



# 建置使用 Web 服務的豐富型 Java 用戶端



---

## 目錄

### 建置使用 Web 服務的豐富型 Java 用戶端 1

簡介：建置使用 Web 服務的豐富型 Java 用戶端	1
單元 1：在視覺化編輯器中設計用戶端 GUI	3
第 1.1 課：設定 Java 專案	3
第 1.2 課：新增和佈置員工表	4
第 1.3 課：執行視覺化類別	8
單元 2：將視覺化元件連結至 Web 服務	10
第 2.1 課：安裝和部署 Web 服務	10

第 2.2 課：將員工表連結至 Web 服務資料來源	12
第 2.3 課：將明細欄位連結至表格選項	17
第 2.4 課：將「更新」按鈕連結至動作連結程式	21
第 2.5 課：啓用「刪除」按鈕和確認對話框	23
第 2.6 課：設定動作和連結來新增員工	25
第 2.7 課：設計「取消」按鈕行爲	30
第 2.8 課：在員工表設定過濾器	30
摘要：建置使用 Web 服務的豐富型 Java 用戶端	32



---

## 建置使用 Web 服務的豐富型 Java 用戶端

這個指導教學教您如何利用 Java 視覺化編輯器來建置連接到 Web 服務的豐富 Java 用戶端。您在指導教學中建置的用戶端稱為 My Company Directory。

My Company Directory 是一個 Java 應用程式，用來維護公司的員工目錄。應用程式會連接 Web 服務範例，提供建立、擷取、更新及刪除員工記錄的方法。

用戶端是在 Java 視覺化編輯器中，以視覺化的方式，及 Swing 元件來建置的。Java 視覺化編輯器提供了一組 Helper 類別（資料來源、資料物件和連結程式），用來連接及使用 Web 服務。Web 服務部署在 IBM WebSphere Application Server 6.0 版安裝架構的本端環境中，工具可協助您根據 Web 服務說明語言 (WSDL) 檔來產生用戶端的 Java Proxy。

請參閱完成產品

### 學習目標

在本指導教學中，您將學到下列課程：

- 如何利用 Java 視覺化編輯器來設計和編排使用者介面
- 如何將介面元素連結至資料物件和 Web 服務

### 所需時間

2 小時 15 分鐘

#### 相關資訊



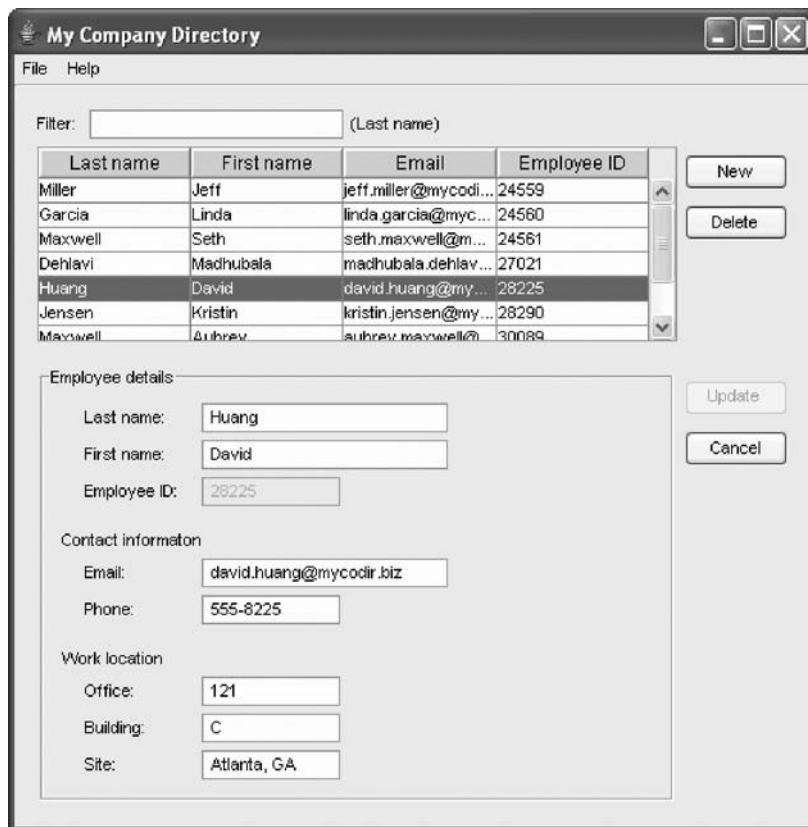
檢視 PDF 版本

指導教學：Hello World Java

---

## 簡介：建置使用 Web 服務的豐富型 Java 用戶端

用戶端的圖形使用者介面 (GUI) 大部分是利用 Swing 元件以視覺化的方式事先建置好的。在第一個單元中，您將利用 Java 視覺化編輯器來完成基本 GUI 元件的佈置。在第二個單元中，您要將 GUI 元件連結至 Web 服務資料來源、服務和資料來源所傳回的物件。當在 Java 視覺化編輯器中建置應用程式時，您將使用資料來源、資料物件和連結程式，它們是由 Java 視覺化編輯器產生且供應用程式使用的 Helper 類別實例。



檢視完成產品的圖片：

## 學習目標

在本指導教學中，您將學到下列課程：

- 如何利用 Java 視覺化編輯器來設計和編排使用者介面
- 如何將介面元素連結至資料物件和 Web 服務

## 所需時間

完成整個指導教學約需 2 小時又 30 分鐘。

## 系統需求

- WebSphere Application Server 6.1 版。此伺服器可能已隨著您的產品一起安裝，或者您也可以使用自己的獨立式安裝。這個指導教學的課程要求您將 Web 服務範例部署在本端執行的 WebSphere Application Server 中。

Web 服務範例可以在其他伺服器上執行，但這個指導教學只有在 WebSphere Application Server 6.0 和 6.1 版上測試過。

## 必備條件

您必須熟悉下列概念：

- 基本 Java 開發
- 基本 Web 服務原則
- 基本工作台技術，如使用專案和導覽視景和視圖

---

## 單元 1：在視覺化編輯器中設計用戶端 GUI

本單元教您如何利用「Java 視覺化編輯器」將視覺化元件新增至應用程式，再以視覺化的方式來佈置元件並設定排列限制。本單元的最後一課教您如何執行 Java 檔，以檢視檔案變成實際應用程式時的外觀。

**記住：**開始本單元之前，您必須具備指導教學簡介所描述的必備知識。

### 學習目標

完成本單元的課程之後，您將瞭解概念且知道如何執行下列動作：

- 在 Java 介面中新增和佈置 JTable
- 執行視覺化類別來測試工作

### 所需時間

完成本單元約需 15 分鐘。

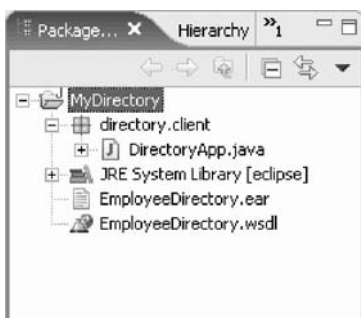
## 第 1.1 課：設定 Java 專案

在這一課，您會將專案匯入工作區來設定 MyDirectory 專案。專案包含單一 Java 類別及後續會用到的其他檔案。

因為這個指導教學的重點在於將視覺化元件連結至 Web 服務，所以 "My Company Directory" 應用程式的大部分 Java GUI 都已設計完成。

MyDirectory 專案是您將在這個指導教學中使用的主要 Java 專案。它包含您的 DirectoryApp.java 檔，也就是包含您在建置的主要 Java 應用程式的 Java 檔。這個指導教學包含多個 MyDirectory 專案版本來協助您：每個單元的開頭各有一個版本，整個完成的專案也有一個完成版本。

1. 匯入 MyDirectory 專案。
2. 在 Java 視景的「套件瀏覽器」中，確定 MyDirectory 專案如下圖所示：



### 課程回顧

在這一課，您已匯入 MyDirectory 專案範例做為這個指導教學的起點。

MyDirectory 專案包含下列資源：

- DirectoryApp.java：包含您在這個指導教學中開發的應用程式之 Java 檔。DirectoryApp.java 檔在名稱為 directory.client 的 Java 套件中。
- EmployeeDirectory.ear：包含 Web 服務範例的企業應用程式。在單元 2 中，您會將這個 Web 服務部署在 WebSphere Application Server 6.0 版的本端安裝架構中。

- EmployeeDirectory.wsdl：利用 Web 服務說明語言 (WSDL) 來說明您將部署的 Web 服務範例之 XML 檔。在「單元 2」，您將利用這個 WSDL 檔來產生 Java Proxy 供應應用程式使用。

## 第 1.2 課：新增和佈置員工表

在這一課，您將利用 Java 視覺化編輯器將 JScrollPane 和 JTable 新增至應用程式。在稍後的練習中，您將設計 JTable 從 Web 服務取得資料，Web 服務會傳回公司目錄中所有員工的記錄清單。

新增 JTable 之後，您將利用 Java 視覺化編輯器的設計視圖來自訂 JTable 的佈置，使它符合下列規格：

- 使 JTable 水平跨越三個資料格，垂直跨越兩個資料格。
- 新增左側插入項 15 像素。
- 將 JTable 重新命名為 employeesTable。

顯示說明文件

### 在 Java 視覺化編輯器中開啓 DirectoryApp.java 檔案

如果要在 Java 視覺化編輯器中開啓 DirectoryApp.java 檔，請執行下列動作：

- 在 Java 視景的「套件瀏覽器」視圖中，展開 MyDirectory 專案和 directory.client 套件。
- 用滑鼠右鍵按一下 DirectoryApp.java 檔案，選取**開啓工具** → **視覺化編輯器**。Java 視覺化編輯器會載入 Java 類別，並於圖形畫布區域上顯示設計。

提示：

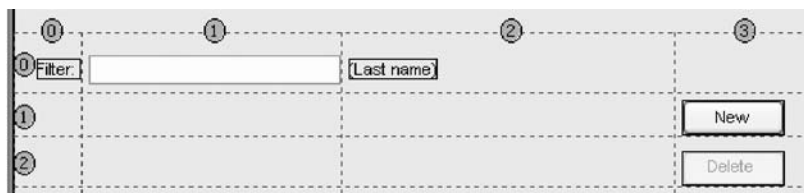
- 若要變更 Java 視覺化編輯器的外觀和操作方式，請跳至**視窗** → **喜好設定** → **Java** → **視覺化編輯器**，指定 Swing 外觀和操作方式。您下次開啓類別時，喜好設定便會生效。這個指導教學使用 Windows 外觀和操作方式。
- 若要讓「視覺化編輯器」成為所有 Java 檔的預設編輯器，請按一下**視窗** → **喜好設定**，再跳至**工作台** → **檔案關聯**頁面來定義您的喜好設定。

### 在 JScrollPane 上新增 JTable

DirectoryApp.java 的主視窗利用含 JPanel 的 JFrame 來作為它的主要內容窗格。我們的應用程式中的 JPanel 稱為 jContentPane。jContentPane 已設成使用稱為 GridBagLayout 的佈置管理程式類型。GridBagLayout 以視覺化元件所能佔用的資料格線為基礎，是一種功能強大的佈置設計。Java 視覺化編輯器會顯示格線邊框，方便您使用 GridBagLayout。它也會在您將新元件放在格線上時顯示放置標記，此外，它亦會顯示在 GridBagLayout 上調整大小或移動的元件之控點。

如果要將員工表 (javax.swing.JTable) 新增至 DirectoryApp.java 使用者介面中，請執行下列動作：

- 在設計視圖或 Java Bean 視圖中，用滑鼠右鍵按一下 jContentPane，選取**顯示格線**。紅色點虛線顯示格線邊框，藍色圓圈編號指示列和直欄號碼。例如，請注意**新建**按鈕佔用第 1 列（格線 y）和第 3 欄（格線 x）的資料格。

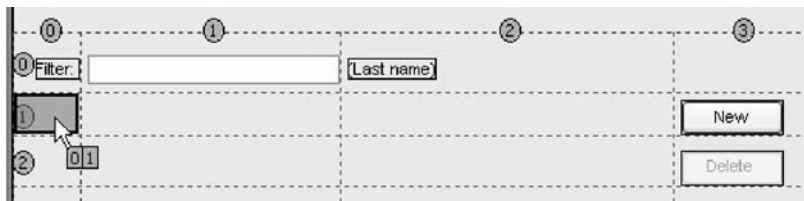


- 在 Java 視覺化編輯器選用區中，選取 **JScrollPane** 上的 **JTable**  Swing 元件，此元件分配在選用區的 **Swing** 元件抽屜下。



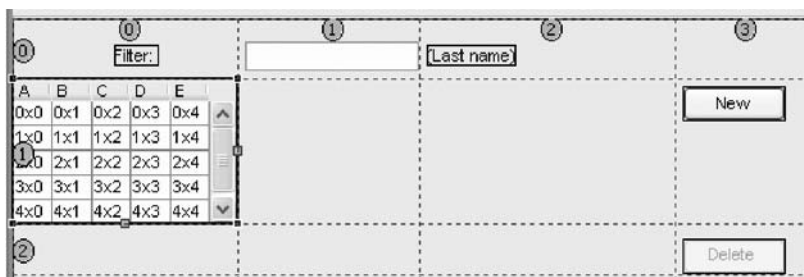
**提示:** 依預設，選用區會收合在設計區的右邊。您可以調整選用區的大小，也可以移動選用區。

3. 將滑鼠指標移至第 1 列、第 0 直欄格線中的資料格：



- 當您將滑鼠指標移到格線上，滑鼠指標會顯示兩個編號方塊來根據滑鼠指標位置說明格線中的 x 和 y 座標。
- 如果您將滑鼠指標直接停在格線邊框上，可能會建立新的列和直欄，現有的列和直欄會重新編碼。在這個情況下，滑鼠指標的黃色方塊、格線之間的黃線，以及黃色直欄和列標籤表示這個行為，且會指出放置所將造成的影響。

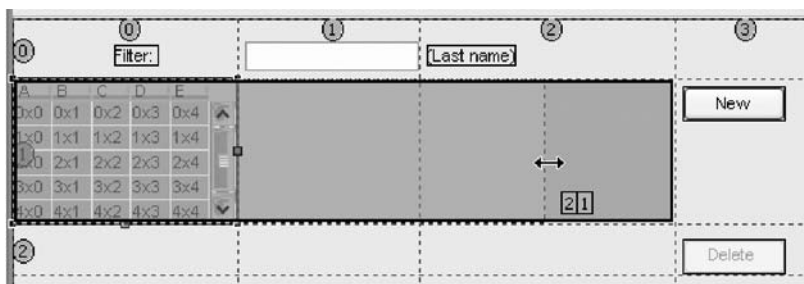
4. 按一下將 JScrollPane 和 JTable 放在第 0 欄和第 1 列的資料格中：



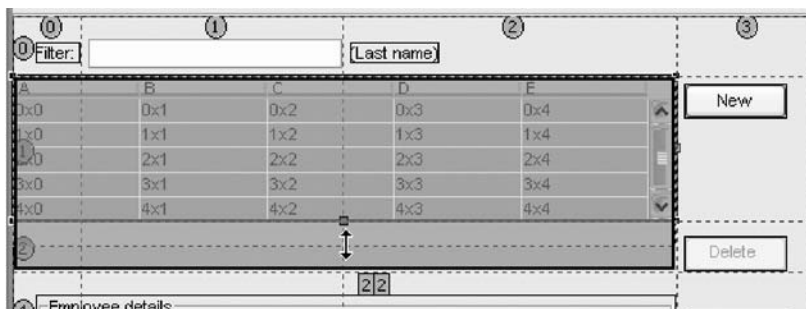
## 使 JScrollPane 和 JTable 跨越格線的多個直欄和列

現在，您要使 JScrollPane（及其子項 JTable）跨越三個直欄和兩列，以便有更好的調整間距和大小的行為。如果要使表格跨越直欄和列，請執行下列動作：

1. 在設計區域或 Java Beans 視圖中選取 JScrollPane（它應該仍在選取狀態，因為它是剛加入的）。請注意 JScrollPane 的右側和底端的小綠方塊。您將利用這些調整大小的控點來拖曳 JScrollPane，使它跨越多列和多直欄。
2. 在 JScrollPane 右側的綠色控點上，按住滑鼠左鍵。
3. 將滑鼠指標向右移，直到位置指出第 1 列、第 2 直欄為止。放開滑鼠按鈕時，深灰色陰影也會指出元件將佔用的資料格。



4. 放開滑鼠按鈕。這時 JScrollPane 會跨越三個直欄。
5. 重複類似的程序，拖曳 JScrollPane 的底端控點，直到 JScrollPane 跨入第 2 列為止：



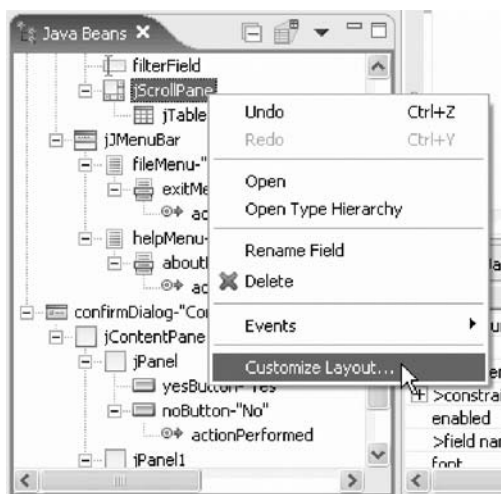
## 在 GridBag 內自訂 JScrollPane 的間距

GridBagLayout 管理程式的另一個特性是您可以指定各種限制來進一步自訂佈置。比方說，您可以指定下列限制：

- **錨點**：指定元件在資料格內的錨點方向，當使用者調整應用程式大小时，將影響元件如何移動。比方說，元件錨點可以定在左上、左中、中間或右下。
- **填入**：指定元件依水平、垂直方向或兩者來佔用一或多個資料格內的所有可用空間。
- **插入項**：指定元件在上、下、左、右的邊距，讓元件和格線邊緣之間留有空隙。

如果要自訂 JScrollPane 的錨點、填入項和插入項，請執行下列動作：

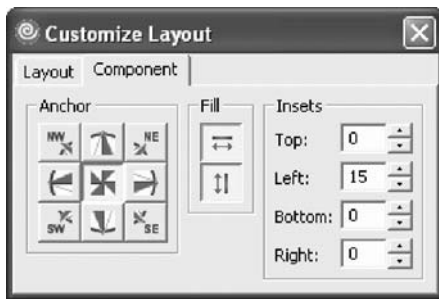
1. 在設計視圖或 Java Bean 視圖中，用滑鼠右鍵按一下 JScrollPane，選取**自訂佈置**。



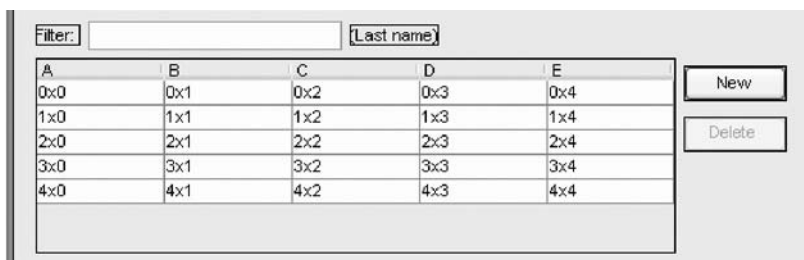
**提示：**在選取和變更不同元件的版面配置時，「自訂佈置」對話框可以開著。您可以隨時按一下功能表列中的「自訂佈置」按鈕來開啓「自訂佈置」對話框：



2. 在「自訂佈置」對話框的「元件」標籤中，確定已按下「置中固定」按鈕。
3. 確定已按下**水平填入**和**垂直填入**按鈕。
4. 增加 15 像素的插入項，讓 JScrollPane 的左邊距和應用程式的其他視覺化元件相似。



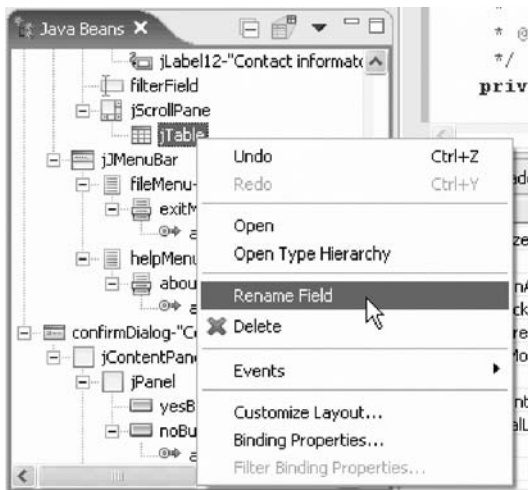
例如，表格現在對齊過濾器標籤。



將新的 **JTable** 重新命名為更有用的值，並設為選取單一列：

由於您稍後會使用表格，因此，將 **JTable** 實例及其 `getter` 方法重新命名將會很有用。如果要重新命名表格，請執行下列動作：

1. 在 Java Bean 視圖中，用滑鼠右鍵按一下 `jTable` 元件，從蹦現功能表中選取**重新命名欄位**。



2. 輸入 `employeesTable`，按一下**確定**。這時 **JTable** 名稱為 `employeesTable`，建立它的實例方法是 `getEmployeesTable`。
3. 將表格設為只能選取一列：
  - a. 在設計視圖中，選取 `employeesTable`。
  - b. 在「內容」視圖中，選取 **selectionMode** 內容並設為 `SINGLE_SELECTION`。

Property	Value
preferredSize	375,80
rowHeight	16
rowSelectionAllowed	true
selectionBackground	49,106,197
selectionForeground	Color:white
selectionMode	SINGLE_SELECTION
showGrid	
showHorizontalLines	true
showVerticalLines	true
toolTipText	
visible	true

c. 儲存 DirectoryApp.java 檔。

## 課程回顧

在這一課，您已瞭解如何利用視覺化編輯器，將表格新增至現有的使用者介面。您也學到如何自訂佈置、位置及間距。

## 第 1.3 課：執行視覺化類別

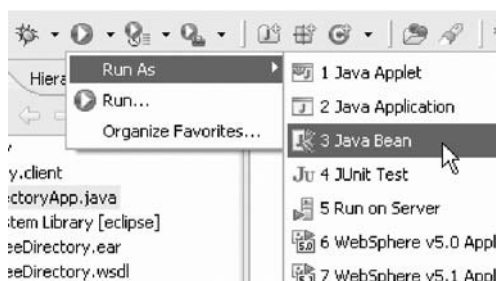
現在，您已準備好開始執行 Java 應用程式來預覽它的外觀。工作台和視覺化編輯器使應用程式非常容易快速執行，在開發過程中，您隨時可以重複這些步驟來測試實際的執行時期外觀及類別行為。

### 顯示說明文件

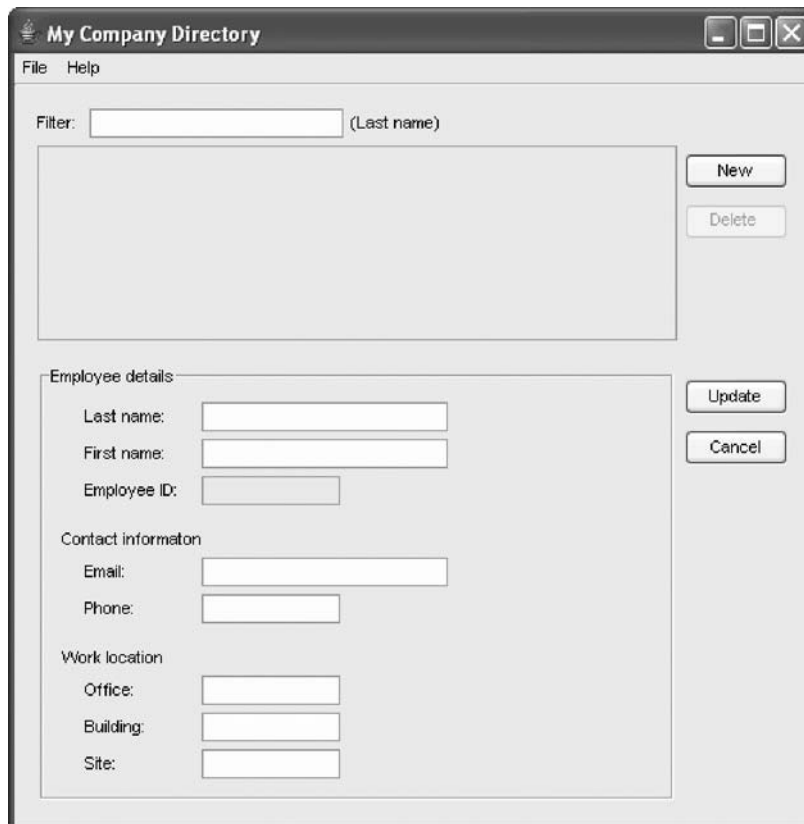
Java 視覺化編輯器提供一個 Java Bean 啟動程式，它能夠執行不含 main() 方法的類別。當它執行視覺化類別時，它會在個別虛擬機器 (VM) 中啟動應用程式。如果您將視覺化類別當作 Java 應用程式來執行，啟動程式會試圖執行類別中的 main() 方法。在這個指導教學中，您的應用程式包括一個呼叫和顯示 DirectoryApp JFrame 的 main() 方法，因此，您可以將它當作一個應用程式或 Java Bean 來執行。

如果要將 DirectoryApp.java 檔當作一個 JavaBean 來執行，請執行下列動作：

1. 確定已在 Java 視覺化編輯器中開啟您的 DirectoryApp.java 檔。
2. 從功能表列中，按一下執行 → 執行為 → **Java Bean**。



**提示：**應用程式以您在「視覺化編輯器」喜好設定中（視窗 → 喜好設定 → Java → 視覺化編輯器）已定義的 Swing 外觀和操作方式在桌面上開啟。另外，您可以也按一下執行 → 執行，定義特定啟動配置的外觀和操作方式，以啟動這個 Java Bean。如果您將這個應用程式當作應用程式而不是 Bean 來執行，它也會使用 Windows 的外觀和操作方式，因為它是定義在 main() 方法中。本指導教學中的畫面顯示 Windows 外觀和操作方式。



## 課程回顧

由於您只設計了介面，但還沒有設計任何資料連線或事件功能，因此，您的應用程式仍無法執行任何動作。不過，您可以查看使用者所見到的它的基本佈置和外觀。您可以試著按一下某些按鈕，但您將發現它們不會執行任何動作。不過，「檔案」功能表和「說明」功能表已完成實作。您可以試試它們，看它們會做什麼，您也可以視察 Java 程式碼來瞭解它們是如何用 `actionPerformed` 事件來實作的。

## 學習成果

這個單元向您介紹如何利用 Java 視覺化編輯器來設計豐富用戶端的介面。不過，除了設計用戶端的視覺化外觀，您還必須執行其他動作來實際使用用戶端更有用。您通常必須加入事件行為或其他邏輯，在這裡是將視覺化元素連結至某種資料來源。

在這個單元中，您學會了如何執行下列作業：

- 利用專案交換匯入來匯入 Java 專案
- 在 `JScrollPane` 中新增一個 `JTable` 到視覺化類別
- 利用 `GridBagLayout` 管理程式，以視覺化的方式將表格佈置在豐富用戶端中
- 執行應用程式來查看豐富 Java 用戶端的實際外觀

在接下來的「單元 2：將視覺化元件連結至 Web 服務」中，您會將一個簡易 `My Company Directory` 介面改造為功能強大的豐富型用戶端，可透過 Web 服務方法在公司目錄中建立、擷取、更新及刪除員工記錄。

---

## 單元 2：將視覺化元件連結至 Web 服務

本單元教您如何將 My Company Directory 的視覺化元素（按鈕、員工表、欄位及其他動作）連結至 Web 服務。這個 Web 服務提供在範例目錄中建立、擷取、更新和刪除員工的真實功能。

### 學習目標

完成本單元的課程之後，您將瞭解概念且知道如何執行下列動作：

- 將表格連結至資料 Web 服務資料來源
- 將欄位連結至物件
- 設計按鈕的動作

完成本單元約需 2 小時。

### 第 2.1 課：安裝和部署 Web 服務

在這個練習中，您會將企業應用程式範例 (EAR) 檔案安裝至 WebSphere Application Server 6.1 版並部署 EmployeeDirectory Web 服務。您的應用程式將利用這個 Web 服務來建立、讀取、更新及刪除員工記錄。

開始之前，必須先完成下列其中一個選項，以確保 MyDirectory 專案有正確的起點：

- 完成第 3 頁的『單元 1：在視覺化編輯器中設計用戶端 GUI』。

或

- 在「單元 2」開始時匯入 MyDirectory 專案

**提示：** 除非在匯入時指定不同的專案名稱，否則會改寫您的 MyDirectory 專案內容。

您的 MyDirectory Java 專案包含一個 EmployeeDirectory.ear 檔。您將利用 WebSphere 管理主控台來安裝 EAR 檔所包含的 EmployeeDirectory 企業應用程式。當您安裝應用程式時，您也會部署應用程式所包含的 Web 服務。完成的 My Company Directory 應用程式會使用這個部署的 Web 服務。

若要安裝 EmployeeDirectory 應用程式範例，並將 Web 服務部署在 WebSphere Application Server 6.1 版環境中，請執行下列動作：

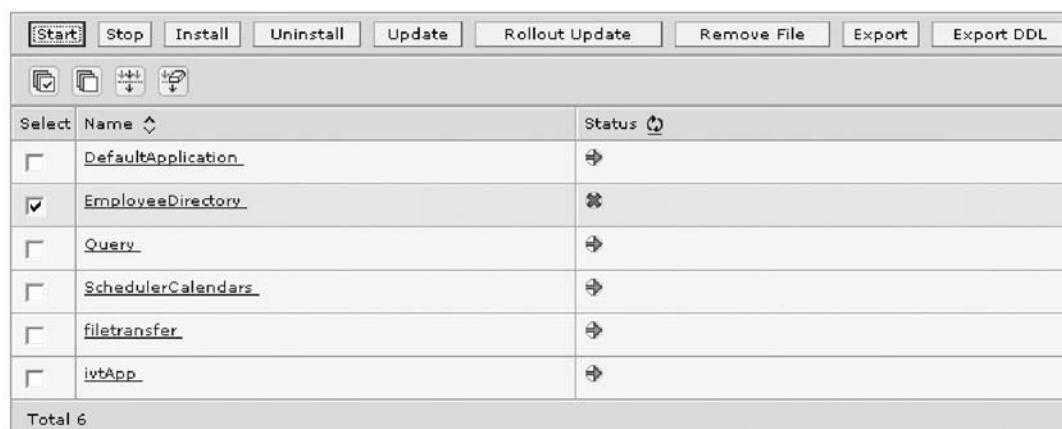
1. 從工作台啟動應用程式伺服器的實例。您可以利用多種不同的方式來啟動伺服器，但這些步驟說明如何從工作台執行它：
  - a. 開啓「伺服器」視圖。若要將「伺服器」視圖新增至 Java 視景，請按一下**視窗** → **顯示視圖** → 「其他」，選取「伺服器」 → **伺服器**。
  - b. 「伺服器」視圖列出已安裝和設定的伺服器。
  - c. 用滑鼠右鍵按一下伺服器，選取**啟動**。當「伺服器」視圖顯示伺服器狀態為**已啟動**時，或主控台指出已開啓 server1 伺服器以進行電子商業時，表示伺服器已順利啟動。這時您可以執行管理主控台。

**註：** 如果「伺服器」視圖中沒有伺服器實例，請建立一個新伺服器：

- a. 用滑鼠右鍵按一下「伺服器」視圖，選取**新建** → **伺服器**。
  - b. 使用「新建伺服器」精靈來新增 WebSphere Application Server 6.1 版。
2. 執行 WebSphere 管理主控台。同樣地，您可以利用其他方式來執行管理主控台，但這些指示說明如何從工作台執行它：
    - a. 在「伺服器」視圖中，用滑鼠右鍵按一下剛啟動的伺服器，選取**執行管理主控台**。這時會在瀏覽器視窗中開啓 WebSphere 管理主控台。



- b. 輸入使用者 ID，按一下**登入**。這時會開啓管理主控台的「歡迎使用」頁面。您輸入的使用者 ID 只用來追蹤特定使用者所進行的伺服器配置資料變更。
3. 利用管理主控台來安裝在 MyDirectory 專案中找到的 EmployeeDirectory.ear 企業應用程式。管理主控台會利用精靈方式來協助您安裝應用程式，您可以在其中按**下一步**，以在頁面之間移動，直到設好所有選項為止。若要安裝企業應用程式範例，內含本指導教學的 Web 服務，請執行下列動作：
  - a. 在「管理主控台」左邊，展開**應用程式功能表**選項，再按一下**安裝新的應用程式**。
  - b. 選取**本端檔案系統**，在**指定路徑欄位**中，輸入 MyDirectory 專案的 EmployeeDirectory.ear 檔案的完整路徑。提示：若要取得完整路徑，請在「套件瀏覽器」中用滑鼠右鍵按一下 EmployeeDirectory.ear 檔案，選取**內容**。「內容」頁面會列出檔案的位置，可以複製並貼到**指定路徑欄位**中。
  - c. 按**下一步**，一直到出現**選取安裝選項**頁面為止。
  - d. 選取**部署 Web 服務**。
  - e. 按**下一步**，一直到出現**摘要**頁面為止，然後按一下**完成**。
  - f. 當系統提示您套用您的本端配置變更時，請按一下**儲存在主要配置鏈結**。請檢視這些變更，再按一下**儲存按鈕**。
4. 利用管理主控台來啓動 EmployeeDirectory 應用程式：
  - a. 按一下**應用程式** → **企業應用程式**。EmployeeDirectory 應用程式列為伺服器中已安裝的應用程式，但狀態為「已停止」。

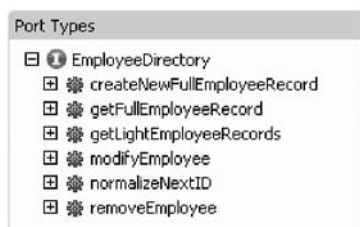


- b. 選取 EmployeeDirectory 旁的勾選框，按一下**啓動**。這時會出現一則訊息，指出已順利啓動 EmployeeDirectory 應用程式，「狀態」圖示會改成綠箭頭。

現在，EmployeeDirectory 應用程式正執行於 localhost 的 9080 埠，Web 服務是可存取的。完成這個指導教學之後，您可以回到管理主控台，停止 EmployeeDirectory 應用程式，再將它解除安裝。

如果您開啓在 MyDirectory 專案中找到的 EmployeeDirectory.wsdl 檔（依預設，它應該開在圖形 WSDL 編輯器中），您可以檢查您剛剛部署的 Web 服務。如果 WSDL 檔沒有開在 WSDL 編輯器中，就表示工作台可能沒有開啓 Web Service Developer 功能。您可以在喜好設定中指定工作台功能（**視窗** → **喜好設定** → **工作台** → **功能**）。

下列 WSDL 編輯器圖形顯示 EmployeeDirectory 服務中可用的作業：



您可以利用 WSDL 編輯器來檢查每項作業及對應的要求訊息和傳回訊息。這可協助您瞭解 Web 服務及其餘練習使用它的方式。

## 第 2.2 課：將員工表連結至 Web 服務資料來源

My Company Directory 應用程式可顯示目錄中所有現行員工的記錄清單。這些記錄顯示在含有姓氏、名字、電子郵件和員工 ID 等可排序直欄的 JTable (employeesTable) 中。爲了取得表格的記錄，必須將 employeesTable 連結至 Web 服務資料來源範例傳回的資料物件。

顯示說明文件

### 資料物件、資料來源和連結程式的概觀

爲了取得本端資料物件供 employeesTable 使用，您將利用視覺化編輯器，將資料來源新增至應用程式。資料來源會連接至 Web 服務 Proxy 範例，且會探索您的應用程式能夠使用的服務方法。之後，您將選擇從資料來源提供的 getLightEmployeeRecord 服務方法。最後，您會將應用程式中的 employeesTable 連結至列資料物件 (lightEmployeeRecordRows) 中傳回的欄位。

您可以利用 Java 視覺化編輯器的內建連結程式類別，又快又容易地建立所有這些資料來源和資料物件。當您將視覺化元件連結至資料 Factory 時，視覺化編輯器會提供一組產生在專案中的通用介面和類別。依預設，連結程式類別會產生在名稱爲 jve.generated 的套件中。視覺化編輯器會將連結程式類別提供成一項通用實作，供您進一步自訂和加強來配合您的應用程式需求。這個指導教學示範預設連結程式類別基本又簡單的用法所擁有的功能和彈性。

**重要：** 開始這個練習之前，最好先閱讀下列說明主題。這些主題有助於進一步瞭解 Java 視覺化編輯器所提供的資料物件、資料來源及連結程式背後的功能和邏輯：

- 資料連結程式概觀
- 連結程式 API 參照

在這個指導教學中，您將在應用程式中使用 Web 服務資料來源、若干類型的資料物件，以及若干類型的連結程式。將這些物件的實例新增至應用程式時，視覺化編輯器會將必要的類別新增至專案的 jve.generated 套件中，您可以在這裡延伸、取代或改寫資料連結邏輯。Java 視覺化編輯器提供了連結物件的視覺化支援，它會在設計視圖的開放式區域中，顯示應用程式所用的資料物件、資料來源和連結程式。視覺化編輯器會在視覺化元件和資料物件及資料來源之間繪製線條來顯示任何所選物件的現行連結。

下圖是視覺化元件、連結程式、資料物件和資料來源如何互動的簡單概觀。您在這個指導教學中建置的應用程式說明稍微複雜且較有創意的連結程式用法。這個圖不一定能完全如實反映您在建置的應用程式範例中的連結程式、資料物件和資料來源。



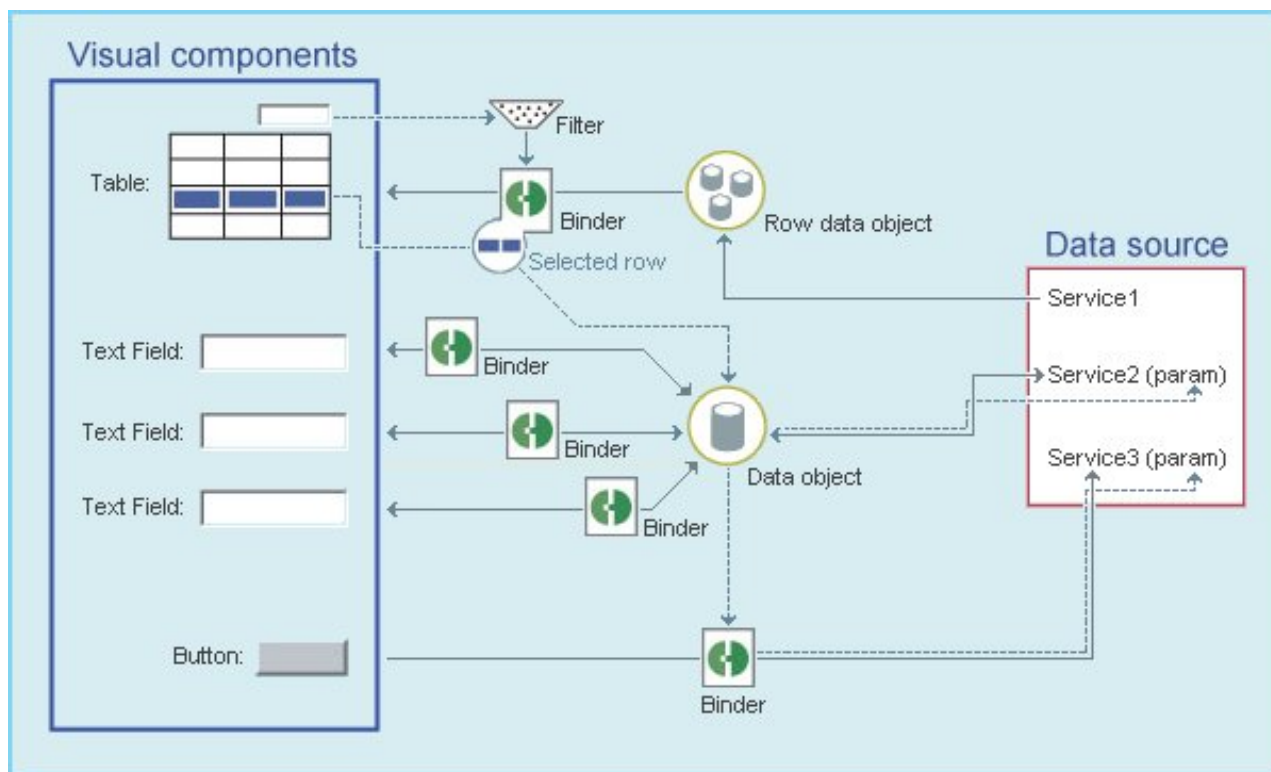


圖 1. 這個圖型示範視覺化元件、連結程式、資料物件及資料來源之間的關係。

在圖 1 中，每個視覺化元件都有它自己的連結程式會將它關聯於資料物件，如果是按鈕，則是關聯於資料來源。文字欄位的連結程式會將該欄位連結到資料物件的特定內容。這個圖解中的列資料物件和資料物件，都會直接呼叫資料來源的服務來取得其資料。文字欄位的資料物件會利用表格中所選的列中的鍵值來作為呼叫 Service2 的引數，這個呼叫會傳回完整的記錄，其中可能包括表格中所選列的詳細資訊。之後，這個完整記錄又用來作為按鈕呼叫 Service3 時，按鈕動作連結程式的引數，Service3 可能是更新欄位中所輸入之值的方法。如果需要資料物件、資料連結程式和資料來源的詳細說明，請遵循先前所提供的鏈結。

## 利用提供的 WSDL 檔在專案中產生 Web 服務 Java Proxy

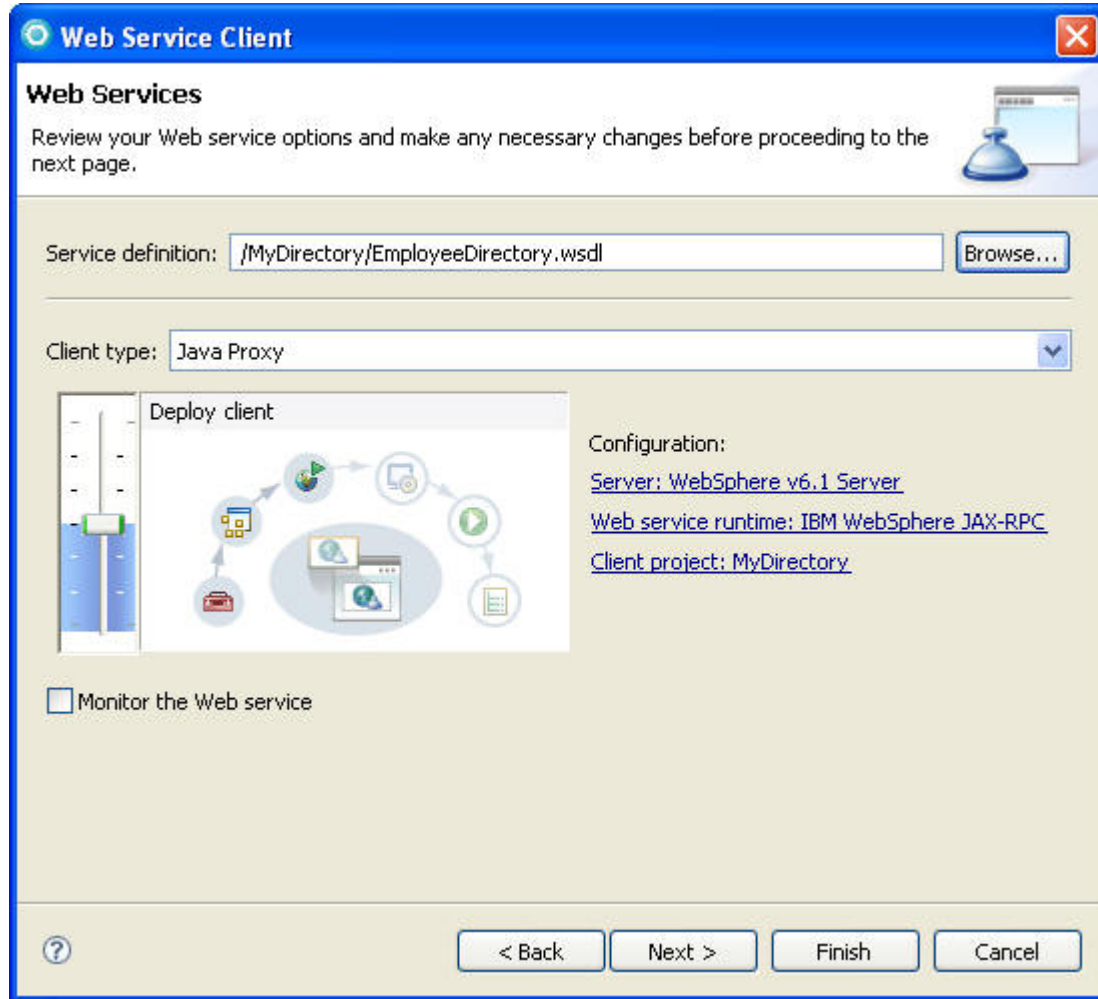
如果要使用在伺服器中執行的 Web 服務，您的 Java 應用程式需要一個與它互動的 Java Proxy 或用戶端。您可以利用「Web 服務用戶端」精靈和 WSDL 檔，在 Java 專案中產生一個 Java Proxy。您的 MyDirectory 專案包括將用來產生這個 Proxy 的 EmployeeDirectory.wsdl 檔。產生 Java Proxy 之後，您可以建立代表 Web 服務的資料來源，並開始連結視覺化元件。

**重要：** 這個練習中所用的 WSDL 檔假設您將 Web 服務部署在本端安裝的 WebSphere Application Server，且 localhost 使用預設埠 (http://localhost:9080)。如果您以不同的方式來部署 EAR 檔，您必須據此編輯 WSDL 檔之後，才能繼續作業。

如果要在專案中產生 Web 服務 Java Proxy，請執行下列動作：

1. 在主功能表中，按一下**檔案** → **新建** → **其他**，再選取 **Web 服務** → **Web 服務用戶端**精靈。如果沒有出現「Web 服務」種類，請選取**顯示所有精靈**。
2. 使用精靈來定義 Web 服務用戶端：
  - a. 在**服務定義**中，輸入您的 MyDirectory 專案中提供的 WSDL 檔案：/MyDirectory/EmployeeDirectory.wsdl
  - b. 在**用戶端類型**欄位中，選取 **Java Proxy**。

- c. 將滑塊列設為**部署用戶端**。
- d. 確定已根據您正在執行的伺服器來妥善設定伺服器和 Web 服務執行時期。這個指導教學已在使用 IBM WebSphere JAX-RPC 執行時期的 WebSphere 6.0 版和 WebSphere 6.1 版上測試過。
- e. 確定 Java Proxy 用戶端輸出至 MyDirectory 專案。



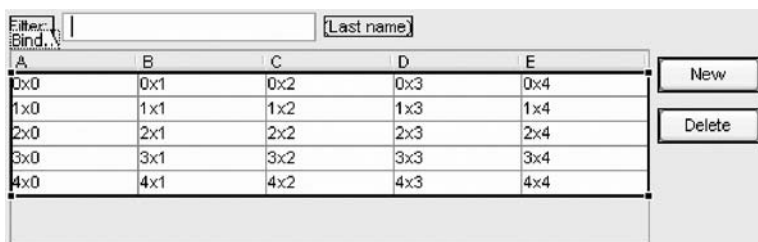
3. 按一下**完成**。「Web 服務用戶端」精靈會在專案的新套件 (directory.service) 中產生 Java Proxy。

### 將 **employeesTable** 連結至 Web 服務傳回的列資料物件。

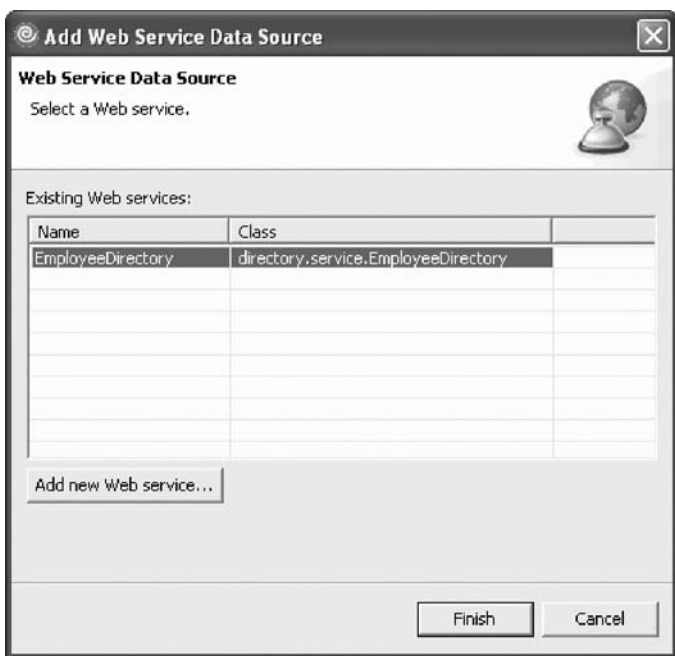
由於 **employeesTable** 是您在這個應用程式中連結的第一個視覺化元件，因此，您必須建立一個資料來源來指向剛新增到專案中的 Web 服務 Proxy 範例。當您在稍後的練習中連結其他視覺化元件時，您會重複使用這個資料來源。在這個步驟中，您將新增 Web 服務資料來源和 **lightEmployeeRecordRows** 資料物件。

如果要連結員工表，請執行下列動作：

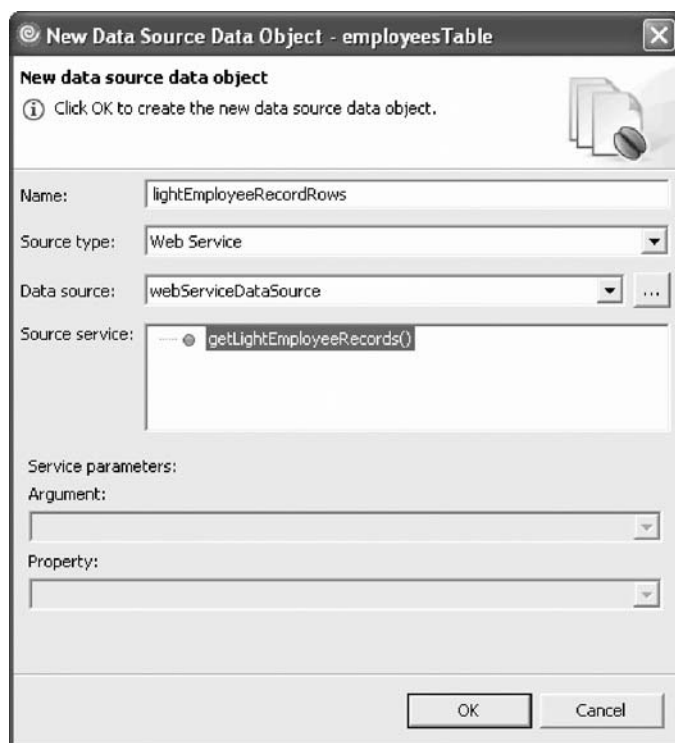
1. 在 Java Bean 視圖或設計視圖中，選取 **employeesTable**。（請確定您沒有選取它的 JScrollPane 母項）。在設計區域中，**employeesTable** 頂端會出現一個標示為**連結**的小標籤。



2. 按一下 employeesTable 的連結標籤。另外，您也可以用滑鼠右鍵按一下 employeesTable，選取連結內容。
3. 由於應用程式中沒有資料物件，因此，您必須新增一個新的資料物件。請按一下**新建資料來源資料物件**。
4. 在**來源類型**欄位中，選取 **Web 服務**。
5. 由於您還沒有新增任何 Web 服務資料來源到應用程式中，因此，您必須現在新增它。在**資料來源**欄位旁邊，請按一下 ... 按鈕來開啓「新增 Web 服務資料來源」對話框，將會在專案中尋找可用的 Web 服務用戶端或 Proxy。
6. 選取 EmployeeDirectory Web 服務，再按一下「完成」。這時會在 DirectoryApp.java 檔中新增新的資料來源。

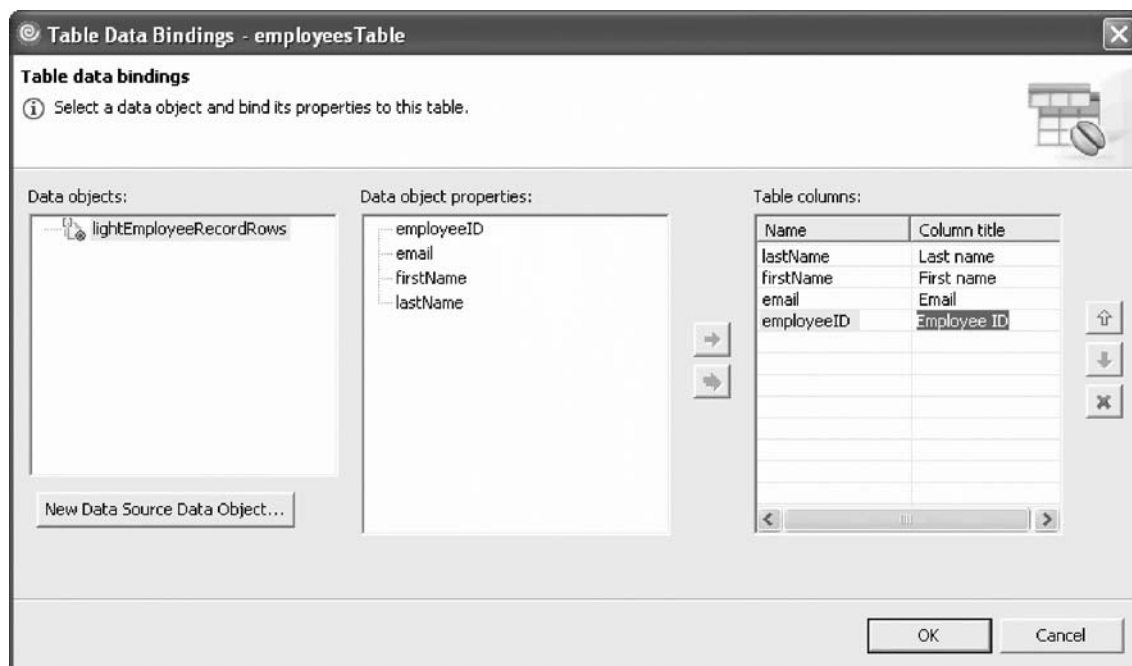


7. 在「新建資料來源資料物件」對話框，在「來源服務」欄位中選取 getLightEmployeeRecords()，接受新資料物件的預設名稱：lightEmployeeRecordRows。這個服務方法不需要任何參數。按一下「確定」。將建立新的資料物件，並顯示在設計視圖的開放式區域。



**提示:** 由於正進行表格連結，「新建資料來源資料物件」對話框只會顯示傳回列資料物件的服務。在這個情況下，getLightEmployeeRecords() 方法是唯一傳回物件陣列的可用服務。

8. 在「表格資料連結」對話框中，選取 lightEmployeeRecordRows 資料物件。
9. 現在，您必須選取 employeesTable 中所要顯示的 lightEmployeeRecordRows 資料物件的內容：



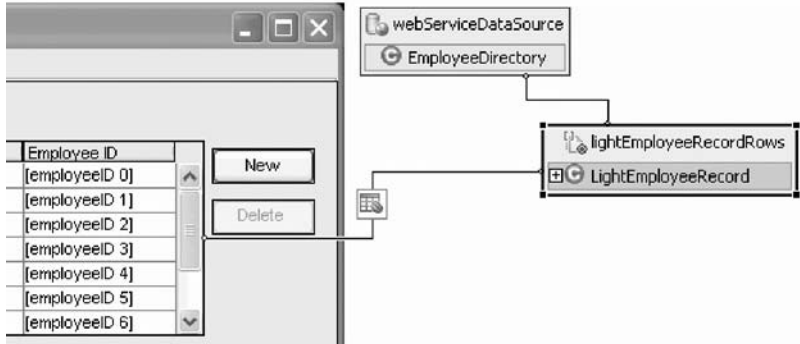
- a. 按一下雙箭頭 ➡ 按鈕，將所有物件內容新增至表格直欄清單。
- b. 利用上移和下移鍵來依照下列次序安排直欄，由上到下分別是：  
lastName、firstName、email、employeeID

- c. 重新命名直欄標題：姓氏、名字、電子郵件、員工 ID

**提示：**連結表格之後，您可以隨時返回連結內容來重新命名和重新排列直欄。

- d. 按一下**確定**。

這時會利用 `JRowTableBinder`，將 `employeesTable` 連結至 `lightEmployeeRecordRows` 資料物件。如果您在開放式區域中按一下 `lightEmployeeRecordRows` 資料物件，視覺化編輯器會畫一條從資料物件至表格的線。在這條線上，`JRowTableBinder` 以表格連結程式圖示來表示。另一條線表示資料物件以 `webServiceDataSource` 為資料來源。



## 課程回顧

請注意專案和應用程式的變更。在這一課，您已新增 Web 服務資料來源、列資料物件及連結程式（將 `employeesTable` 連結至列資料物件）。

查看專案中已建立新套件 (`jve.generated`) 來存放 Java 視覺化編輯器所產生的所有連結程式類別。也請注意存放 Web 服務 Java Proxy 的新套件 (`directory.service`)。請描述或概述這一課學到什麼。



現在，當您執行 `My Company Directory` 應用程式時，Web 服務會在員工表填入現有的員工記錄。

## 第 2.3 課：將明細欄位連結至表格選項

在上一個練習中，您將 `employeesTable` 連結至 Web 服務中的 `getLightEmployeeRecords()` 服務所傳回的 `lightEmployeeRecordRows` 資料物件。現在，您必須根據表格中所選取的員工來移入詳細資料欄位。

如果要取得每個所選員工的額外詳細資料，還會使用另一個資料物件。您將新增的 `selectedEmployeeRecord` 資料物件是 `getFullEmployeeRecord()` 服務所傳回的。這個服務以表格中之所選員工的 ID 為參數，且會提取員工的其他詳細資料，其中包括電話和工作位置。

您將表格連結至列資料物件時所用的 `JRowTableBinder` 會簡化這個步驟。`JRowTableBinder` 會將表格中的所選元素顯現成可用來作為 `getFullEmployeeRecord(java.lang.Integer)` 方法的參數的個別資料物件。之後，您就可以輕鬆地將每一個文字欄位連結至 `selectedEmployeeRecord` 資料物件中相對應的內容。

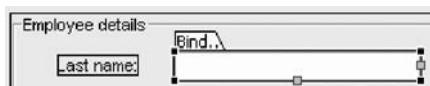
**進一步瞭解這個 Web 服務:** Web 服務包含兩個服務，可取得每一位員工的所有詳細資料。表格會列出所有員工，但只會顯示部分資料。之後，當選取某單一員工時，您可以只擷取這個所選員工的其餘員工資訊。如果在表格要求資料時，Web 服務傳送了每個員工的所有資料，Web 資料傳輸可能會太重，可能會降低應用程式的效能。

比方說，如果員工記錄包括照片或附件，當您只要取得完整員工清單時，您不會想擷取所有照片。因此，`getLightEmployeeRecord` 服務用來將資料移入表格，`getFullEmployeeRecord` 用來取得表格中所選取之員工的完整記錄。

## 連結「姓氏」欄位

在這個步驟中，您要將**姓氏**欄位連結至 `selectedEmployeeRecord` 資料物件的 `lastName` 內容：

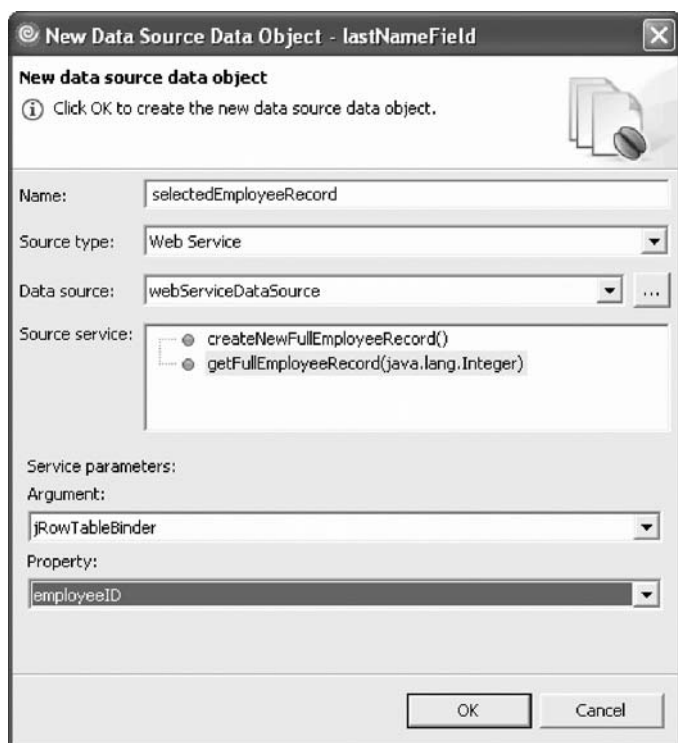
1. 在 Java Bean 視圖或設計視圖中，選取姓氏 (`lastNameField`) 的 `JTextField`。設計區域會在文字欄位上顯示**連結**標籤。



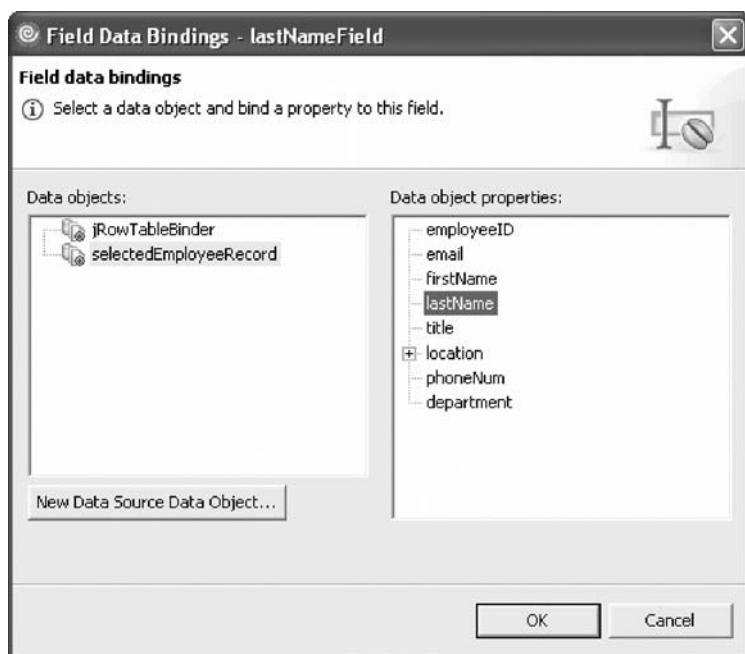
2. 按一下**連結**標籤，開啓「欄位資料連結」對話框。
3. 請按一下**新建資料來源資料物件**。雖然現有的 `JRowTableBinder` 資料物件會傳回正確的姓氏，但它不包括完整的員工記錄。您必須建立新的資料物件來代表完整的員工記錄。
4. 在**來源類型**欄位中，確定已選取 **Web 服務**，在**資料來源**中，確定已選取 **`webServiceDataSource`**。
5. 在**來源服務**清單中，選取 `getFullEmployeeRecord(java.lang.Integer)`。這時「新建資料來源資料物件」對話框會列出傳回相容於文字欄位之資料物件的服務。
6. 在**名稱**欄位中，輸入 `selectedEmployeeRecord`。
7. 在**引數**欄位中，選取 `jRowTableBinder`，在**內容**欄位中，選取 `employeeID`。所選列的員工 ID 現在設為將作為 `getFullEmployeeRecord()` 服務方法的引數。

**註：**`getFullEmployeeRecord(java.lang.Integer)` 需要整數做為引數。您要利用員工表格現行選項的員工 ID 來擷取一份完整記錄。當您連結表格時，視覺化編輯器自動產生 `JRowTableBinder`，`JRowTableBinder` 會接聽員工表的現行選項。對於整數參數，您將使用 `JRowTableBinder` 中所選取之列的 `employeeID`。





8. 按一下**確定**。
9. 在「欄位資料連結」對話框中，確定**資料物件**清單中已選取 `selectedEmployeeRecord`。請注意，`selectedEmployeeRecord` 資料物件的可用內容比 `JRowTableBinder` 資料物件多。
10. 在**資料物件內容**清單中，選取 `lastName` 內容。



11. 按一下**確定**。現在，應用程式中的姓氏欄位連結至 `selectedEmployeeRecord` 資料物件的 `lastName` 內容，它是 `getFullEmployeeRecord()` 所傳回的。

這時會建立一個名稱爲 `selectedEmployeeRecord` 的新資料物件，且會將它加入您的應用程式中。資料物件的視覺化表示法會加入設計視圖的開放式區域中，如下圖所示：



現在，當您在設計區域中選取 `lastName` 欄位時，會有一條線指示它連結到 `selectedEmployeeRecord`。在這條線中間，文字連結程式圖示代表這個連結所用的 `SwingTextComponentBinder`。如果您在設計區域中選取代表連結程式的這條線或圖示，您可以在「內容」視圖中檢查連結程式的內容。

## 連結其餘明細欄位

如果要連結員工的每個其餘詳細資料欄位，請遵循類似姓氏欄位的程序，但不需要新增資料物件。由於已新增 `selectedEmployeeRecord` 資料物件，您只需要將每一個欄位連結至 `selectedEmployeeRecord` 資料物件中相對應的內容。

如果要連結欄位，請完成下列步驟來處理應用程式員工詳細資料區段中的每個欄位：

1. 在設計視圖中選取欄位，按一下**連結**標籤。
2. 在「欄位資料連結」對話框中，從**資料物件**清單中選取 `selectedEmployeeRecord`。
3. 在**資料物件內容**清單中，選取您要連結之欄位的適當內容。下圖顯示每個文字欄位所需連結的內容：

欄位	<b>selectedEmployeeRecord</b> 資料物件中的內容
<b>lastNameField</b>	lastName
<b>firstNameField</b>	firstName
<b>idField</b>	employeeID
<b>emailField</b>	email
<b>phoneField</b>	phoneNum
<b>officeField</b>	location.office
<b>buildingField</b>	location.building
<b>siteField</b>	location.site

4. 按一下**確定**。

文字欄位連結完成之後，設計區應該看起來如下圖：





Employee details	
Last name	{lastName}
First name	{firstName}
Employee ID	{employeeID}
Contact information	
Email	{email}
Phone	{phoneNum}
Work location	
Office	{location.office}
Building	{location.building}
Site	{location.site}

## 將員工 ID 欄位設成唯讀

員工 ID 欄位是停用的，因為欄位的可編輯內容設為 `false`。不過，當資料物件包含值時，文字欄位連結程式的預設行為會在欄位上變更已啟用狀態。您可以關閉這個連結程式行為，使欄位維持最初的唯讀狀態。

如果要防止連結程式自動切換可編輯的內容，請執行下列動作：

1. 選取「員工 ID」欄位。設計區域上會顯示一條線，並以圖示  表示欄位的連結程式。
2. 按一下「員工 ID」欄位的連結程式  圖示。
3. 在「內容」視圖中，將 `autoEditable` 內容改成 **false**。按 **Enter** 鍵。

## 課程回顧

現在，當您執行應用程式以及從表格中選取某個員工時，員工記錄的詳細資料會顯示在詳細資料欄位中。

## 第 2.4 課：將「更新」按鈕連結至動作連結程式

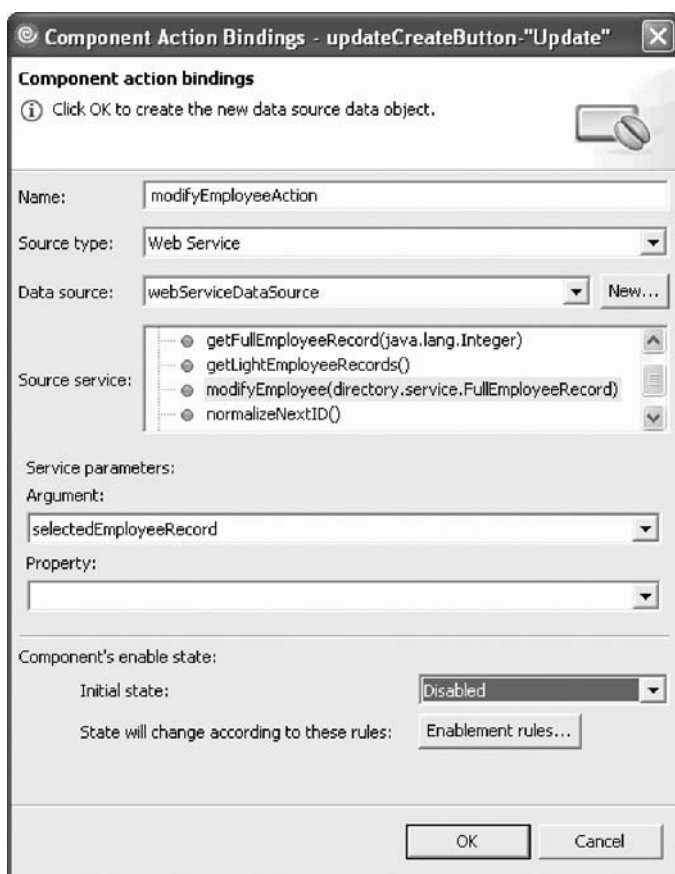
Java 視覺化編輯器提供在按一下按鈕時，用來在資料來源上呼叫服務的動作連結程式。例如，按一下「更新」按鈕時，應用程式會在 Web 服務上執行 `modifyEmployee()` 方法，並於詳細資料欄位中輸入變更。在這一課，您會將「更新」按鈕連結至動作連結程式。

如果要連結「更新」按鈕，請執行下列動作：

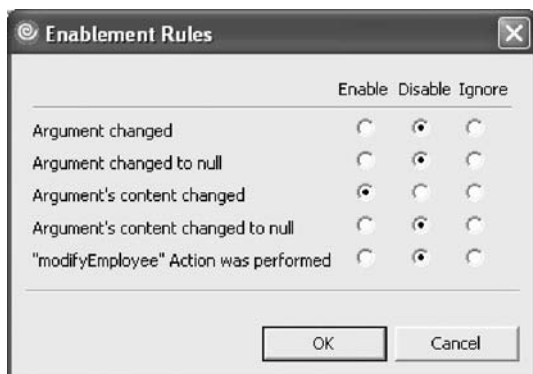
1. 在設計區域中，選取**更新**按鈕，再按一下**連結**標籤，開啟「元件動作連結」對話框。



2. 在來源類型欄位中，選取 **Web 服務**。
3. 在資料來源欄位中，選取 **webServiceDataSource**。
4. 從來源服務清單中，選取 **modifyEmployee(directory.service.FullEmployeeRecord)**。
5. 名稱欄位會自動改成 **modifyEmployeeAction**。請接受這個預設值。
6. 在引數欄位中，選取 **selectedEmployeeRecord**。
7. 由於 `modifyEmployee()` 方法以完整的員工記錄為引數，內容欄位必須保持空白。
8. 將按鈕的起始狀態設為已停用。



9. 若要定義按鈕如何變更狀態，請按一下**啟用規則**。指定只有在引數內容變更時才啟用按鈕，其他情況下皆停用。按一下**確定**。



這表示**更新**按鈕會一直停用到 selectedEmployeeRecord 的內容變更為止。換言之，當您在某個連結於 selectedEmployeeRecord 的詳細資料欄位中輸入新值時，連結程式會啟用按鈕。如果選取新的記錄或按一下**更新**，按鈕將再度停用。

10. 按一下**確定**。

這時會新增**更新**按鈕的新的 SwingDataServiceAction 連結程式。如果您在設計區域中選取這個按鈕，視覺化編輯器會畫一條線來表示按鈕已連結至 Web 服務資料來源。這時會有粉紅色的帶點箭頭從 selectedEmployeeRecord 物件指向這條線。這個箭頭表示 selectedEmployeeRecord 是服務呼叫的引數。

## 課程回顧

現在，當您執行應用程式時，您可以更新員工的記錄。

請在表格中選取一個員工來變更姓氏。當您變更姓氏時，會立即啓用**更新**按鈕。當您按一下**更新**時，會呼叫 `modifyEmployee` 服務及更新員工。員工表會反映這個新姓氏。

## 第 2.5 課：啓用「刪除」按鈕和確認對話框

在這個練習中，您將設計 `My Company Directory` 應用程式來刪除員工記錄。

下列清單說明您要應用程式使用的行為：

- 當您在表格中選取某個員工時，會啓用**刪除**按鈕。
- 當您按一下**刪除**按鈕時，會開啓「確認刪除」對話框，要求您確認刪除。
- 如果您在「確認刪除」對話框中按一下**是**按鈕，則會刪除員工記錄、關閉「確認刪除」對話框，以及重新整理員工清單。
- 如果按一下**否**，將取消刪除並關閉「確認刪除」對話框。

### 設計成根據表格中是否選取一行來決定啓用或停用「刪除」按鈕

若要將「刪除」按鈕設計成啓用或停用，請在表格中新增接聽器，當選取一行時啓用按鈕。

1. 在 Java Bean 視圖中，選取 `employeesTable`。程式碼視圖會強調顯示下面這一行：

```
employeesTable = new JTable();
```

2. 緊接在這一行後面，將新的 `ListSelectionListener` 和 `valueChanged` 事件新增至 `employeesTable`：

```
employeesTable.getSelectionModel().addListSelectionListener(new ListSelectionListener() {  
    public void valueChanged(ListSelectionEvent e) {  
        getDeleteButton().setEnabled(getEmployeesTable().getSelectedRowCount() != 0);  
    }  
});
```

3. 新增這幾行程式碼之後，程式碼編輯器會將它們標示為錯誤，直到您匯入 `ListSelectListener` 和 `ListSelectionEvent` 為止。若要新增必要的匯入，請在主功能表按一下**程式碼** → **組織匯入**。這時會在類別的匯入區段中新增下面幾行：

```
import javax.swing.event.ListSelectionEvent;  
import javax.swing.event.ListSelectionListener;
```

現在，當選取表格中的某列時，會啓用**刪除**按鈕。

### 設計成按一下「刪除」時開啓「確認刪除」對話框

將 `actionPerformed` 事件新增至「刪除」按鈕，並設計事件來開啓「確認刪除」對話框。

1. 用滑鼠右鍵按一下**刪除**按鈕，選取**事件** → **actionPerformed**。下列事件 Stub 會新增至 `getDeleteButton()` 方法：

```
deleteButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent e) {  
        System.out.println("actionPerformed()");  
        // TODO 自動產生的事件 Stub actionPerformed()  
    }  
});
```

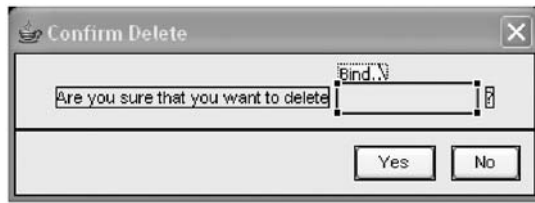
2. 將這個產生的 Stub 取代為下列程式碼，設為按一下按鈕時顯示「確認刪除」對話框：

```
deleteButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent e) {  
        getConfirmDialog().setVisible(true);  
    }  
});
```

## 連結「確認刪除」對話框中的文字欄位

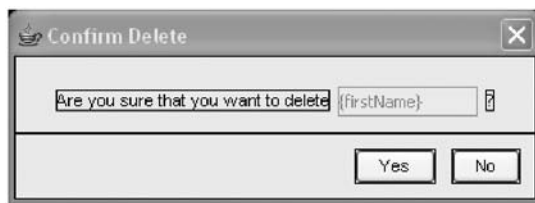
連結「確認刪除」對話框中的文字欄位來顯示要刪除的員工的名字。

1. 在 Java Bean 視圖或設計區域中，選取 `employeeToDeleteField` 文字欄位，再按一下連結標籤。



2. 在「欄位資料連結」對話框中，選取 `selectedEmployeeRecord` 資料物件和 `firstName` 欄位，再按一下確定。

文字欄位現在連結至 `employeesTable` 中選取列的 `firstName` 直欄。



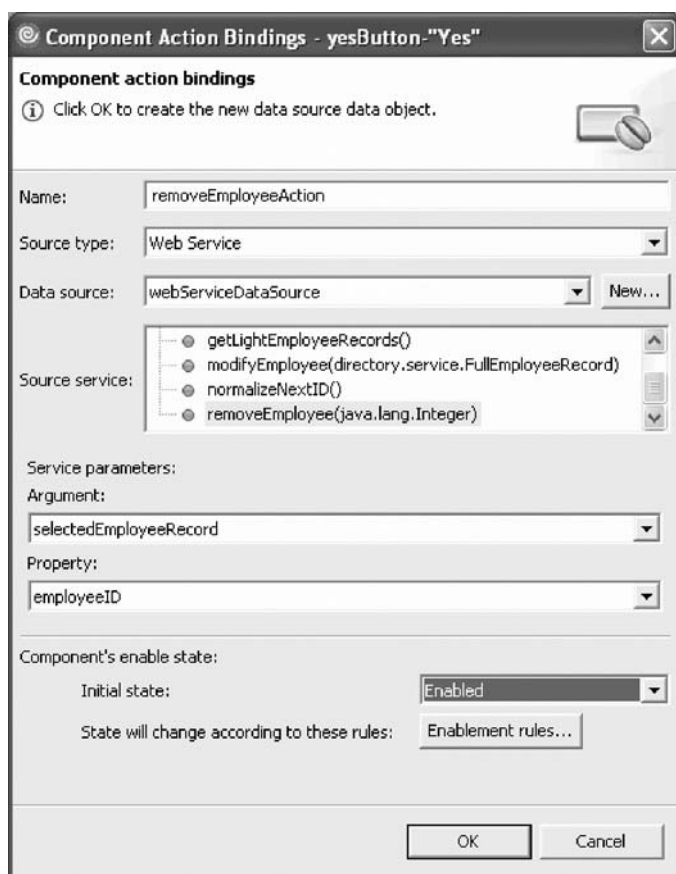
3. 如果要確定這個欄位是唯讀，請將欄位連結程式的 `autoEditable` 內容設為 `false`。

## 連結「是」按鈕來執行刪除

連結是按鈕，在 Web 服務上呼叫 `removeEmployee(java.lang.Integer)` 方法。

1. 選取是按鈕，再按一下連結標籤，開啓「元件動作連結」對話框。
2. 在來源類型欄位中，選取 **Web 服務**。
3. 在資料來源欄位中，選取 **webServiceDataSource**。
4. 從來源服務清單中，選取 **removeEmployee(java.lang.Integer)**。
5. 名稱欄位會自動改成 **removeEmployeeAction**。請接受這個預設值。
6. 在引數欄位中，選取 **selectedEmployeeRecord**。
7. 在內容欄位中，選取 **employeeID**。由於 `removeEmployee()` 方法以整數為引數，因此，您使用 `selectedEmployeeRecord` 的員工 ID。
8. 將按鈕的起始狀態設為已啓用。
9. 在啓用規則中，每一個條件都選取忽略。

這個元件狀態表示一律啓用「是」按鈕，因為不需要變更狀態。



10. 按一下**確定**。

## 新增事件在刪除員工之後隱藏「確認刪除」對話框

在這個步驟中，您新增事件至是按鈕的連結程式（不是是按鈕本身）。您想要在移除員工之後，也就是在連結程式順利呼叫資料來源的服務之後，關閉「確認刪除」對話框。

將下列程式碼新增至 `getRemoveEmployeeAction()` 方法：

```
removeEmployeeAction.addActionBinderListener(new jve.generated.IActionBinder.ActionBinderListener() {
    public void afterActionPerformed(jve.generated.IActionBinder.ActionBinderEvent e) {
        getConfirmDialog().setVisible(false);
    }
    public void beforeActionPerformed(jve.generated.IActionBinder.ActionBinderEvent e) {}
});
```

這個事件程式碼會在執行連結程式的動作之後，隱藏「確認刪除」對話框。

## 課程回顧

現在，當您執行 `My Company Directory` 應用程式時，您可以在表格中選取員工，按一下**刪除**按鈕，再按一下**是**來確認刪除。這時會從目錄中移除這個員工記錄，員工清單會反映移除的結果。

## 第 2.6 課：設定動作和連結來新增員工

在這一課，您將要啟動 `My Company Directory` 應用程式以新增員工記錄。

由於應用程式新增新員工的行為動態而複雜，因此，這個練習先天上就比較複雜，程式碼必須進行一些手動變更。另外，這個練習也示範了資料物件的一些進階功能，且提供了可配合您的需求來使用連結程式和資料物件的創造性的方式範例。

下列清單說明應用程式的必要行為：

- 按一下**新建**按鈕時會發生下列行為：
  - 取消選取員工表的選項，停用這份表格。
  - 清除表格選項會導致**刪除**按鈕停用。
  - 停用**過濾器**欄位。
  - 除了新員工 ID 之外，會清除詳細資料欄位的任何值。
  - **更新**按鈕上的文字會改成**新增**。
- 按一下**新增**按鈕時會發生下列行為：
  - 輸入詳細資料欄位的值會新增到目錄中，成為一項新的員工記錄。
  - 啟用表格，重新整理值。
  - 啟用**過濾器**欄位。
  - **新增**按鈕上的文字會切換回**更新**。

## 新增資料來源資料物件來呼叫 `createNewFullEmployeeRecord()`

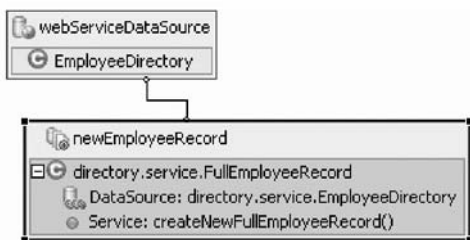
Web 服務範例提供一項 `createNewFullEmployeeRecord` 服務，這項服務則提供一份將移入下個可用的員工 ID 號碼的新空白員工記錄。然後，就可以在這份空白記錄中填入新員工的資訊，並送回至 Web 服務。

1. 在 Java 視覺化編輯器的選用區中，展開資料物件抽屜，選取**資料來源資料物件**。
2. 將滑鼠指標移到設計視圖的空白區域或開放式區域，按一下滑鼠左鍵來放下資料來源資料物件。這時會在開放式區域中，新增和顯示新的資料來源資料物件：



3. 用滑鼠右鍵按一下「資料來源資料物件」，選取**重新命名欄位**。請將資料物件重新命名為 `newEmployeeRecord`。
4. 用滑鼠右鍵按一下 `newEmployeeRecord` 資料物件，選取**連結內容**。這時會開啓「資料連結」對話框。
5. 在**資料來源**欄位中，選取 `webServiceDataSource`
6. 在**服務**欄位中，選取 `createNewFullEmployeeRecord()`
7. 按一下**確定**。

在開放式區域中，您會看到 `newEmployeeRecord` 資料來源資料物件已連結至 Web 服務。



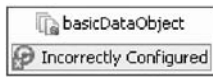
## 新增基本資料物件來協助切換資料物件

由於詳細資料欄位和「更新」按鈕需要切換模式（用來執行更新和建立新員工），因此，它們必須在不同時間連結至兩個不同資料物件。為了使這個步驟更容易執行，您將新增一個名稱爲 `switchingDataObject` 的基本資料物件。您將利用這個「基本資料物件」在 `selectedEmployeeRecord` 和 `newEmployeeRecord` 之間切換文字欄位的連結。



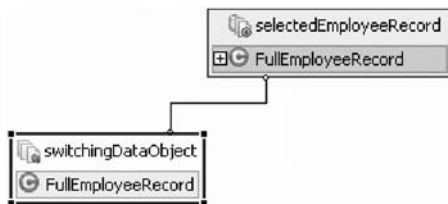
新的基本資料物件只會指向先前的練習所定義的另一個資料物件 (selectedEmployeeRecord)。當您建立一個方法來通知這個基本資料物件使用先前所建立的 newEmployeeRecord 時，這個新的資料物件會變得很有用。換言之，這個基本資料物件會扮演中間資料物件的角色來切換 selectedEmployeeRecord 資料物件和 newEmployeeRecord 資料物件，使應用程式中的視覺化元件能夠使用兩種不同的資料物件。

1. 在視覺化編輯器選用區中，選取**基本資料物件**，將它放在開放式區域中。這樣會新增 basicDataObject。



2. 將資料物件重新命名為 switchingDataObject
3. 在 switchingDataObject 的「內容」視圖中，將 **sourceObject** 內容設成 **selectedEmployeeRecord**。您可以從內容的「值」直欄下拉功能表中，選取 selectedEmployeeRecord。

現在，switchingDataObject 參照 selectedEmployeeRecord 且反映相同的值：

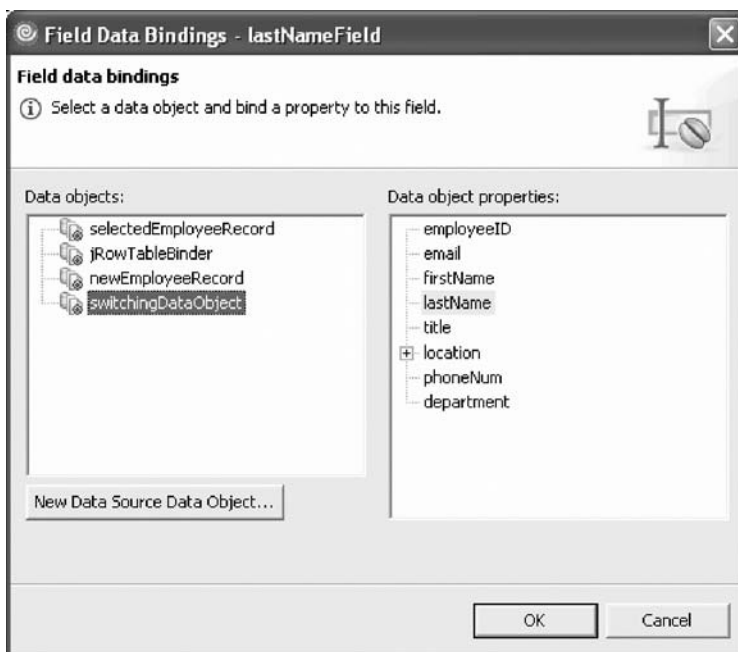


## 將每一個員工欄位重新連結至 switchingDataObject

雖然每個員工詳細資料欄位都已經連結至 selectedEmployeeRecord，現在，您仍要將它們連結至 switchingDataObject。連結欄位之後，您可以動態切換欄位的資料物件，視您在修改現有的員工記錄或新增員工記錄而定。

對於「員工詳細資料」區段中的每一個欄位，請完成下列步驟：

1. 選取欄位，按一下**連結**標籤。
2. 在「欄位資料連結」對話框中，選取 switchingDataObject。先前已將欄位連結至 selectedEmployeeRecord。



3. 確定欄位仍連結至正確的資料物件內容，再按一下**確定**。如果您在設計視圖中選取了欄位，您可能會發現，連結程式線條現在指向 `switchingDataObject`。



## 定義旗標和方法來更新和切換模式

下列 `updateMode()` 方法會檢查模式旗標是否設為新建，再據此變更應用程式的行為。依預設，`Boolean` 旗標 `isNewMode` 設為 `false`，`updateMode()` 方法會啟用員工表和過濾欄位，並將「更新」按鈕的文字設為「更新」。如果 `isNewMode` 設為 `true`，則會停用員工表並取消選取任何選項、過濾欄位會停用，且「更新」按鈕的文字會設為「新增」。

請將下列程式碼新增至 `DirectoryApp.java` 類別中，就在最後一個右大括弧之前：

```
private boolean isNewMode = false;
private void updateMode() {
    if (isNewMode) {
        getEmployeesTable().clearSelection();
        getEmployeesTable().setEnabled(false);
        getFilterField().setEditable(false);
        getUpdateCreateButton().setText("Add");
    } else {
        getEmployeesTable().setEnabled(true);
        getFilterField().setEditable(true);
        getUpdateCreateButton().setText("Update");
    }
}
```

## 將 `actionPerformed` 事件新增至「新建」按鈕

在這個步驟中，您新增按一下**新建**按鈕時的事件碼。事件會指示 `switchingDataObject` 使用 `newEmployeeRecord` 資料物件、將模式旗標設為 `new`，並執行您在上一個新增的 `updateMode()` 方法。

1. 在設計視圖中，用滑鼠右鍵按一下**新建**按鈕，選取事件 → **`actionPerformed`**。`getNewButton()` 方法中會產生下列程式碼：

```
newButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent e) {
        System.out.println("actionPerformed()"); // TODO 自動產生的事件 Stub actionPerformed()
    }
});
```

2. 將產生的 `Stub` 取代為下列程式碼：

```
newButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent e) {
        getSwitchingDataObject().setSourceObject(getNewEmployeeRecord());
        getNewEmployeeRecord().refresh();
        isNewMode = true; //將應用程式設成新建模式
        updateMode(); //根據新建模式來變更 UI
        getLastNameField().grabFocus();
    }
});
```

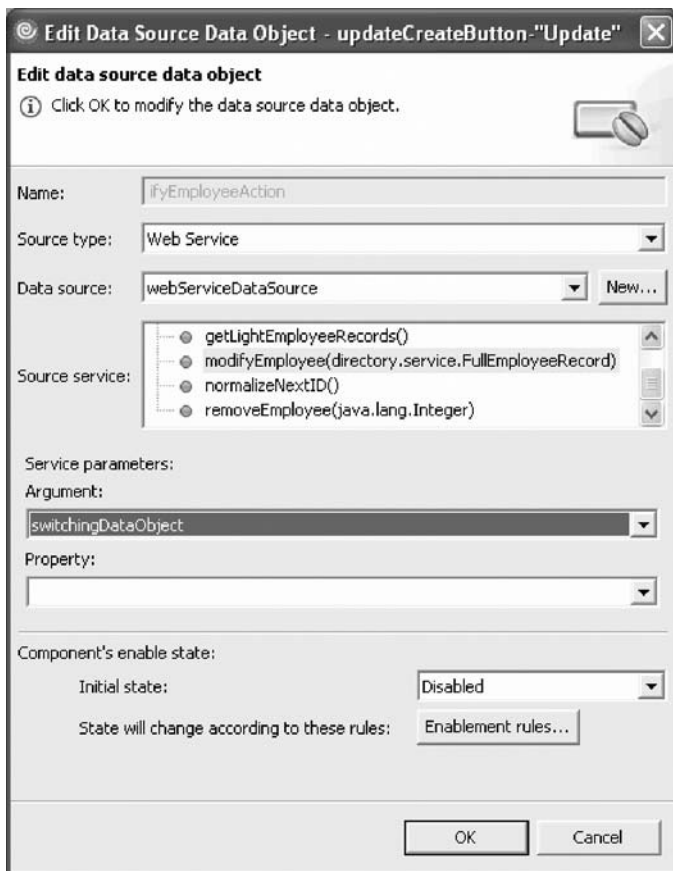
## 重新連結「更新」按鈕

在上一個，您已設計**更新**按鈕來使用 Web 服務的 `modifyEmployee` 方法。這個動作實作成 `SwingDataServiceAction`。`SwingDataServiceAction` 的內容之一是用來作為服務引數的來源物件。修改動作的來源物件目前設為 `selectedEmployeeRecord`。為了設計按鈕來控制更新和新增，您將重新配置按鈕的動作來使用 `switchingDataObject` 做為 `modifyEmployee` 服務的引數。

1. 在設計視圖中，選取**更新**按鈕。請注意，粉紅色虛線箭頭表示 `selectedEmployeeRecord` 是服務呼叫的引數。

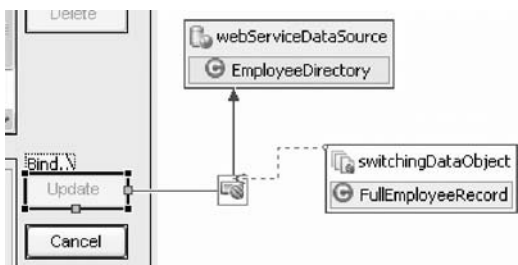


2. 按一下**更新**按鈕上的連結標籤。
3. 在**引數**欄位中，選取 `switchingDataObject`。



4. 按一下**確定**。

現在，請注意，按鈕的動作現在配置成以 `switchingDataObject` 為 `modifyEmployee` 方法的引數：



## 將事件新增至「更新」按鈕的連結程式來重設模式

按一下**更新**按鈕且在 Web 服務上完成動作之後，您要應用程式回復它的預設模式和行為。如果要執行這個動作，您要在按鈕的動作連結程式上新增一個事件接聽器，以便在執行更新或新增之後，更新模式和重新整理表格。

將下列程式碼新增至「更新」按鈕的 `getModifyEmployeeAction()` 方法：

```
modifyEmployeeAction.addActionBinderListener(
    new jve.generated.IActionBinder.ActionBinderListener() {
        public void afterActionPerformed(jve.generated.IActionBinder.ActionBinderEvent e) {
            if (isNewMode) {
```

```

        //返回使用 selectedEmployeeRecord
        getSwitchingDataObject().setSourceObject(getSelectedEmployeeRecord());
        //離開新建模式
        isNewMode = false;
        updateMode();
    }
    // 重新整理表格的資料物件
    getLightEmployeeRecordRows().refresh();    }
    public void beforeActionPerformed(jve.generated.IActionBinder.ActionBinderEvent e) {}
});

```

## 課程回顧

現在，當您執行 My Company Directory 應用程式時，您可以按一下**新建**按鈕來新增員工記錄。

## 第 2.7 課：設計「取消」按鈕行為

當使用應用程式時，如果您決定不要送出變更，您會希望能夠輕易取消您在某項員工記錄上開始進行的任何變更。換言之，您需要能夠取消和清除欄位，以便重新開始。如果要新增這項功能，您可以在**取消**按鈕上設定一些 `actionPerformed` 事件。

下列清單說明**取消**按鈕所需要的行為：

- 如果您在新建模式中按一下**取消**按鈕，應用程式會離開新建模式。
- 如果在修改員工記錄時按一下**取消**按鈕，您已變更的任何值將回復為原始值。

如果要將 `actionPerformed` 事件新增至**取消**按鈕中來執行必要的行為，請執行下列動作：

1. 在設計視圖中，用滑鼠右鍵按一下**取消**按鈕，選取事件 → **actionPerformed**。 `getCancelButton()` 方法中會產生下列程式碼：

```

cancelButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent e) {
        System.out.println("actionPerformed()"); // TODO 自動產生的事件 Stub actionPerformed()
    }
});

```

2. 將產生的事件 `Stub` 取代為下列程式碼：

```

cancelButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent e) {
        if (isNewMode) {
            getSwitchingDataObject().setSourceObject(getSelectedEmployeeRecord());
            isNewMode = false;
            updateMode();
        } else {
            getSelectedEmployeeRecord().refresh();
        }
    }
});

```

## 課程回顧

在這一課，您已學到如何以 `actionPerformed` 事件來設計**取消**按鈕。

## 第 2.8 課：在員工表設定過濾器

您可以利用文字過濾器連結程式來過濾員工表的內容。過濾器取得文字欄位的輸入，再根據表格中的特定內容或直欄來過濾表格。

在應用程式中，您將利用**過濾器**欄位所輸入的字元來依員工姓氏進行過濾。如果員工記錄的姓氏完全符合**過濾器**欄位中所輸入的值，這筆員工記錄就會出現在表格中。

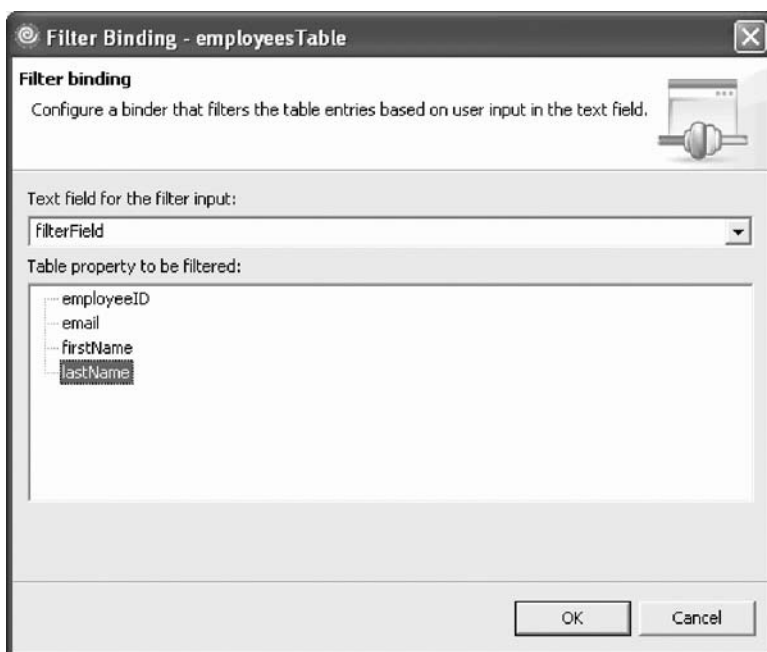
Filter:  (Last name)

Last name	First name	Email	Employee ID
Maxwell	Seth	seth.maxwell@my...	24561
Maxwell	Aubrey	aubrey.maxwell@...	30089
Martinez	James	james.martinez@m...	31780

New Delete

如果要建立表格的過濾器，請執行下列動作：

1. 選取 `employeesTable` 的連結程式圖示，再選取**過濾器連結內容**。這時會開啓「過濾器連結」對話框。
2. 在過濾器輸入文字欄位清單中，選取 `filterField`。
3. 在**要過濾的表格內容**清單中，選取 `lastName`。



4. 按一下**確定**。

這時會產生新的 `SwingPropertyFilter`。表格連結程式的過濾器內容會設成使用新過濾器。新過濾器配置成使用**過濾器欄位**作為輸入，並在表格的 `lastName` 內容上進行過濾。

## 課程回顧

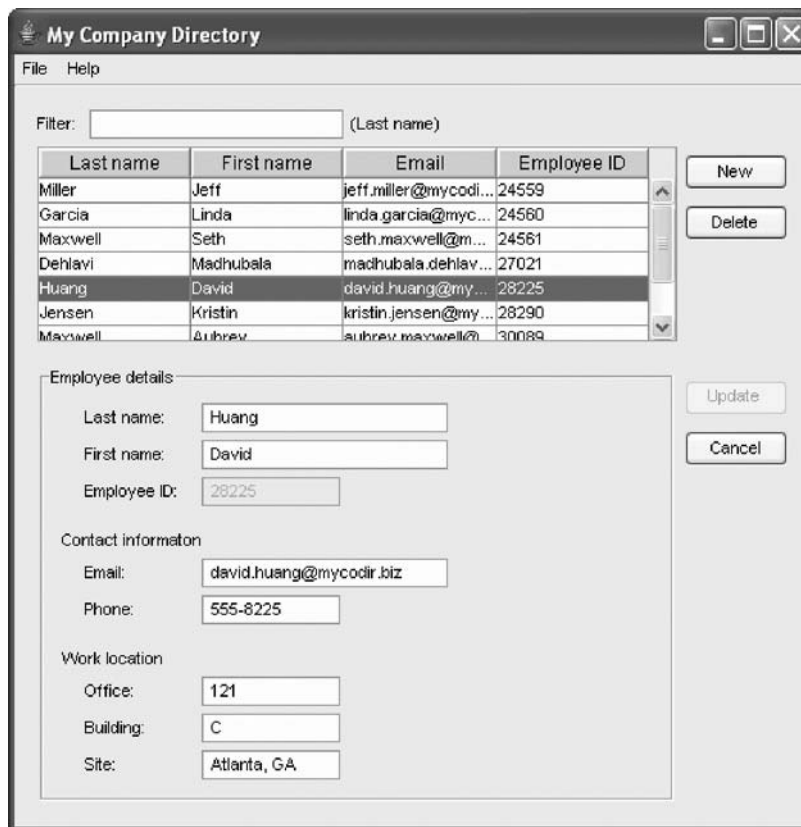
在這一課，您已瞭解如何設定表格的過濾器。

現在，當您執行 `My Company Directory` 應用程式時，您可以在**過濾器欄位**中鍵入字元，表格將會過濾來顯示姓氏中含有輸入字元的列。

恭喜您！`My Company Directory` 應用程式已完成。

## 摘要：建置使用 Web 服務的豐富型 Java 用戶端

恭喜您！您已瞭解如何利用 Java 視覺化編輯器來建置 My Company Directory 應用程式，這是一個豐富型 Java 用戶端，將連接 Web 服務範例來維護員工目錄。



檢視完成產品的圖片：

## 學習成果

您利用視覺化編輯器來完成圖形式使用者介面，利用 GridBagLayout 來安排員工表。之後，您將表格、欄位和按鈕連結至適當的資料物件和資料來源，使應用程式使用您產生的 Web 服務 Java Proxy。您也編寫了一些複雜的程式碼，使應用程式有適當行為，既容易使用，又符合直覺。另外，您也學會如何將企業應用程式安裝在 WebSphere Application Server 6.0 版中以及部署 Web 服務。

最重要的是，您學習了完整 Java 視覺化編輯器所提供用來處理資料的功能強大的連結程式類別。現在您已準備好開始獨立嘗試在新鮮有趣的用途下使用連結程式。

您現在應該能夠執行下列作業：

- 利用 Java 視覺化編輯器，將元件佈置在 GridBagLayout 中。
- 將視覺化類別當作 JavaBean 來執行。
- 將 Java 應用程式的視覺化元件連結至 Web 服務所傳回的方法和資料物件。
- 將事件新增至視覺化元件中。

## 其他資源

匯入 My Directory 應用程式完成版本

這個專案包含已完成的應用程式、含連結程式類別的 `jve.generated` 套件，以及針對 WebSphere Application Server 6.1 版所配置的 Web 服務 Java 用戶端。如果未經過指導教學就直接匯入這個已完成的專案，則可能需要配置 Java 建置路徑變數。必須指向 WebSphere 6.1 版 Web 服務小型用戶端 JAR 檔。

**提示：** 除非在匯入時指定不同的專案名稱，否則會改寫您的 MyDirectory 專案內容。



## 讀者意見表

爲使本書盡善盡美，本公司極需您寶貴意見；懇請您閱讀後，撥冗填寫下表，惠予指教。

請於下表適當空格內，填入記號（√）；我們會在下一版中，作適當修訂，謝謝您的合作！

評估項目	評 估 意 見	備 註
正 確 性	內容說明與實際程序是否符合 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	參考書目是否正確 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
一 致 性	文句用語及風格，前後是否一致 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	實際產品介面訊息與本書中所提是否一致 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
完 整 性	是否遺漏您想知道的項目 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	字句、章節是否有遺漏 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
術語使用	術語之使用是否恰當 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	術語之使用，前後是否一致 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
可 讀 性	文句用語是否通順 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	有否不知所云之處 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
內容說明	內容說明是否詳盡 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	例題說明是否詳盡 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
排版方式	本書的形狀大小，版面安排是否方便閱讀 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	字體大小，顏色編排，是否有助於閱讀 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
目錄索引	目錄內容之編排，是否便於查找 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	索引語錄之排定，是否便於查找 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	※評估意見爲"否"者，請於備註欄提供建議。	

其他：（篇幅不夠時，請另外附紙說明。）

---

---

---

---

---

---

---

---

---

---

上述改正意見，一經採用，本公司有合法之使用及發佈權利，特此聲明。

註： 您也可將寶貴的意見以電子郵件寄至 [tscadmin@tw.ibm.com](mailto:tscadmin@tw.ibm.com)，謝謝。

建置使用 Web 服務的豐富型 Java 用戶端

折疊線

台北市 110 信義區松仁路 7 號 3 樓

臺灣國際商業機器股份有限公司  
大中華研發中心 軟體國際部

啟



廣 告 回 信
臺灣地區郵政管理局 登記
北台字第
號

(免貼郵票)

寄件人 姓名：  
地址：

寄

折疊線

讀者意見表





