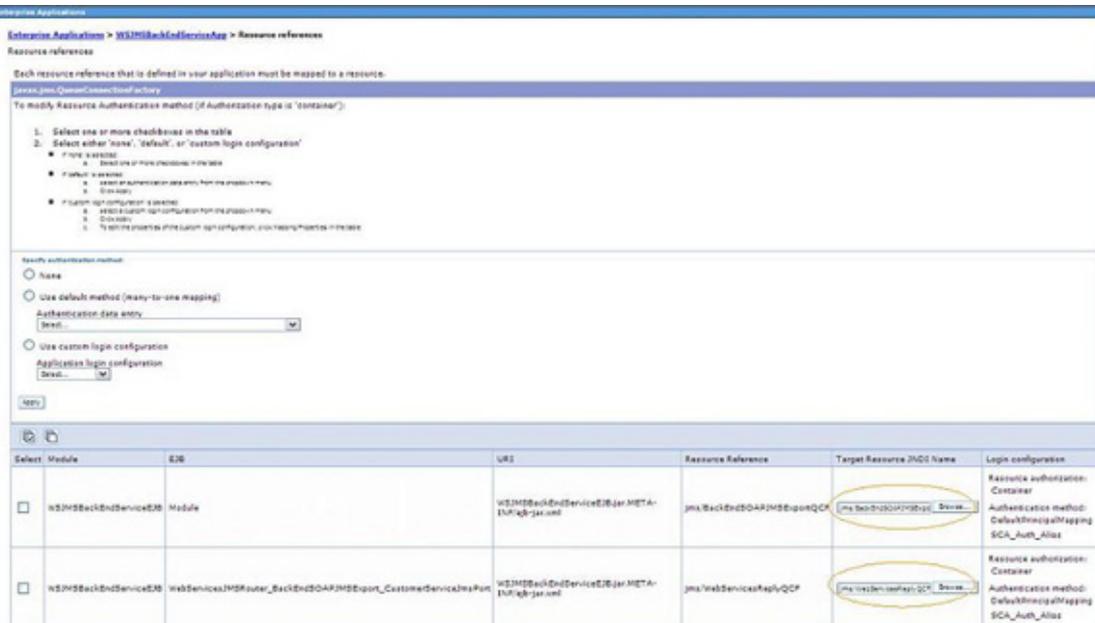


圖 7：適用於要求與回應連線 Factory 的 JNDI 資源

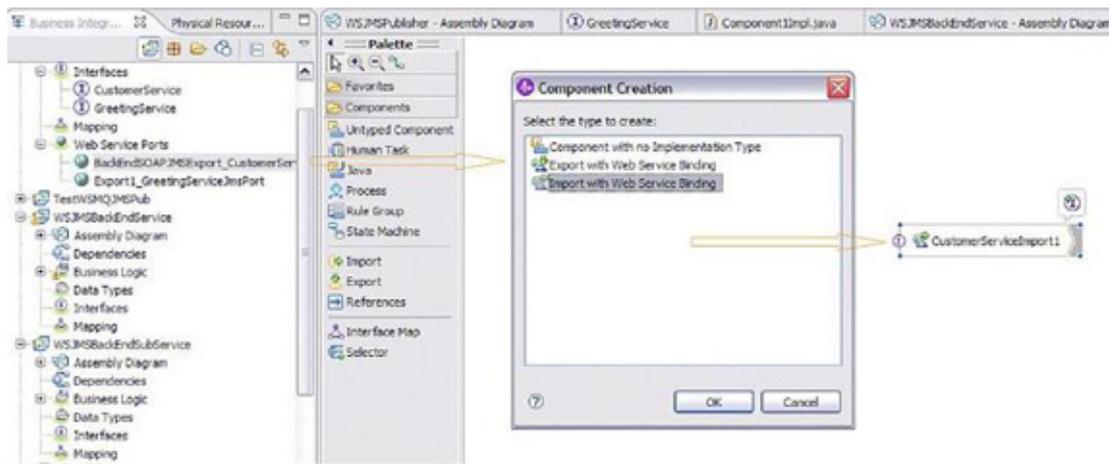


## 為點對點傳訊模型建立用戶端應用程式

在 WebSphere Integration Developer 中，您可以藉由使用 Web 服務 JMS 上的 SOAP 連結匯入，來建立 Web 服務用戶端應用程式：

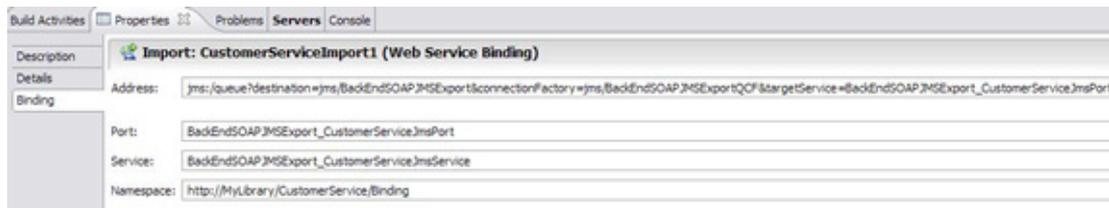
1. 建立名為 `WSJMMessageSender` 的新模組。
2. 在商業整合視景中，導覽至 `MyLibrary` 下的 **BackEndSOAPJMSExport\_CustomerServiceJmsPort**。
3. 將 **BackEndSOAPJMSExport\_CustomerServiceJmsPort** 拖放至組合圖中。
4. 在產生的對話框中，選取 **Import with Web Service Binding** 選項，以建立圖 8 所示的 `CustomerServiceImport1`。

圖 8：建立 MessageSender 應用程式的範例



您可以選取組合圖中的 **CustomerServiceImport1**，藉此檢視匯入連結內容，如圖 9 所示。

圖 9：JMS 上的 SOAP 匯入連結內容



附註：此位址內容和圖 5 中匯出的 JMS 上的 SOAP 連結位址內容是相同的。

匯入中的 destination 是用來傳送 JMS 上的 SOAP 要求訊息。匯入中的 connectionFactory 則是用來連接 JMS 提供者（在此為 SIBus JMS 提供者）以及存取要求與回覆目的地。

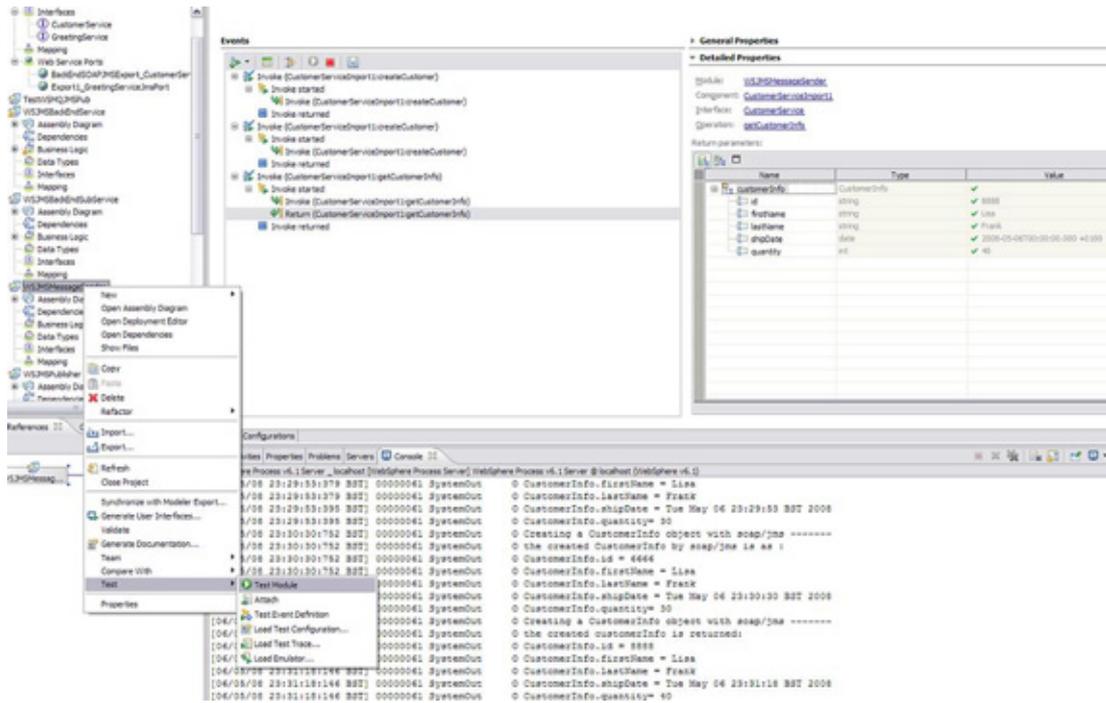
在部署模組時，不會為匯入建立任何 JMS JNDI 資源，因為它使用參照的匯出資源。

### 使用測試模組來測試 JMS 上的 SOAP WS 匯入應用程式

您可以使用測試模組功能，從 WebSphere Integration Developer 測試 WSJMSServiceSender。不過前提是您已在伺服器配置中，配置了 WebSphere Process Server 或 WebSphere Enterprise Service Bus 執行時期：

1. 從 WebSphere Integration Developer 啟動伺服器。
2. 將 **WSJMSServiceSender** 專案新增至伺服器。
3. 在 **WSJMSServiceSender** 上用滑鼠右鍵按一下以測試模組。
4. 選取要測試的作業，諸如 **getCustomerInfo**，並且輸入 ID 參數的值。
5. 按一下 **Run** 按鈕以呼叫 WSJMSServiceSender 中的 **getCustomerInfo**。隨即呼叫 **getCustomerInfo** 函數，如主控台視窗所示。傳回 **CustomerInfo** 物件並且顯示於測試視窗，如圖 10 所示。

圖 10：測試模組與呼叫結果



即使並未事先安裝，您還是可以測試 WSJMSPublisher 專案，因為如果尚未安裝 WSJMSPublisher，測試作業將會安裝並且啟動它。

## 發佈/訂閱傳訊模型的應用程式實務練習

使用發佈/訂閱模型時，許多應用程式可能正在接收要求訊息。如果所呼叫的作業是要求/回應（換言之，具有輸出或者在介面中定義了錯誤訊息），這些應用程式全都傳回回應，而且需要額外邏輯來選取某個回應或者聚集回應。這就是為何 WebSphere Process Server 與 WebSphere Enterprise Service Bus 只能支援發佈/訂閱傳訊模型使用單向作業。

發佈主題所屬訊息給感興趣訂閱者群組（請參閱圖 11）。

圖 11：發佈/訂閱傳訊模型的應用情境

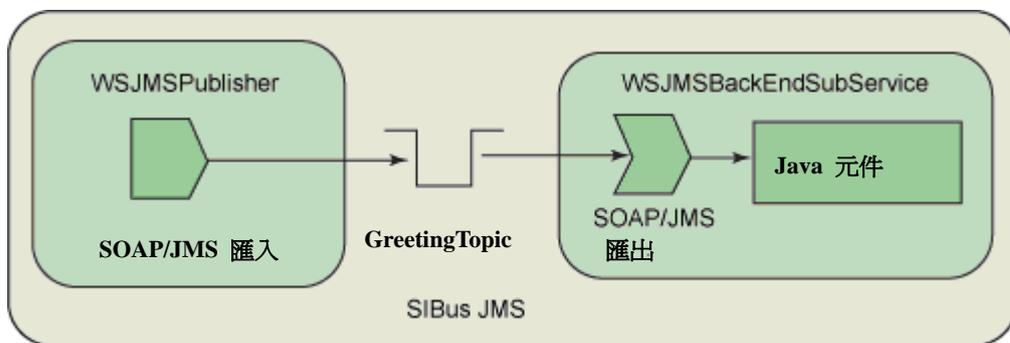


圖 11 顯示發佈/訂閱傳訊模型的應用情境。WSJMSPublisher 應用程式使用 greetingTopic 主題發佈訊息。擁有主題的 JMS 提供者將主題傳送給主題的訂閱者。在範例中只有一個訂閱者，即

WSJMSBackEndSubService 應用程式。下列章節說明如何實作此情境。

## 為發佈/訂閱模型建立含單向作業的介面

1. 在 MyLibrary 下建立名為 GreetingService 的新介面。
2. 建立單向作業：`void sayHello(string message);`

您可以使用點對點傳訊模型的自動產生 SIBus JMS 資源。在發佈/訂閱傳訊模型中，您必須從管理主控台配置及建立發佈/訂閱 SIBus JMS 資源。

## 為發佈/訂閱模型建立 SIBus JMS 資源

在安裝應用程式之前，您必須在 Node 範圍中建立一些 SIBus JMS 資源。

1. 建立名為 greetingTopicCF 的 Topic Connection Factory，方法是選取 **JMS > Topic connection factory**。
2. 藉由選取 **JMS > Topics** 來建立名為 greetingTopic 的主題。
3. 藉由選取 **JMS > Activation specifications** 來建立名為 topicJMSAS 的啟動規格。

表 1 顯示當您在管理主控台中建立資源時，所需定義的資源內容。

表 1：發佈/訂閱模型所需的 SIBus JMS 資源

SIBus JMS 資源	Topic Connection Factory	主題	啟動規格
名稱	greetingTopicCF	greetingTopic	topicJMSAS
JNDI 名稱	jms/greetingTopicCF	jms/greetingTopic	jms/topicJMSAS
匯流排名稱	SCA.APPLICATION.hostNode01Cell		

Topic Connection Factory 用於發佈者的 Web 服務 JMS 上的 SOAP 匯入連結，以便連線 SIBus JMS 提供者執行訊息發佈。匯流排名稱中最後的 hostNode01Cell，在每部機器上都不同。

對於主題，您還需要提供名稱與主題空間，如圖 12 所示。啟動規格將會用到主題，您可以在該處使用預設的主題空間。

圖 12：在管理主控台中配置主題

Topics

Topics > greetingTopic

A JMS topic is used as a destination for publish/subscribe messaging. Use topic destination administrative objects to manage JMS topics for the default messaging provider.

Configuration

**General Properties**

Administration

Scope: Node=L3D4047Node02

Provider: Default messaging provider

Name: greetingTopic

JNDI name: jms/greetingTopic

Description: A topic for JMS messages

**Connection**

Topic name: Greetings

Bus name: SCA.APPLICATION.L3D4047Node02Cell.Bus

Topic space: Default.Topic.Space

JMS delivery mode: Application

Time to live: milliseconds

Message priority:

**Advanced**

Read ahead: Inherit from connection factory

Apply OK Reset Cancel

**Related Items**

= Buses

對於啓動規格，您還需要指定目的地類型與目的地 JNDI 名稱，如圖 13 所示。

圖 13：在管理主控台中配置啓動規格

### Activation specifications > topicJMSAS

A JMS activation specification is associated with one or more message-driven beans and provides the configuration necessary for them to receive messages.

Configuration

#### General Properties

**Administration**

Scope  
Node=L3D4047Node02

Provider  
Default messaging provider

Name  
topicJMSAS

JNDI name  
jms/topicJMSAS

Description  
a topic JMS activation specifications

**Destination**

Destination type  
Topic

Destination JNDI name  
jms/jmsTopicDest

Message selector

Bus name  
SCA.APPLICATION.L3D4047Node02Cell.Bus

Acknowledge mode  
Auto-acknowledge

Target

Target type  
Bus member name

Target significance  
Preferred

Target inbound transport chain

**Additional**

Authentication alias  
(none)

#### Related Items

- JMS - J2C authentication data
- Buses

## 為訂閱主題建立使用 JMS 上的 SOAP 的 Web 服務

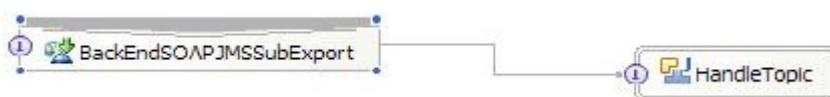
訂閱者就是訊息消費者，他們就特定主題接收由訊息發佈者所發佈的訊息。

在 WebSphere Process Server 與 WebSphere Enterprise Service Bus 中，包含使用 Web service JMS 上的 SOAP 連結匯出的模組是訂閱者。屬於匯出連結實作之部分的 MDB 會從主題目的地接收訊息，並且將它們轉換成其連線元件所需的格式。

具有使用 Web service JMS 上的 SOAP 連結匯出的模組則是職司發佈主題所屬訊息的發佈者。

建立模組 WSJMSBackEndSubService（包括使用 Web 服務 JMS 上的 SOAP 連結匯出與 Java 元件）的步驟，和點對點傳訊模型中的所述步驟完全相同。模組如圖 14 所示。

圖 14：配置匯出 MDB 內容



建立專案之後，您需要在 WebSphere Integration Developer 中建置它。建置程序所建立的構件，必須修改

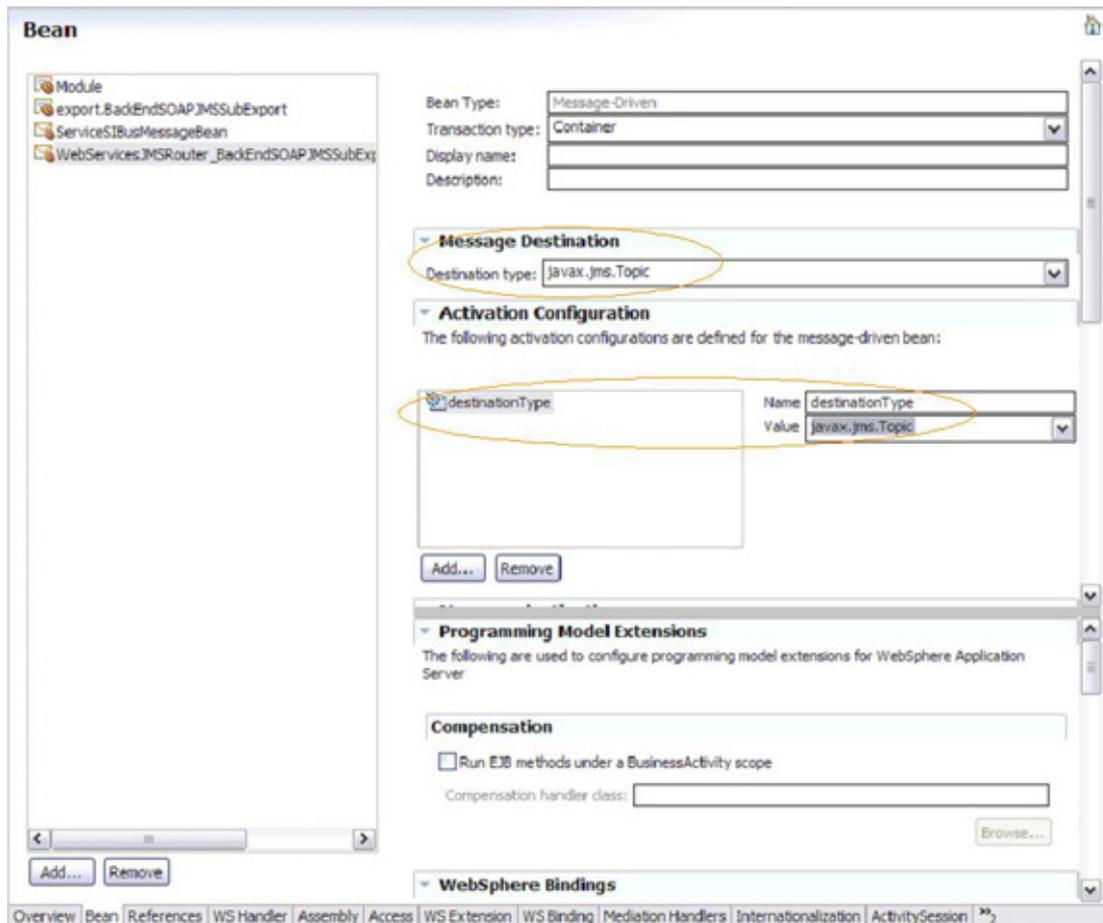
為使用發佈/訂閱傳訊模型。其中包括模組當中部署描述子檔案 `ejb-jar.xml` 的部分 J2EE 構件，這些構件可以在 J2EE 專案中找到。依預設：

- 訊息目的地類型為 `javax.jms.Queue` ；
- 啟動規格目的地類型為 `javax.jms.Queue` 。

您必須變更它們，如圖 15 所示：

1. 使用 EJB 專案中的部署描述子編輯器開啓 `ejb-jar.xml` 檔，請務必儲存此檔案，以免需要再次建置專案，因為使用 **JMS 上的 SOAP** 專案建置 Web 服務將會產生預設的 JMS 構件，這些構件和訊息佇列相關。如果您是在變更部署描述子之後建置專案，這些構件會被預設值改寫。
2. 圖 15 顯示您需要為 `WebServicesJMSRouter_BackEndSOAPJMSSubExport_GreetingServiceJmsPort` 變更訊息目的地類型與啟動配置中的 `destinationType` 。

圖 15：訂閱者 Web 服務應用程式範例



1. 您還需要修改 BackEndSOAPJMSSubExport\_GreetingServiceJmsPort.wsdl 檔案，藉此變更 Web 服務 JMS 上的 SOAP 連結位址內容。若要如此，請選擇 **MyLibrary > Web Service Ports** 並且將

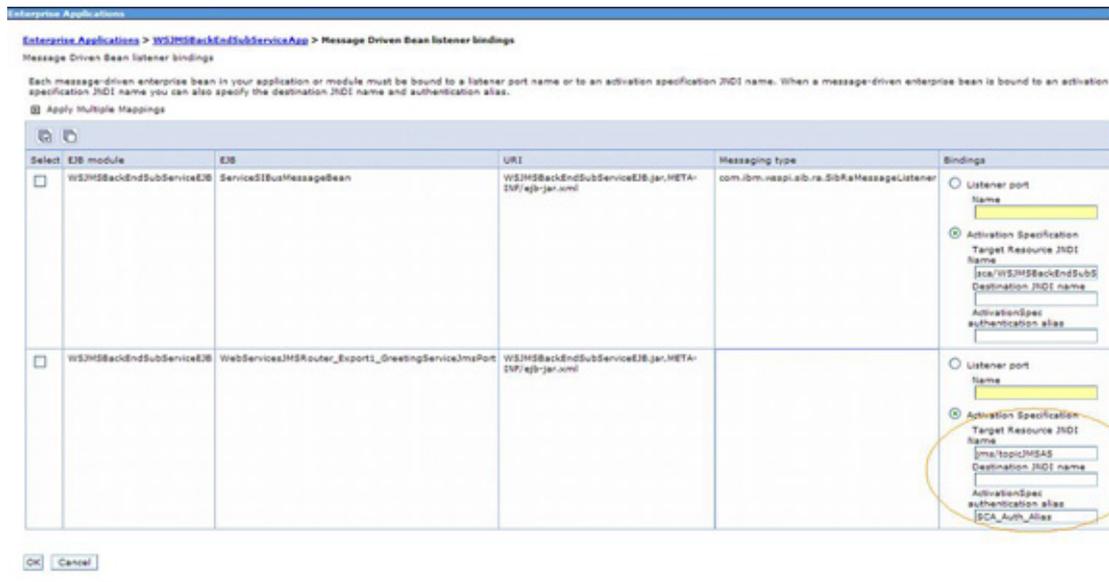
```
jms:/queue?destination=jms/BackendSOAPJMSSubExport&connectionFactory=jms/BackendSOAPJMSSubExportQCF&targetService=BackendSOAPJMSSubExport_GreetingServiceJmsPort
```

變更爲

```
jms:/topic?destination=jms/BackendSOAPJMSSubExport&connectionFactory=jms/BackendSOAPJMSSubExportQCF&targetService=BackendSOAPJMSSubExport_GreetingServiceJmsPort
```

2. 按一下 **File > Export > Business Integration > Integration Module**，將專案匯出成 EAR 檔。然後按一下 **Next** 以選取 WSJMSBackEndSubService，並且提供用來儲存 WSJMSBackEndSubService.ear 檔的資料夾。
3. 從管理主控台安裝 WSJMSBackEndSubService.ear 檔，並且修改**訊息驅動 Bean 接聽器連結**以指向所定義的啟動規格 JNDI，如圖 16 所示（請參閱圖 16 的放大版）。

圖 16：修改匯出 MDB 啟動規格



## 呼叫使用 JMS 上的 SOAP 的 Web 服務以發佈主題

內含使用 Web 服務 JMS 上的 SOAP 連結匯入的 Web 服務模組可以發佈 Greeting 主題的所屬訊息；訊息將供 WSJMSBackEndSubService 消費。此模組的範例稱為 WSJMSPublisher。

1. 建立名為 `WSJMSPublisher` 的新模組。
2. 將 `BackEndSOAPJMSSubExport_GreetingServiceJmsPort.wsdl` 檔從 `MyLibrary` 拖曳至組合圖，以建立使用 `Web 服務 JMS 上的 SOAP 連結匯入`。選取 **Import with Web Service Binding** 選項以建立匯入。
3. 變更匯入元件的連結內容，從

```
jms:/topic?destination=jms/BackEndSOAPJMSSubExport&connectionFactory=jms/BackEndSOAPJMSSubExportQCF&targetService=BackEndSOAPJMSSubExport_GreetingServiceJmsPort
```

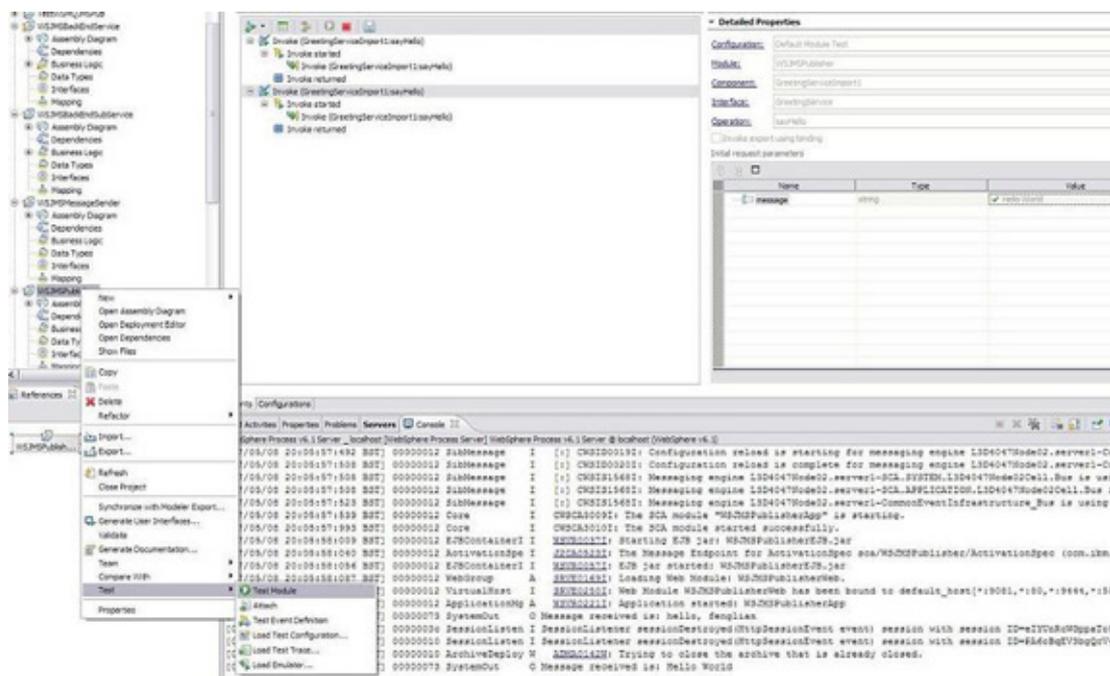
變更爲

```
jms:/topic?destination=jms/greetingTopic&connectionFactory=jms/greetingTopicCF&targetService=BackEndSOAPJMSSubExport_GreetingServiceJmsPort
```

4. 儲存專案並且將它從 `WebSphere Integration Developer` 部署至 `WebSphere Process Server` 伺服器。

現在您可以使用 `WebSphere Integration Developer` 中的測試用戶端來測試 `WSJMSPublisher`；圖 17 顯示其結果（請參閱圖 17 的放大版）。

圖 17：發佈/訂閱模型中的測試結果



總結

本文示範如何在 WebSphere Integration Developer 中建置及呼叫使用 JMS 上的 SOAP 的 Web 服務，以及如何在 WebSphere Process Server 或 WebSphere Enterprise Service Bus 中安裝及執行它們。Web 服務 JMS 上的 SOAP 匯出連結是用來建立 Web 服務提供者，而 Web 服務 JMS 上的 SOAP 匯入連結則是用來建立 Web 服務用戶端應用程式。

本文中的範例說明在 WebSphere Process Server 與 WebSphere Enterprise Service Bus 中，Web 服務功能所支援的點對點與發佈/訂閱傳訊模型。WebSphere Integration Developer 工具提供 GUI，可讓您輕鬆且有效率地建置 Web 服務應用程式。

## 下載

說明	名稱	大小	下載方法
適用點對點傳訊的應用程式	WSJMSPointToPoint.zip	80KB	<a href="#">HTTP</a>
適用發佈/訂閱傳訊的應用程式	WSJMSPubSub.zip	49KB	<a href="#">HTTP</a>

[下載方法相關資訊](#)

## 資源

### 學習

- 閱讀文章「[建置使用 JMS 上的 SOAP 與 WebSphere Studio 的 JMS Web 服務](#)」(developerWorks, 2004 年 2 月)。
- 從文章「[WebSphere Process Server 的 Web 服務用戶端程式設計](#)」獲得進一步資訊 (developerWorks, 2006 年 8 月)。
- 查看 IBM WebSphere 開發人員技術日誌「[在 Service Integration Bus 中部署發佈與訂閱應用程式](#)」(developerWorks, 2005 年 8 月)。
- IBM developerWorks 中的 [SOA 與 Web 服務區](#)提供數百篇文章，以及有關如何開發 Web 應用程式的初級、中級與進階教學指導。
- 在 [IBM SOA 沙盤推演](#)中進行操兵！透過運用 IBM SOA 進入點的現成實務體驗，提升您的 SOA 技能。
- [IBM SOA 網站](#)提供有關 SOA 與 IBM 如何協助您達成目標的概觀。
- 透過 [developerWorks 技術活動與網路廣播](#)隨時獲得最新消息。
- 瀏覽 [Safari 書店](#)中和這些與其他技術主題相關的書籍。
- 查看[隨需應變 Web 服務快速示範](#)。

## 取得產品與技術

- 使用透過下載或 DVD 取得的 [IBM 試用軟體](#)，為您的下個開發專案進行創新。

## 討論

- [參加討論區](#)。
- 加入 [developerWorks 部落格](#)，藉此參與 developerWorks 社群。

## 關於作者

Fenglian Xu 是英國 IBM Hursley Lab WebSphere Enterprise Service Bus 開發團隊的軟體開發人員，她的專長領域包括 WebSphere Enterprise Service Bus 和 WebSphere Application Server 中的 JMS 與 Web Services over JMS Bindings。她曾在多家 IT 公司任職，性質從中介軟體到應用程式都有，工作資歷長達十年，主要專精於 Java、J2EE 與 SOA。她在 1998 年取得南安普敦大學的電腦科學博士學位。

Simon Holdsworth 則在 IBM Hursley Lab 擔任軟體工程師長達 15 年之久，其間曾接手各式各樣的專案，包括大型主機組譯器程式設計、C、C++ 與最近的 Java。Simon 目前負責 IBM WebSphere Enterprise Service Bus 和 WebSphere Process Server 產品的整體軟體架構與高階連通性設計，他同時也是 OASIS SCA Bindings 技術委員會的主席。從 2001 年起，他就開始設計及實作各種支援 Web 服務的系統。在從事尖端技術開發期間，他曾經獲得 15 項專利，堪稱是 IBM 的傑出發明家。