

## IBM WebSphere 開發人員技術日誌：使用 WebSphere ESB 建置企業服務匯流排，第三部分

新增 Web 服務與提升內容

層級：中等

[Rachel Reinitz \(rreinitz@us.ibm.com\)](mailto:rreinitz@us.ibm.com)，IBM 資深顧問 IT 專家

[Andre Tost \(atost@us.ibm.com\)](mailto:atost@us.ibm.com)，IBM 資深技術人員

2006 年 1 月 24 日

先前的系列文章討論了主要的 [WebSphere ESB 功能](#) 以及 [跨 JMS 的傳訊交換](#)，現在我們以新增 Web 服務為例，以擴充 IBM® WebSphere® ESB 所運用的技術集。此外，也要特別介紹在 WebSphere ESB 與 WebSphere Integration Developer 6.0.2 版推出的新功能。

摘自 [IBM WebSphere 開發人員技術日誌](#)。

### 前言

我們的情境示例包括透過 JMS 傳送單向訊息給後端應用程式，指出包裹已遞送給客戶。我們將此對映至服務的「單向」作業（意指作業已提出要求但無回應），並且給予 JMS 連結。本文還會探討透過「要求-回應」作業（意指每項要求已獲回應），在服務要求者與提供者之間進行同步互動。服務將使用 SOAP/HTTP 連結做為通訊協定。如同前一個案例，我們透過「企業服務匯流排」來遞送每一個訊息，由 WebSphere ESB 6.0.2 版實作，分開服務者要求者與提供者。服務者要求者與提供者不會發現訊息透過 ESB 流入，同時也完全不知道 ESB 的遞送與記錄。

您還會瞭解如何配置執行時期，遞送訊息給不同的端點位址，以及如何變更已記載訊息的明細，而這兩項操作都無須重新部署中介。您可以直接在 WebSphere ESB 的管理主控台進行配置，這是 6.0.2 版產品的新功能，該版本已在[第二部分](#)發佈以後推出。從本文開始，此系列以後的文章都會使用此版本的產品。

### 情境示例不斷演進

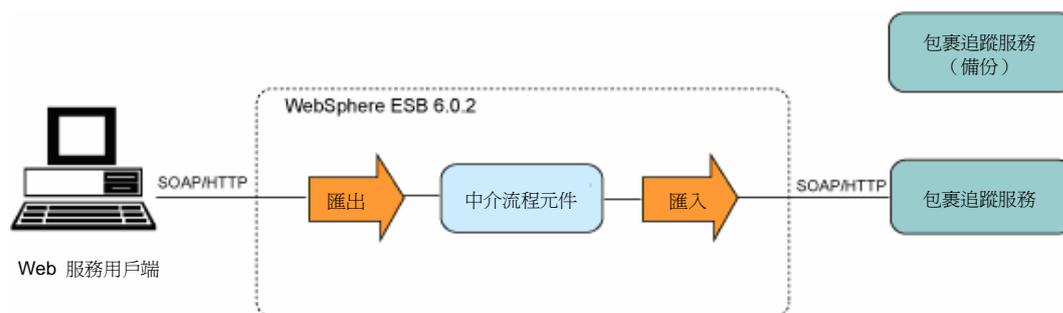
情境示例的商業背景是以 Posts-R-Us，讓客戶與員工擷取包裹遞送的狀態。為

了盡可能讓各種用戶端平台存取此功能，提供的 Web 服務會使用包裹的追蹤編號做為輸入參數，並且將內含包裹狀態的 XML 文件傳回，當中更附有估計（或實際）的遞送日期。

新服務的每一個要求都會透過 ESB 遞送，並由 WebSphere ESB 中介模組展現。中介模組會記載每個要求與回應訊息，並且遞送要求至兩個實際服務邏輯建置之一。為提高整體系統的彈性，服務建置有兩個，其中一個為備份。請注意，在示例提及的不是含應用程式伺服器的叢集環境（在發生故障時，應用程式伺服器成為自動備份系統），而是位於不同端點位址的第二個服務實例。以下示範如何配置中介模組的匯入連結，以便使用 WebSphere ESB 管理主控台時，在執行時期切換至後備服務。

按照慣例，中介流程元件會分別透過匯出與匯入，與其用戶端及實際的服務提供者進行通訊。圖 1 顯示系統架構的樣貌。

圖 1：情景示例的架構



## 服務提供者

示例需要實作一個服務提供者，將 trackingNumber 的變數做為輸入，並且會傳回 packageStatus 結構，內含包裹的資訊。

1. 本文稍後將引導您執行測試，在進行其中一個測試前，請務必確定已適當配置測試伺服器。在「伺服器」視圖的 **WebSphere ESB Server 6.0** 版按兩下，確定已選取 **Run server with resources on Server**（圖 2）。在啟動伺服器，以及在伺服器上部署任何本文所述之 EAR 保存應用程式之前，必須執行前述動作。

本部分與稍後部分不會逐步說明，而是請您匯入預先定義的 EAR 檔案，您可以在下載部分找到該檔。在建置整個情景示例時，將強調說明各應用程式的重要部分。

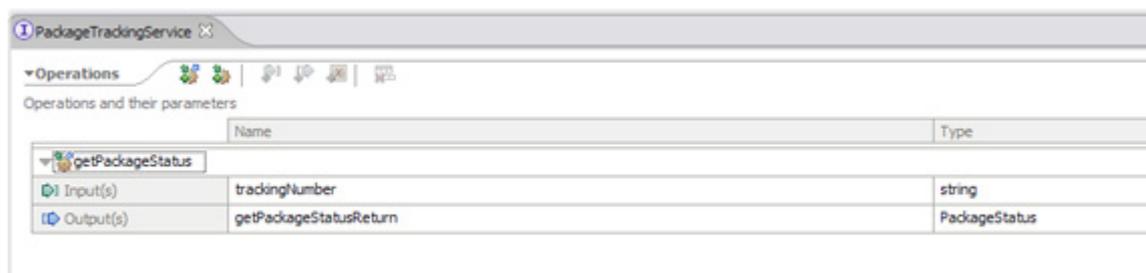
圖 2：測試伺服器配置

The image shows the 'Server Overview' configuration page in WebSphere. It is divided into several sections:

- General:** Server name: WebSphere ESB Server v6.0, Host name: localhost, Runtime: WebSphere ESB Server v6.0.
- Server:** WebSphere profile name: esb, Update server status interval (in milliseconds): 5000.
- Server connection type and admin port:** RMI (Better performance) is selected with ORB bootstrap port: 2811. SOAP (More firewall compatible) is also an option with SOAP connector port: 8882.
- Checkboxes:** Enable hot method replace in debug mode (checked), Enable universal test client (checked), Optimize server for testing and developing (unchecked), Terminate server on workbench shutdown (unchecked).
- Publishing:** Run server with resources within the workspace (unchecked), **Run server with resources on Server (checked and circled in red)**, Minimize application files copied to the server (unchecked), Enable automatic publishing (unchecked). Publishing interval (in minutes): 1.
- Security** and **Network Deployment** sections are visible but not expanded.

2. 可以在 PackageSatusServiceEAR.ear 檔案中尋找示例服務提供者，該檔案包含於本文隨附的下載資料中。請將此檔案匯入至 WebSphere Integration Developer 6.0.2 版，然後為目標伺服器選取 **WebSphere ESB Server 6.0** 版。現在執行服務的 WebSphere Application Server 實例，與安裝在 WebSphere ESB 的實例相同；但在真實環境中，服務很可能在不同的伺服器上分別執行。
3. 在 J2EE 視景可以看到名為 PackageStatusService 的 Web 專案。請注意，專案包含名為 PackageTrackingService.wsdl 的 WSDL 檔，其中含有服務定義。如果在工具的「介面編輯器 (Interface Editor)」開啓這個 WSDL 檔，可以看到已定義的要求-回應作業，名為 getPackageStatus，如圖 3 所示。

圖 3：PackageTrackingService 介面



	Name	Type
Input(s)	trackingNumber	string
Output(s)	getPackageStatusReturn	PackageStatus

4. 接下來到實際的實作類別，名為 `PackageTrackingServiceSoapBindingImpl.java`，位於 `postsrus.service` 套件中。如果在工具的 Java 編輯器中開啓此類別，會看到在示例中處理的兩個追蹤編號「123」與「456」。其他編號將導致異常狀況。稍後測試情境示例時，會用到此資訊。
5. 另一個屬意類別位於相同套件的 `PackageStatus.java`，其中包含呼叫服務作業以後，傳回的資料結構。資料結構由四個項目組成：

```
package postsrus.service;

public class PackageStatus {
    private java.util.Calendar actualDeliveryDate;
    private java.lang.String location;
    private java.util.Calendar projectedDeliveryDate;
    private java.lang.String status;

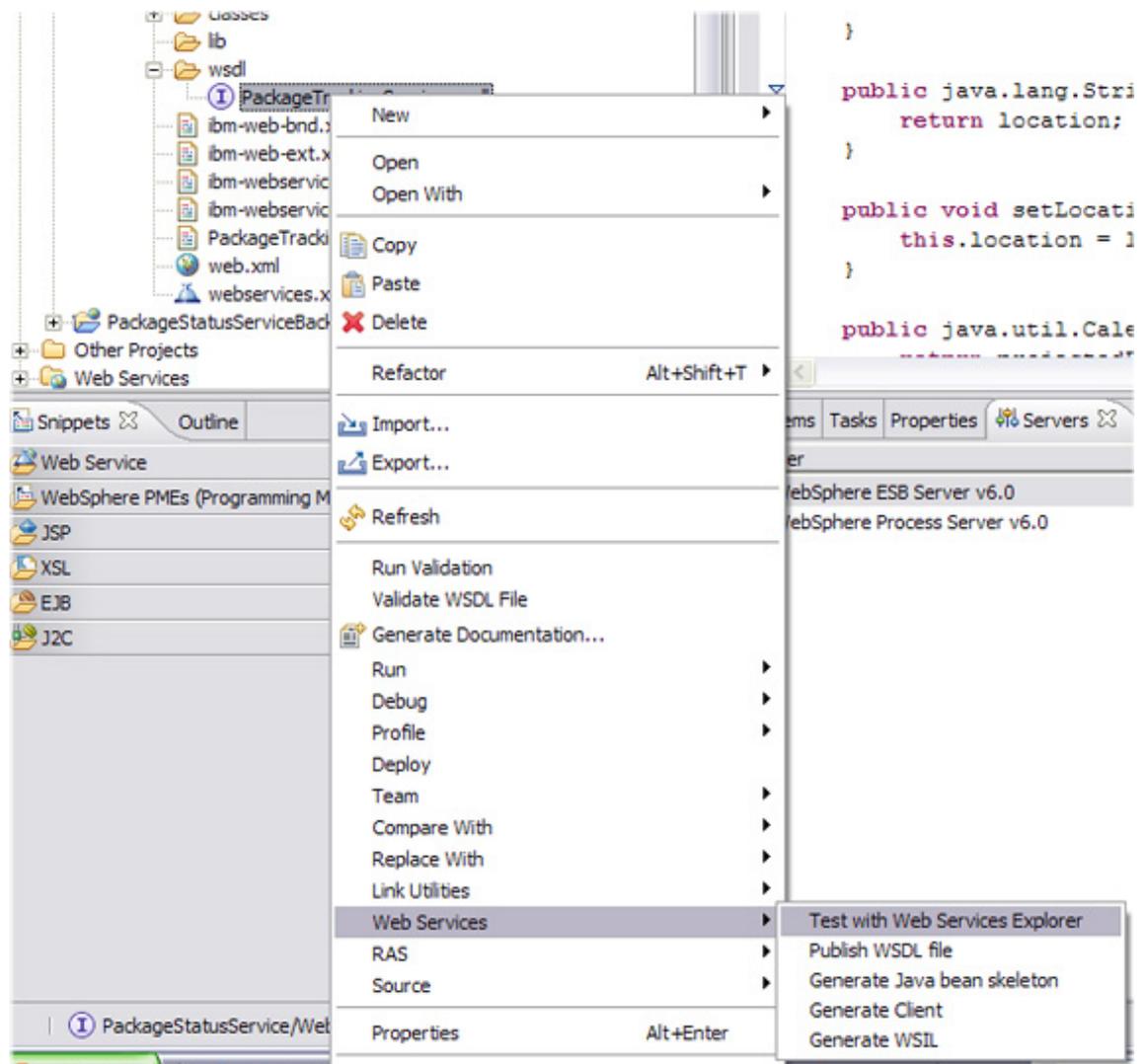
    public PackageStatus() {
    }
    ...
}
```

請注意 `java.util.Calendar` 類型如何呈現遞送時間與日期。後文會說明如何對映至 XML 綱目中 `<xsd:dateTime>` 類型的元素。

6. 您現在可以簡單測試此服務，確定服務已適當部署並能順利運作。若要進行測試，請啓動工具中的 **WebSphere ESB 6.0.2 測試伺服器**，然後使用 **Add and remove projects...** 功能表選項，在伺服器上部署 `PackageStatusServiceEAR` 應用程式。
7. 部署與啓動應用程式以後，要使用 **WebSphere Integration Developer Web**

服務測試工具。若要啓用 WebSphere Integration Developer 中的 Web 服務功能，請展開「Preferences」下的 **Workbench => Capabilities**，並且查看 **Web Service Developer**。請用滑鼠右鍵按一下 **PackageStatusService.wsdl**，然後選取 **Web Services => Test with Web Service Explorer**，如圖 4 所示。

圖 4：選取 Web Services Explorer 選項

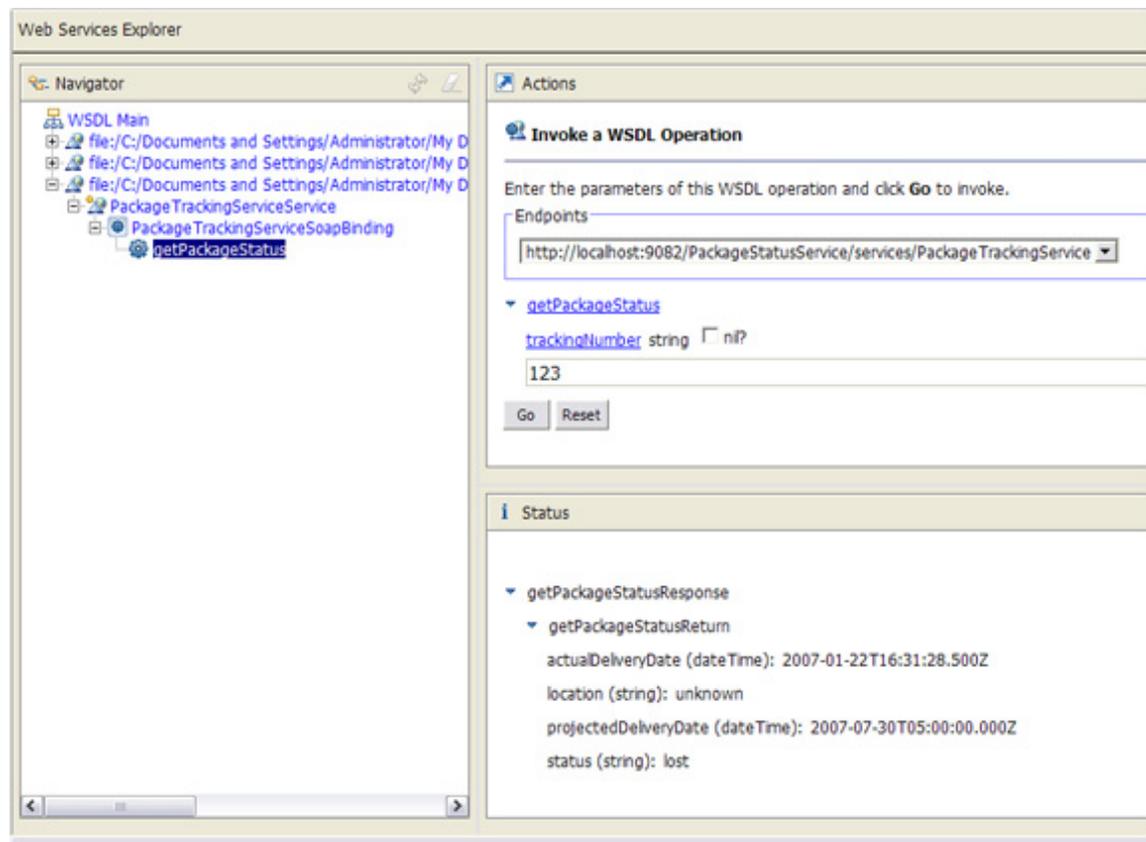


這樣可以開啓 Web Service Explorer 視圖，無須實際產生測試用戶端應用程式，也能測試 Web 服務。

8. 在 **getPackageStatus** 操作鏈結上按一下，輸入 123 做為追蹤編號，然後按一下「Go」。在測試伺服器指派連接埠的方式各有不同，如果遇到連線錯誤，請選取 Web Services Explorer 中的 **PackageTrackingServiceSoapBinding**，然後新增另一個端點。可以將連

埠設定為 9081，或查看測試伺服器設定，以決定正確的連接埠。在伺服器主控台中應該可以看到列印成品，指出已經呼叫服務，並傳回 PackageStatus 物件（物件 5）。

圖 5：Web Services Explorer



9. 請繼續作業，並將 PackageStatusServiceBackupEAR.ear 檔匯入 WebSphere Integration Developer，然後選取 **WebSphere ESB** 做為目標伺服器。若有需要，請變更端點連接埠。此專案包含相同的 Java 類別集與其他構件，同時也有已定義的 Web 服務，當中包含 getPackageStatus 作業。換言之，這是與剛才一樣的服務；唯一的差異是部署的端點位址不同，而且指定已呼叫「後備服務」的 System.out 陳述式也有所不同。
10. 現在可以將此專案新增至 WebSphere ESB 伺服器，並且使用 Web Services Explorer 測試專案。請務必用滑鼠右鍵按一下 PackageStatusBackupService 專案的 WSDL 檔。執行測試之後，伺服器主控台中應該可以看到以下這行的資料，指出已經呼叫後備服務：

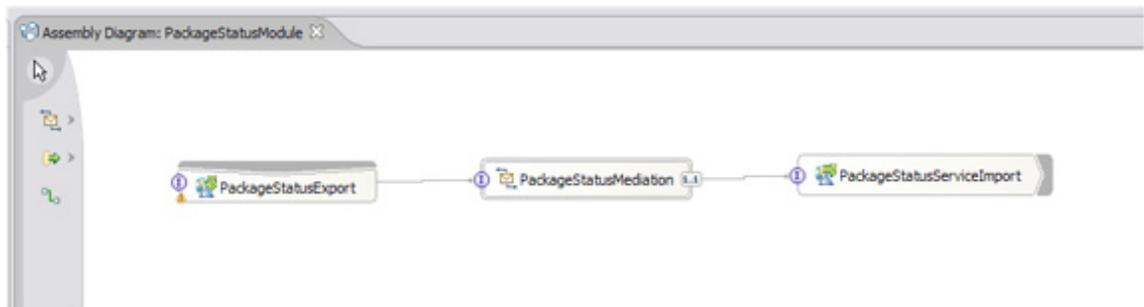
```
[2007/1/22 10:52:59:422 CST] 00000048 SystemOut      O Package status  
request in backup service for tracking number 456
```

## 中介模組

以下討論中介模組，在剛安裝的服務提供者與使用服務的任何用戶端之間，中介模組是其中的媒介。

11. 將模組儲存為 **Project Interchange** 檔，名為 **PackageStatusModule.zip**。匯入檔案之後，請切換至 **WebSphere Integration Developer** 中的 **Business Integration** 視景。在 **PackageStatusModule** 模組中，應該可以在「**Interfaces**」下看到 **PackageStatusService** 檔。檔案與服務提供者所用的 **WSDL** 檔相同；換言之，中介模組會展現相同介面做為原始服務，因此無須在 **ESB** 進行轉換。請注意，在「**資料類型**」的 **PackageStatus** 定義，其中含有服務傳回的欄位。
12. 請在「**配件 (Assembly)**」編輯器中，為模組開啓配件圖。此配件並無任何特別之處，其中只顯示中介流程元件、匯出與匯入（圖 6）。

圖 6：配件圖



13. 選取 **PackageStatusServiceImport** 並查看相關的內容視圖。在「**Binding**」標籤，可以看到箭頭指向早前部署服務實作的端點位址。請注意，埠號設定為 **9082**。如果測試伺服器使用埠號 **9081**，可以在內容視圖中更新。同樣的，如果選取 **PackageStatusExport**，可以看到包含新端點位址的 **Web** 服務連結。稍後在 **Web Services Explorer** 中測試完整解決方案時，將會用到這個位址。
14. 在 **PackageStatusMediation** 上按兩下，開啓「**中介流程 (Mediation Flow)**」編輯器。由於已預先建置中介流程元件，因此這裡沒有特別的狀況：每個要求與回應訊息都透過 **MessageLogger** 中介基本元素遞送，因此會記載於預設資料庫中。

圖 7：中介流程元件



在 **MessageLogger** 基本元素的內容中，可以定義 **XPath** 陳述式，指示實際要記載哪一部分的訊息。在「內容 (**Properties**)」視圖的「明細」(**Details**) 標籤中可以有關設定。「根 (**Root**)」的內容預設為 `"/body"`，意指記載整個訊息。在示例中，我們變更此內容為 `"/body/getTrackingStatus/trackingNumber"`，意指僅記載實際編號。

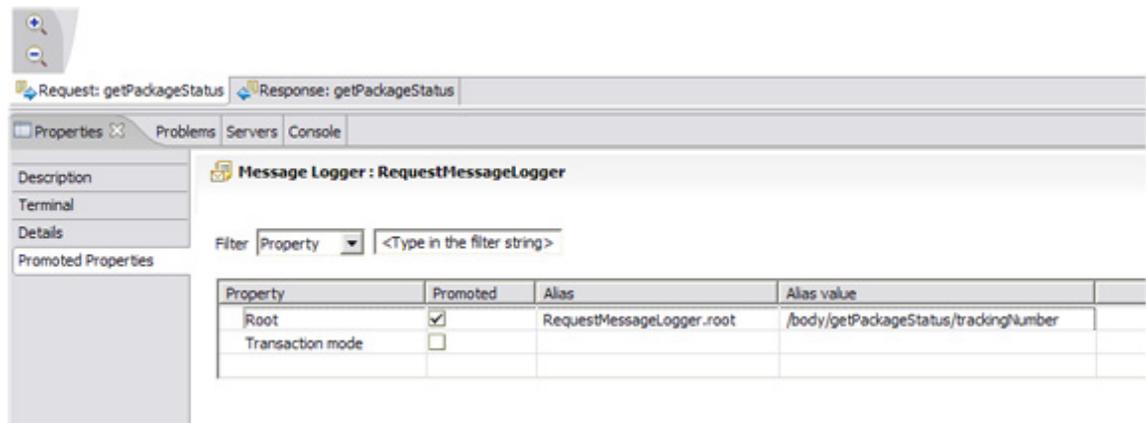
## 提升內容

這裡我們使用 **WebSphere ESB 6.0.2** 版的新功能，稱為**提升內容**。在較早的版本，若要變更內容如 **MessageLogger** 基本元素的「根 (**Root**)」內容，唯一的方法是將模組載入 **WebSphere Integration Developer**，然後在伺服器重新部署模組。在 **6.0.2** 版，如果內容有所「提升」，便可以在執行時期透過管理主控台變更。在「內容 (**Properties**)」視圖中已新增「提升內容 (**Promoted Properties**)」標籤，選取特定的中介基本元素，變更內容。每個中介基本元素都擁有特定的內容集，這些內容集都可以提升。

在情境示例中，您希望提升內容，以定義要記載哪一部分的訊息。有了前述的功能，可以在執行時期調整要記載的訊息量，無須重新部署模組。假設服務有效負載大量訊息，但為了節省儲存空間，您決定僅記載小部分的訊息，可是希望執行特定訊息的深入分析時，需要完整的訊息負載。現在只要透過管理主控台，便可以解決問題。

15. 請開啓中介流程內，**RequestMessageLogger** 基本元素的「內容 (**Properties**)」視圖，然後選取「提升內容 (**Promoted Properties**)」視圖 (圖 8)。

圖 8：提升內容



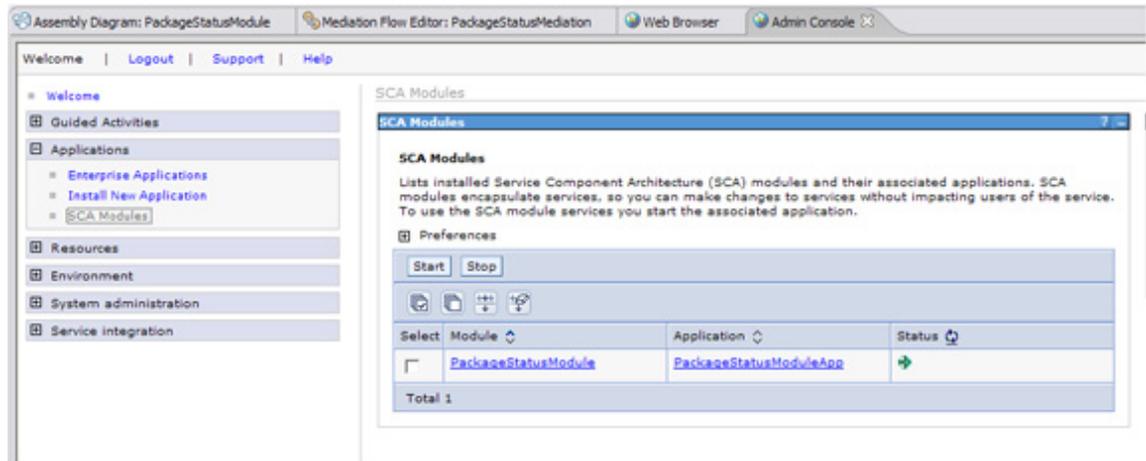
注意每個已提升的內容均有別名。若相同模組中具有多個名稱相同的內容，在管理主控台也可以區分內容。在此情況下，內容稱為 **RequestMessageLogger.root**。如果在回應佇列中查看 **message logger** 基本元素，會發現相同內容也在該處提升，但有不同的別名：**ResponseMessageLogger.root**。別名相同的「提升內容」將共享相同的值。

#### 完成解決方案

現在已準備就緒，可以部署中介模組並且執行測試。

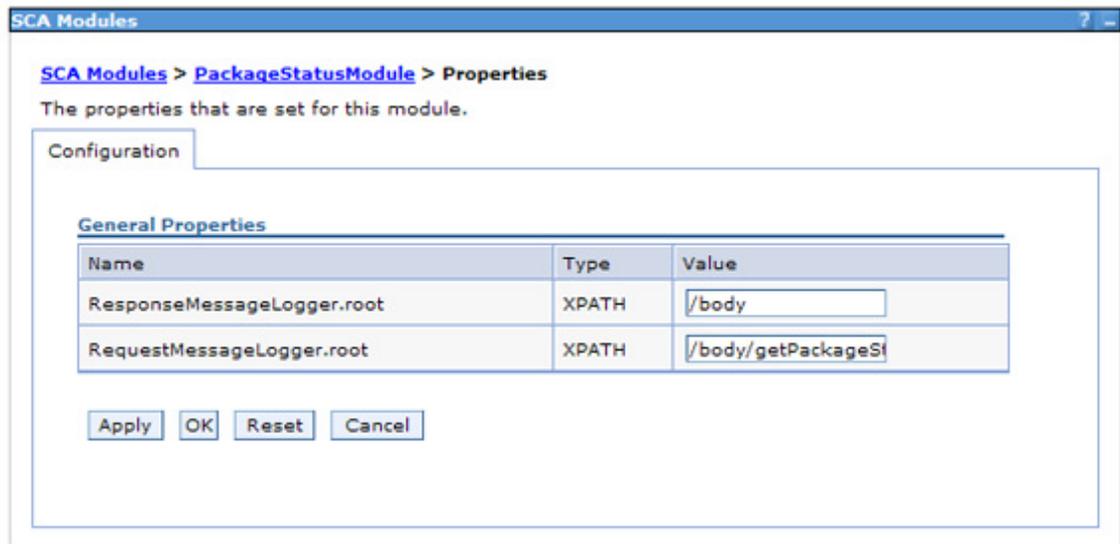
16. 請繼續進行操作，將 **PackageStatusModuleApp** 新增至您的測試伺服器。啓動應用程式以後，請切換至 **J2EE** 視景，在 **PackageStatusModule** 專案的「其他專案 (Other Projects)」下，尋找名為 **PackageStatusExport\_PackageTrackingServiceHttp\_Service.wsdl** 的檔案。這檔案是為匯出中介模組而建立 **Web** 服務時，所產生的部分結果。請用滑鼠右鍵按一下該檔案，然後選取 **Web services => Test with Web Services Explorer**。
17. 測試包含追蹤號碼 **123** 的 **ESB** 服務呼叫。結果應該與之前相同，但由於呼叫日誌程式基本元素的關係，這次會看到更多列印成品。該基本元素預設在 **Cloudscape®** 資料庫記載訊息。稍後將查看其中記載的項目。
18. 請先在「伺服器 (Servers)」視圖中，用滑鼠右鍵按一下測試伺服器，然後選取 **Run administrative console**，以開啓測試伺服器的主控台。在管理主控台，瀏覽 **SCA Modules** 頁面，如圖 9 所示。

圖 9：SCA Modules 頁面



- 按一下 **PackageStatusModule** 模組鏈結。在接著的頁面，按一下「其他內容 (Additional Properties)」的 **Module Properties**，然後在下個頁面，留意已顯示較早前提提升的兩個內容與內容值：

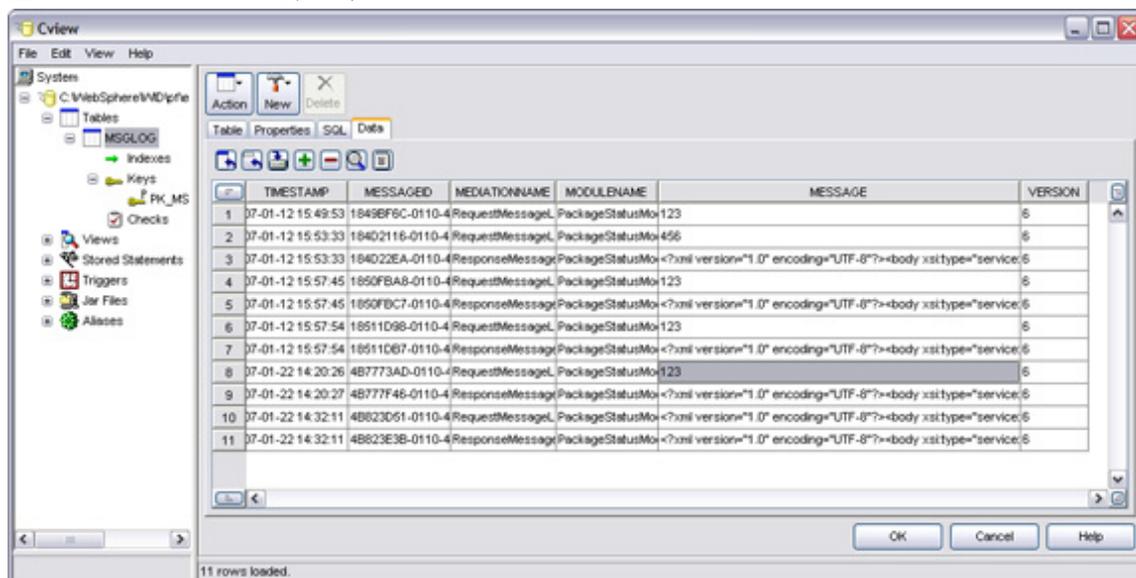
圖 10：模組內容



- 將 RequestMessageLogger.root 內容變更為 "/body"。按一下 **OK**，儲存變更。登出管理主控台並關閉視窗。
- 請在 Web Services Explorer 較早之前執行的測試。然而，請改用「456」做為追蹤編號，以此區別不同的記載訊息。
- 執行測試後停止伺服器，以解除日誌程式資料庫的鎖定，查看所記載的訊息。請使用 cview 公用程式，程式位於 [WebSphere Integration Developer install directory]\runtimes\bi\_v6\cloudscape\bin\embedded directory。開啓

公用程式，選取 **File => Open** 功能表選項，然後尋找 **EsbLogMedDB** 資料庫，資料庫位於 [WebSphere Integration Developer install directory]\pf\esb\databases directory。開啓 **MSGLOG** 標籤，並使用 **Data** 標籤查看內容。視窗應與圖 11 類似。

圖 11 : cview 公用程式



## 變更端點位址

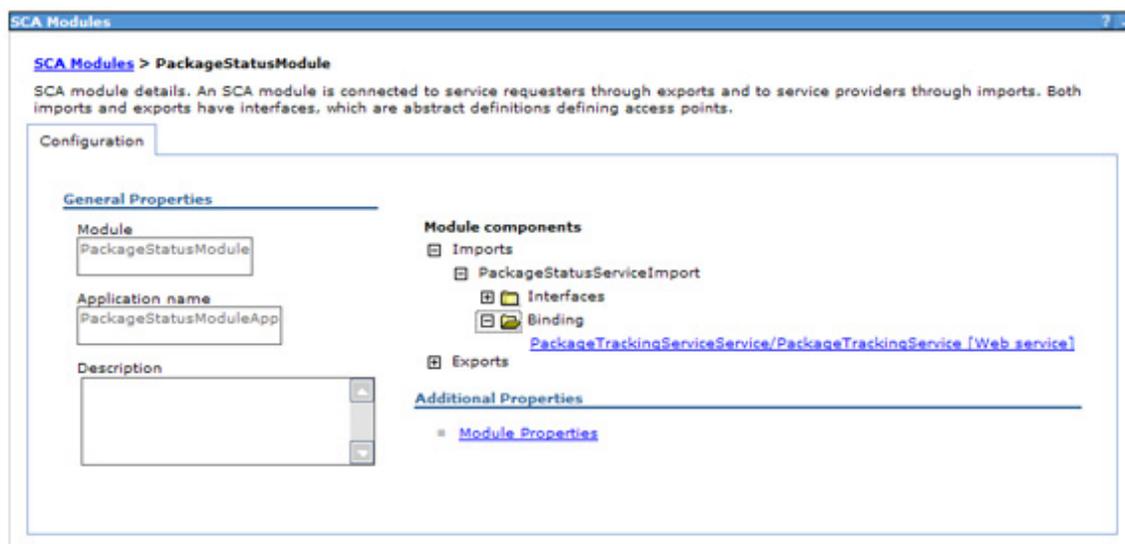
較早之前曾部署兩個實際的服務提供者實例，一個是主要服務 **PackageStatusService**，另一個是名為 **PackageStatusBackupService** 的額外服務。在許多情況下需要輕易切換呼叫服務的端點，從開發環境移至測試環境便是其中一個情況。那如何遞送訊息至備援服務，而無須重新部署，變更 **WebSphere Integration Developer** 的模組呢？方法有兩種，這裡說明其中一種。

如果使用 **Cloudscape** 做為資料庫，只需要停止伺服器檢視記載訊息。實際的正式作業環境會使用生產量的資料庫，如 **DB2®**，因此無須停止伺服器。

1. 可以在流程插入其他的基本元素（即 **MessageElementSetter** 基本元素），藉此動態遞送訊息至新的端點位址。這基本元素可以更新環境定義標題，讓訊息不根據匯入連結的配置，轉往另一個位址。基本元素也可以選擇結合更新與登錄查閱，如 **WebSphere Registry** 與 **Repository** (**WebSphere ESB** 與 **WebSphere Integration Developer 6.0.2** 版中提供另一個新基本元素)。有關此方法的詳細資訊超出本文介紹範圍，但可以在 **WebSphere Enterprise Service Bus 6.0.2** 版的「動態遞送加強功能」找到更多資訊與詳細的情境示例。

- SCA module 的端點位置可以在伺服器管理主控台變更。若曾停止伺服器，請重新啟動伺服器，並執行管理主控台，然後如前述般開啓 SCA Modules 頁面。選取 **PackageStatusModule** 模組，展開「模組 (Module)」元件下的 **Imports** (圖 12)。

圖 12：模組元件



按一下 **Binding** 鏈結開啓頁面，置換此匯入所使用的端點 URL。將端點 URL 變更為後備服務的端點 URL，即 `http://localhost:9082/PackageStatusServiceBackup/services/PackageTrackingService`。如前所述，我們假設測試伺服器使用 9082。如果並非使用 9082，則 URL 需要更新，才能使用正確的連接埠。

最後，按一下 **OK** 儲存變更。然後再次執行測試；在伺服器主控台應該看到後備服務的列印成品。

## 總結

本文首先介紹透過 Web 服務 (SOAP/HTTP)，讓服務與 WebSphere ESB 進行互動。在不同的端點位址部署了兩個相同的 Web 服務，兩者都傳回包裹的狀態資訊。在服務要求者與服務提供者之間，已建立的中介流程元件十分簡單，只用來記載資料庫的訊息。

其次說明在 WebSphere ESB 6.0.2 版中，提升內容的支援可於執行時期，在管理主控台變更中介基本元素的內容。此示例變更了記載訊息的明細層級。

最後是如何使用管理主控台，為包含 Web 服務連結的匯入變更其端點位址，將要求從某個服務提供者重新遞送至另一個服務提供者，無須重新載入模組至開發工具，或重新部署模組。

## 本系列其他部分

- 第一部分：使用 WebSphere ESB，或 WebSphere ESB 和 SIBus 比較簡介
- 第二部分：JMS 傳訊示例

## 下載

說明	名稱	大小	下載方法
程式碼示例	part3-downloads.zip	52KB	FTP

下載方法的相關資訊

## 資源

- [An introduction to the IBM Enterprise Service Bus](#)
- [WebSphere Enterprise Service Bus product page](#)
- [Getting started with WebSphere Enterprise Service Bus and WebSphere Integration Developer](#)
- [Developing custom mediations for WebSphere Enterprise Service Bus](#)
- [Building a powerful, reliable SOA with JMS and WebSphere ESB](#)
- [WebSphere Enterprise Service Bus V6.0.2](#)
- [Tutorial: Invoking a Web service with a JMS client](#)
- [Service Component Architecture](#)
- [Redbook: Enabling SOA Using WebSphere Messaging](#)

## 關於作者

**Rachel Reinitz** 是 IBM Software Services for WebSphere 的資深顧問 IT 專家，專長為 Web 服務。Rachel 為客戶和 ISV 提供的諮詢，主要是有關服務導向架構及 Web 服務如何能達成商業與技術上的目標。她開發了「IBM 進階 Web 服務訓練」課程，也經常出席會議發表演說。Rachel 也是「IBM 知識大學 (IBM Academy)」的成員，具有豐富的 eXtreme 程式設計課程教學經驗，並有四年的 XP 實務經歷。她現在居住於美國加州灣區，興趣是登山、參加社交活動和國外旅行。

**Andre Tost** 於 IBM「軟體事業處」的「整合解決方案」部門，擔任資深技術人員，協助 IBM 的顧客建置「服務導向架構」(SOA)，他的專長是 Web 服務技術。在此之前，他在 IBM 軟體開發中有長達十年的時間服務於合作夥伴啓用、開發和架構等各種相關業務，現於「WebSphere 業務開發處」服務。他來自德國，目前居住和就職於美國明尼蘇達州羅契斯特市。他休閒時喜歡與家人相處，有時間便踢足球、觀賞足球比賽。