## 12.2 使用 Java 開發 Windows Azure 應用程式

在使用 Java 開發 Windows Azure 應用程式前，我們需要先準備好它的開發工具，除了標準的 Java SDK 以及 Java 執行環境外，還需要一個開發工具，在這裡我們直接使用目前最受歡迎的 Java IDE－Eclipse[1]作為 Java 開發 Windows Azure 應用程式的工具，微軟也特別在 Eclipse 上開發了一個 Windows Azure 的工具，稱為 Windows Azure Toolkit for Java，可相容於 Eclipse Juno SR1 與 Helios SR2 版本。

於 2013/5/1，微軟發行了 Windows Azure Tools for Eclipse 2.0，將雲端服務訂閱帳戶與儲存帳戶等資訊的整合管理做得更齊全了，開發人員只需要像在 Visual Studio 內匯入發行設定檔即可輕鬆整合 Windows Azure 上的所有訂閱帳戶。

安裝這個工具很簡單：

Step 1. 在 Eclipse 的 Help 功能表中選擇 Install New Software:



圖 12.2　安裝新軟體

---

[1] 可參考 http://faq.programmerworld.net/programming/best-java-ide-review.html
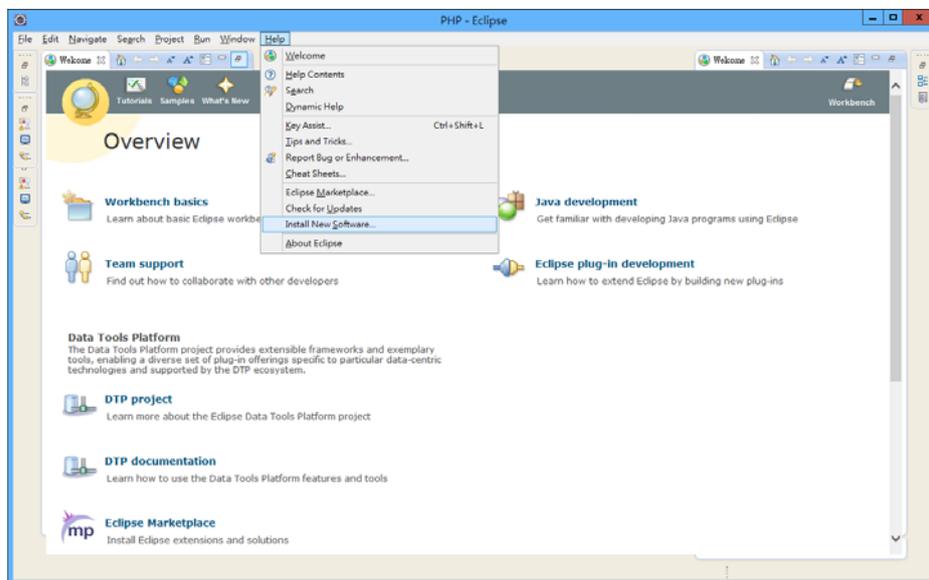
Step 2. 在安裝的對話盒的 Work With 選單中輸入
『http://dl.msopentech.com/eclipse』，並按下 ENTER，即可看到 Windows
Azure Toolkit for Java 的資訊，請全選：
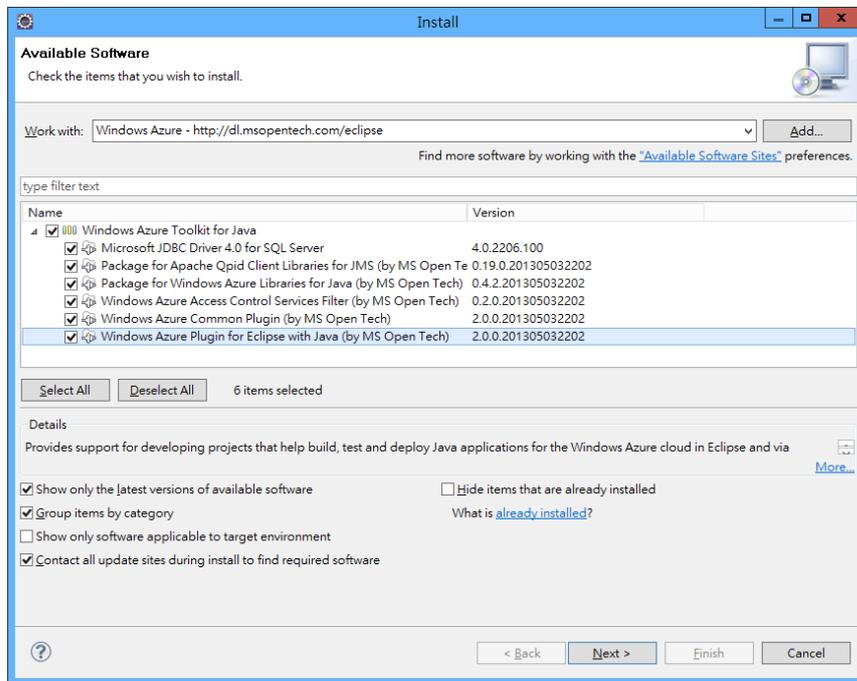


圖 12.3　搜尋 Windows Azure Tools for Eclipse

Step 3. 按下 Next，會進入確認安裝資訊的畫面，確認這次安裝會有哪些軟體
或套件會被安裝到電腦內：
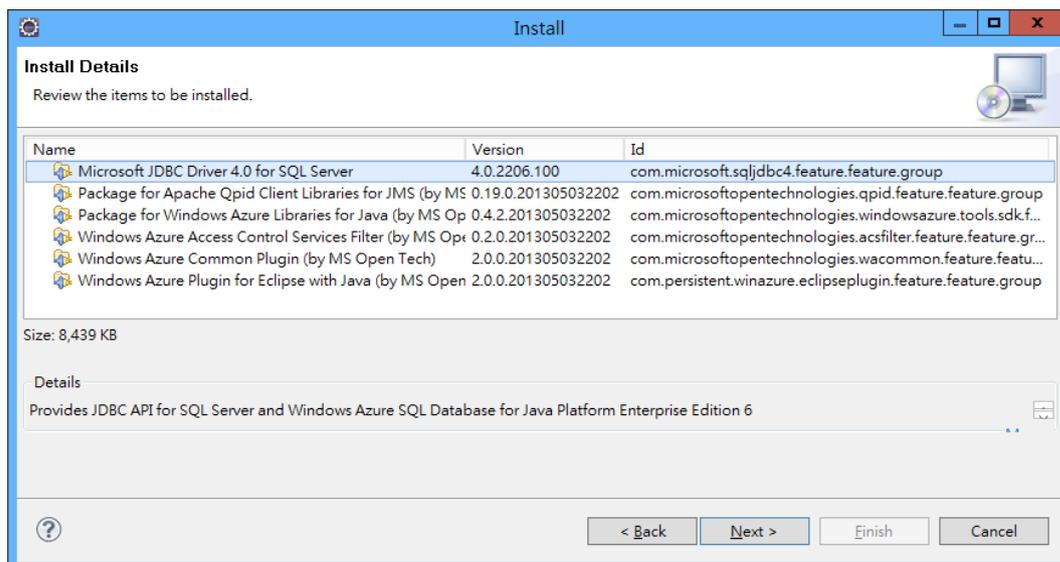


圖 12.4　確認 Windows Azure Tools for Eclipse 套件

Step 4. 按 Next，會進入授權合約的畫面：
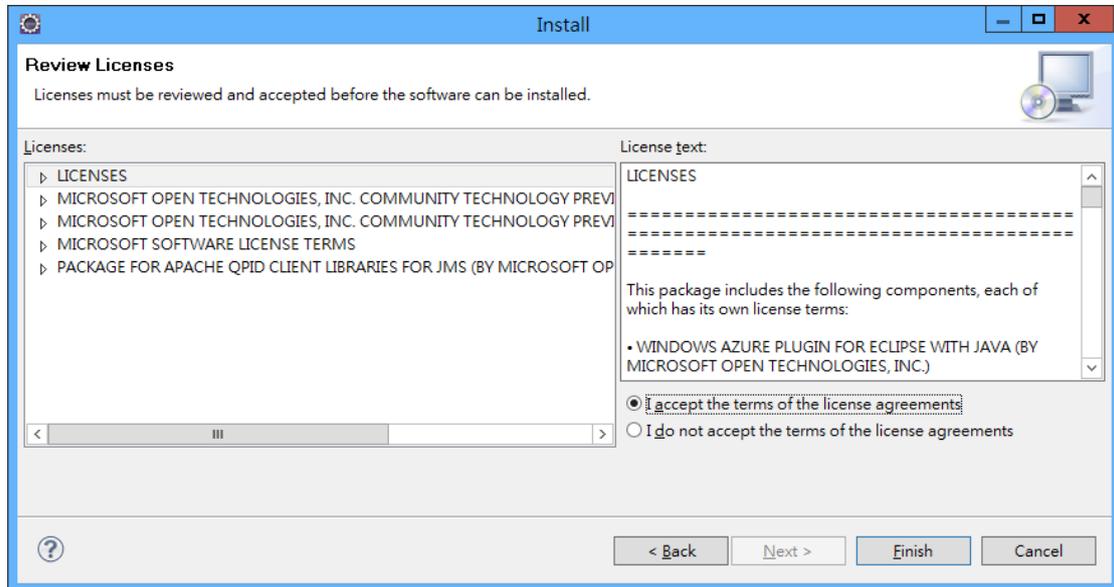


圖 12.5 授權合約檢視

請選擇『I accept the terms of the license agreement』後，按 Finish，即可開始安裝
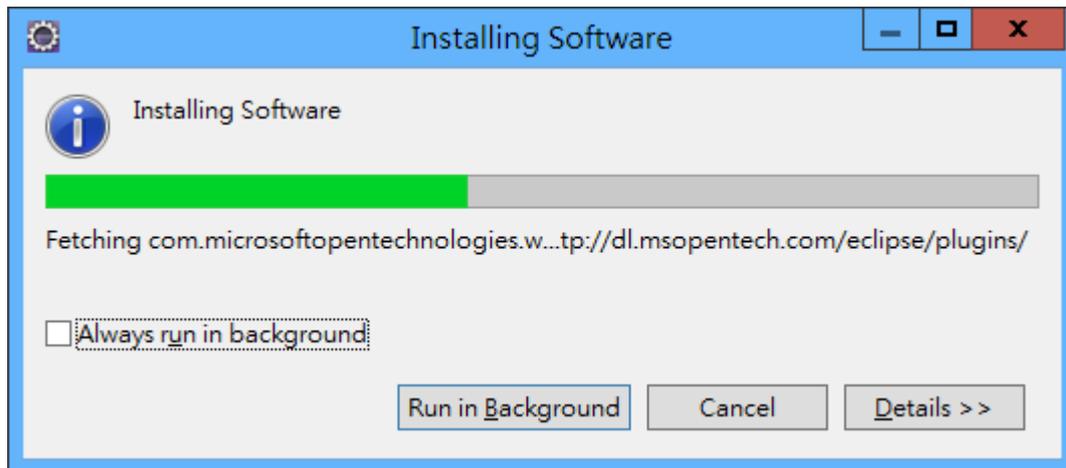


圖 12.6 安裝 Windows Azure Tools for Eclipse 套件

Step 5. 安裝完成後，重新啟動 Eclipse 即可，打開 New->Project，能在專案的範本中看到 Windows Azure Deployment Project，我們可以運用這個專案檔來封裝與發行 Java 的專案。
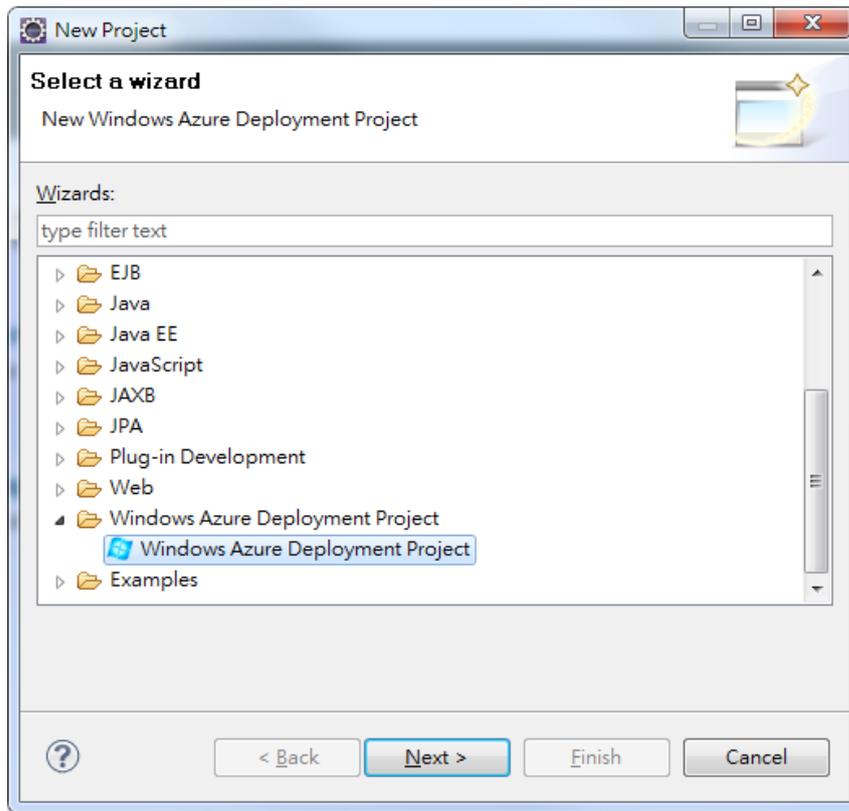
圖 12.7　新增 Windows Azure 專案

---

**NOTE**
若是現有 *v1.8* 工具的使用者，請使用 *Check for Updates* 的功能來取得新的 *Windows Azure Tools* 工具。

---

準備完成後，我們就可以新增一個 JavaHelloAzure 的專案，來簡單的說明怎麼開發可掛上 Windows Azure Cloud Service 環境的 JSP 應用程式。

Step 1. 請在 New->Project 開啟新專案精靈，並選擇 Web 內的 Dynamic Web Project (即 JSP 應用程式)：
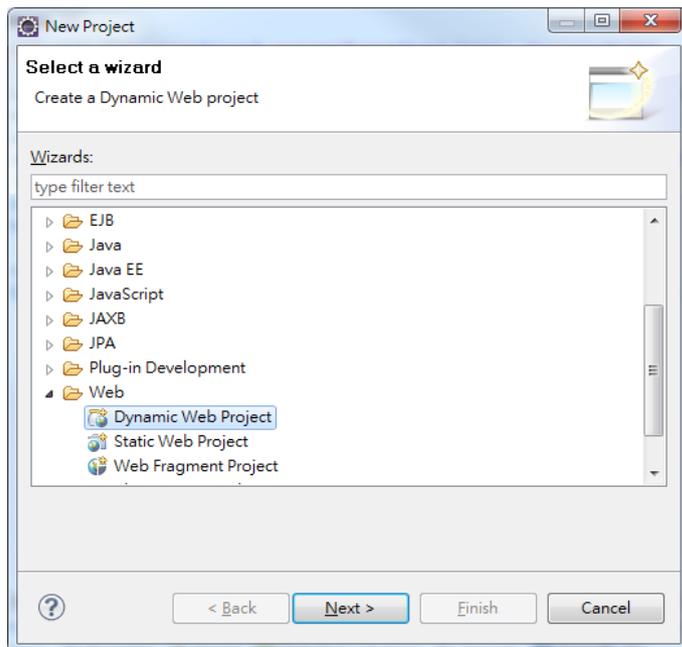
圖 12.8 新增 Dynamic Web Project (JSP) 專案

Step 2. 按 Next，出現 Dynamic Web Project 的設定，只要在 Project name 欄位中填入 JavaHelloAzure 即可，然後按 Finish：
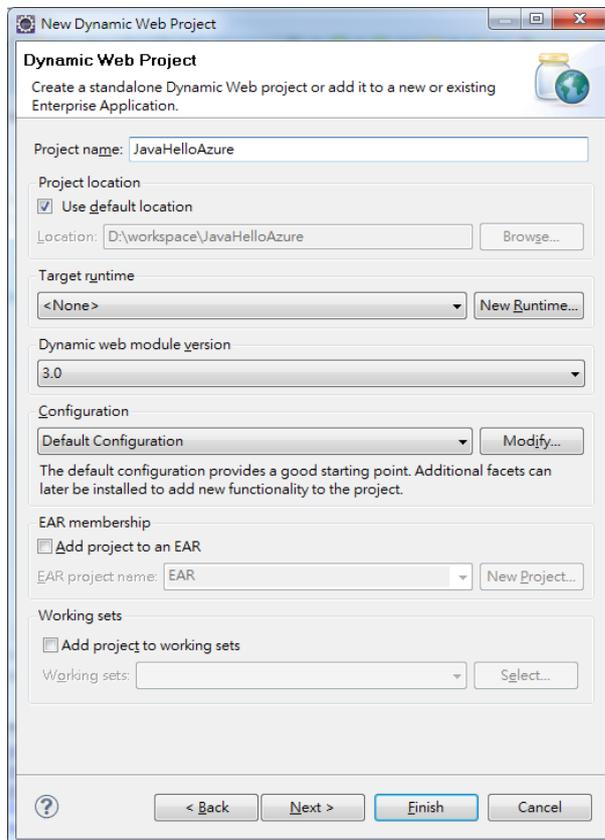


圖 12.9 設定新專案屬性

Step 3. 專案建立完成後，進入 Eclipse 的專案瀏覽器，並在 WebContent 中新增一個 index.jsp 檔案，於 body 中填入下列程式碼：

```
<body>
 <b><% out.println("Hello Azure!"); %></b>
</body>
```

Step 4. 在 Project Explorer 內的 JavaHelloAzure 專案上按右鍵，選擇 Windows Azure 項目，並選擇 Packaging for Windows Azure 項目：
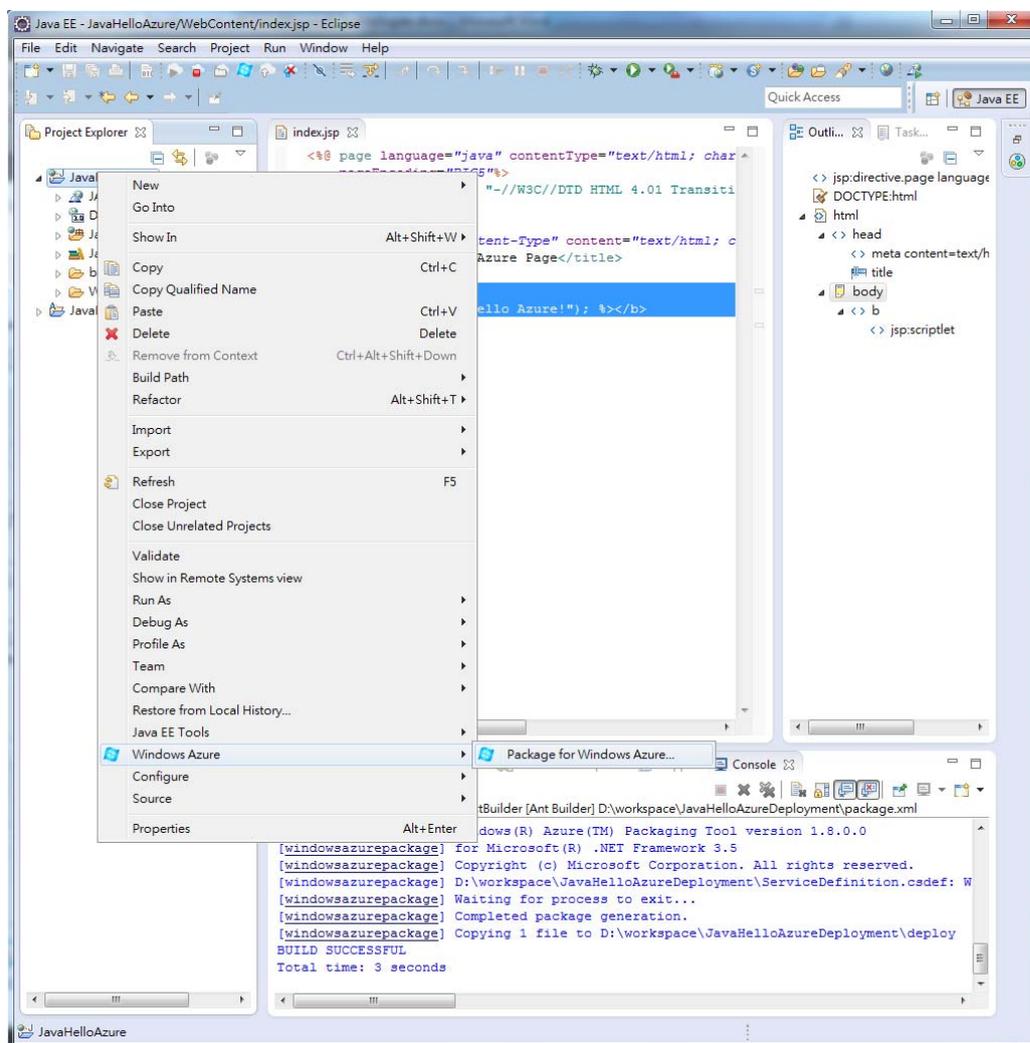


圖 12.10 新增 Windows Azure 發行專案

此時會出現新增 Windows Azure 專案的視窗，在專案名稱欄位輸入 JavaHelloAzureDeploy：
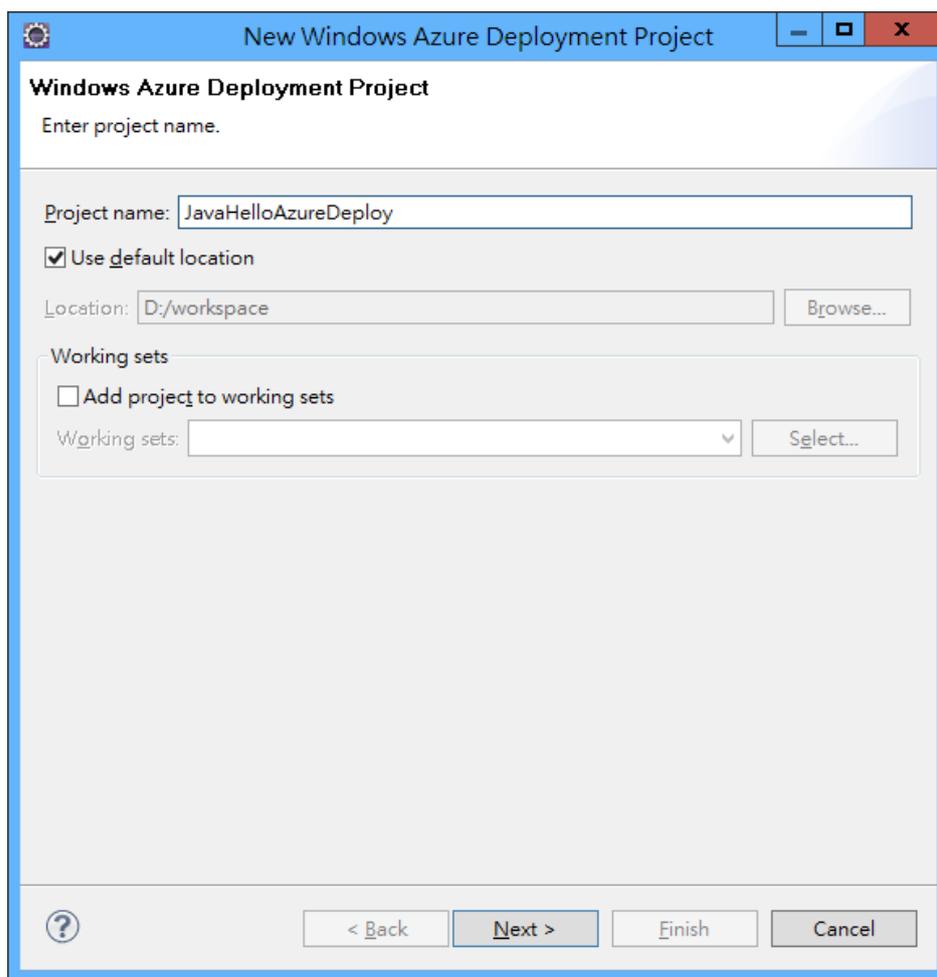
圖 12.11　設定 Windows Azure 發行專案

Step 5. 按 Next，接著會出現套件的設定，開發人員需要設定 JDK/Tomcat 以及應用程式的部署資訊。

在 Windows Azure Tools for Eclipse v1.8[2]時，JDK/Tomcat/Application 的設定是放在一起的，但在 v2.0 時，微軟將這三個設定項目分成分頁的方式，除了包含新增的功能外，也讓設定的畫面更加的清楚。

---

[2] 也就是書付梓時所用的版本。
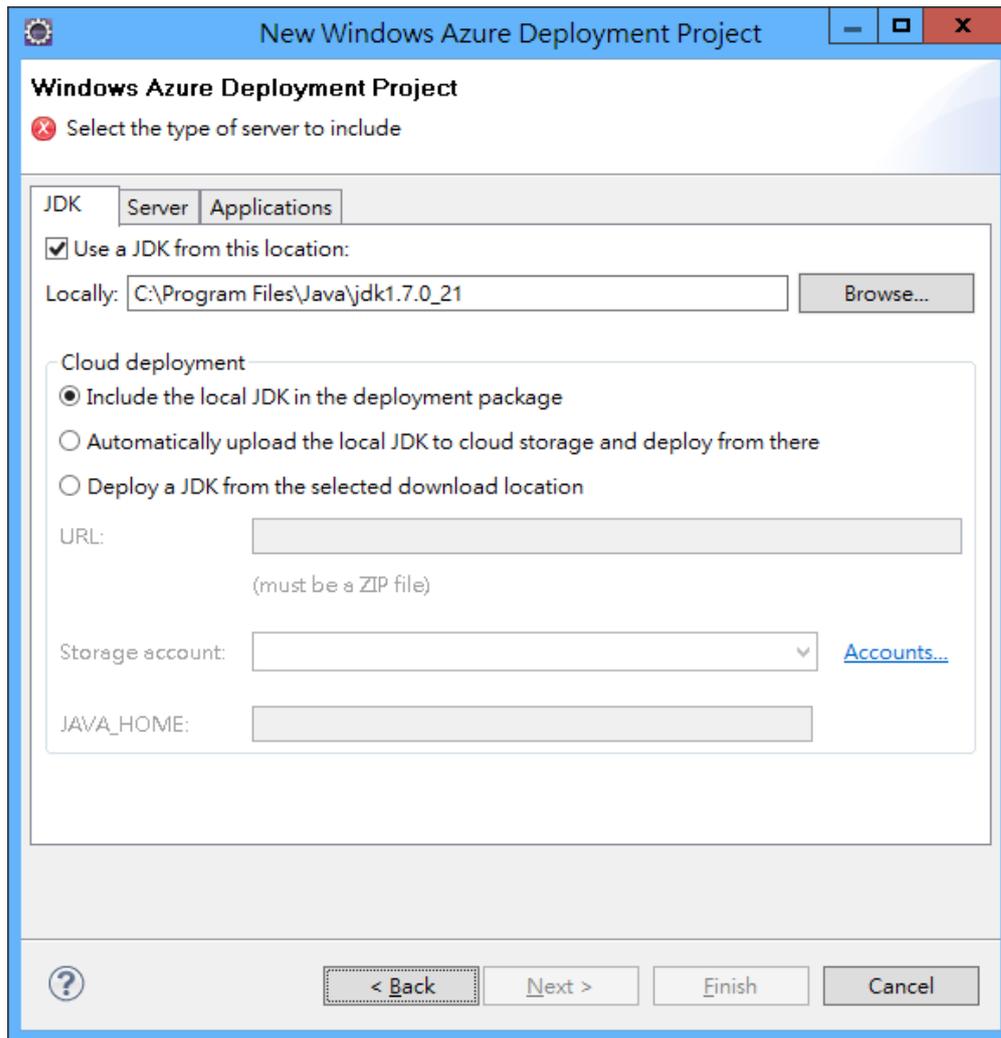
圖 12.12a 設定要包含在 Windows Azure 發行專案內的 JDK 來源位置

JDK 頁籤所包含的是 JDK 的選項，包含本地路徑的來源以及部署時的選項，
而在 Server 頁籤則是 Tomcat 的選項：
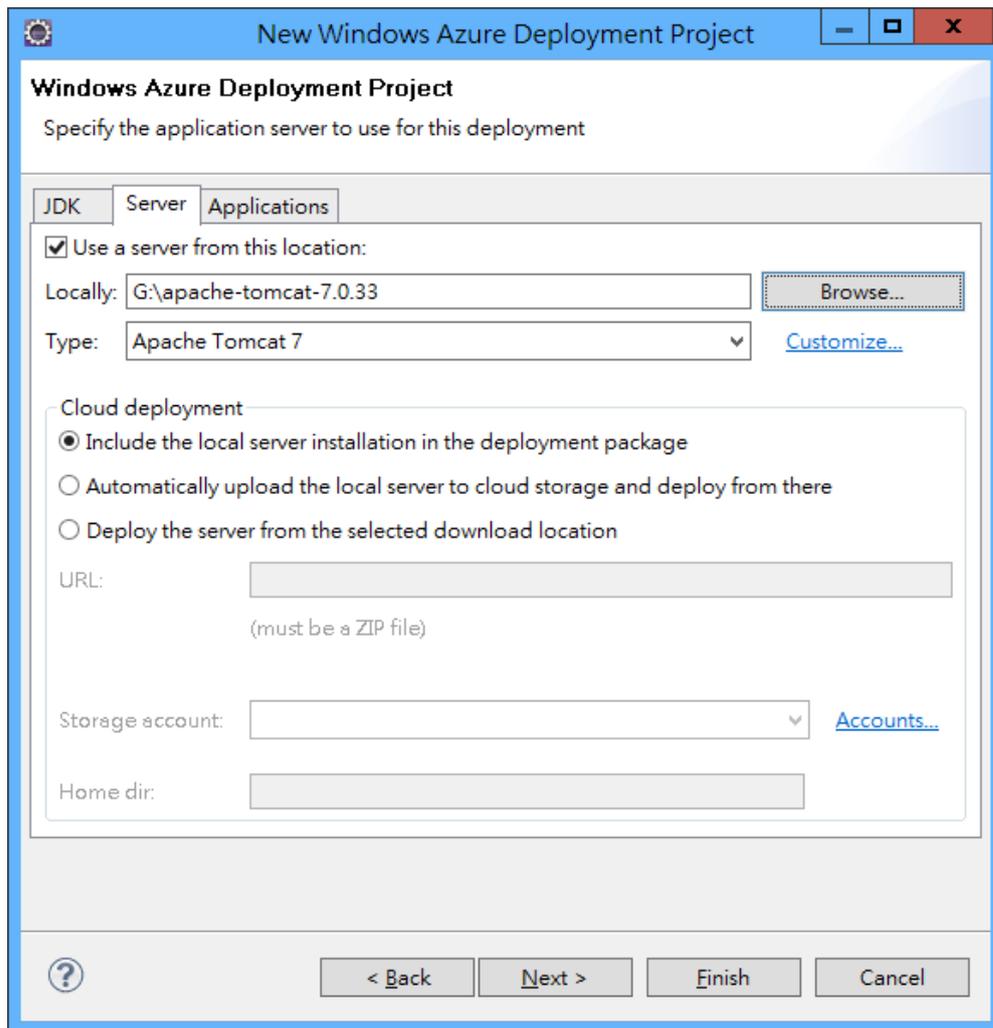
圖 12.12b 設定要包含在 Windows Azure 發行專案內的 Tomcat 來源位置

Server 頁籤允許開發人員在封裝中包含 Tomcat 的服務檔案，筆者在實驗的時候，使用的是 Tomcat 網站下載的 Windows 64 位元 zip 壓縮檔，裡面已包含了必要的執行檔與資源，我們只要將 Directory 指向我們將 zip 壓縮檔解壓縮的路徑即可。另外，如果 Tomcat 版本不是預設的 7 的話，要記得在 Select 下拉選單中選擇正確的版本，否則在啟動 Compute Emulator 的測試以及發行到雲端環境時會造成伺服器啟動失敗的錯誤。

另外自 2.0 版工具開始，不論是 JDK 或是 Tomcat，都允許開發人員以分開部署的方式進行，在 JDK/Server 頁籤中，都包含了一個 Cloud Deployment 的設定：

圖 12.12c 利用雲端儲存部署 JDK/Tomcat 的選項

這三項設定的說明如下：

- Include the local server installation in the deployment package 表示將 JDK/Tomcat 的來源檔案一起封裝在部署套件中上傳，這個方法是相容於 v1.8 或更早期版本的項目，因為發行的檔案會變大且無法重覆使用，所以較不適合會進行持續發行的專案。
- Automatically upload the local server to cloud storage and deploy from there 表示會將本地端的 JDK/Tomcat 封裝成不同的套件上傳到 storage 內，然後由 storage 內的 JDK/Tomcat 進行部署，優點為不必上傳包含 JDK/Tomcat 的來源檔，時間會少一些，它的預設路徑為 storage 帳戶下的 /eclipsedeploy 容器。
- Deploy the server from the selected download location 表示當發行時會自動檢查 storage 的指定路徑內是否含有必要的 JDK/Tomcat 來源，若沒有的話會自動由給定的 URL 來下載，這會運用在由不同的 storage 帳戶提供 JDK/Tomcat 的來源，而應用程式封裝放在不同的 storage 帳戶的情境時，而且它允許自訂路徑。

開發人員可以選擇適合的作法來發行應用程式與 JDK/Tomcat 來源檔。

Application 頁籤則是要包含的應用程式封裝的來源位置，若需要部署其他套件或是檔案時，可利用這個功能來加入：

圖 12.12d 設定要包含在 Windows Azure 發行專案內的 Application 來源位置

Step 6. 按 Next，可以選擇進一步的功能設定，包含 Session、Caching 與
Remote Debugging 等功能：
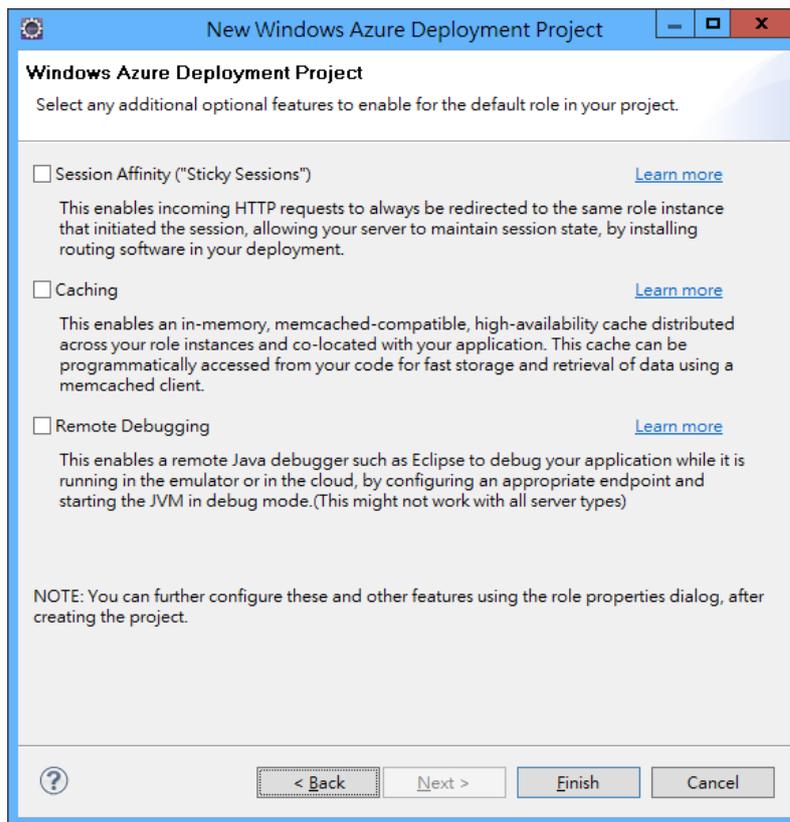
圖 12.13  設定 Windows Azure 的延伸功能

Step 7. 按下 Finish，即可產生這個發行專案。在這個專案內包含了 Worker
Role 所需要的 Tomcat 伺服器檔案以及 Java JRE 執行環境等，並且也包含了啟
動必要的指令碼，當然 Windows Azure 必要的服務定義檔與服務組態檔都少不
了。
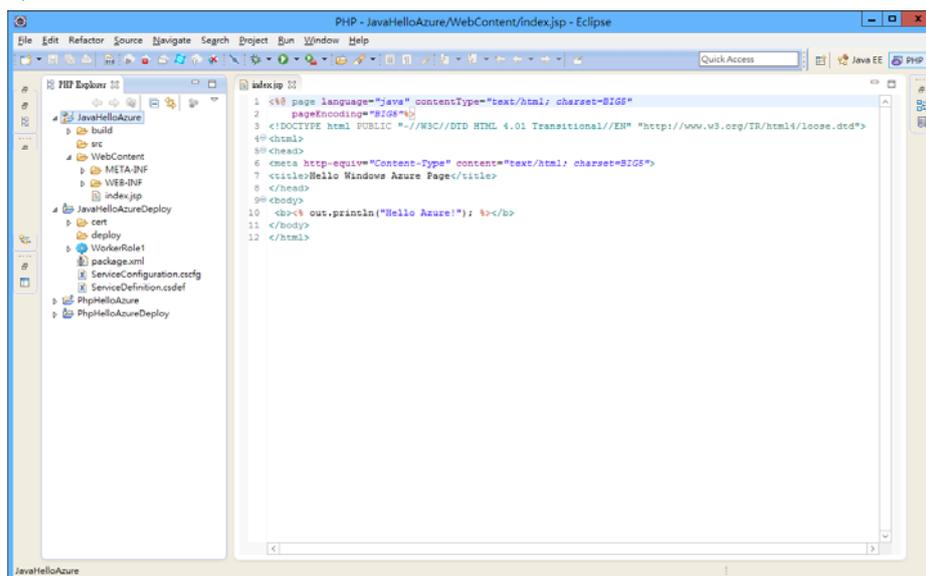


圖 12.14  完成時的 Windows Azure 專案內容

接下來，我們就可以按下工具列上的啟動模擬環境按鈕[3]，啟動 Compute Emulator 來瀏覽 JSP 網頁。
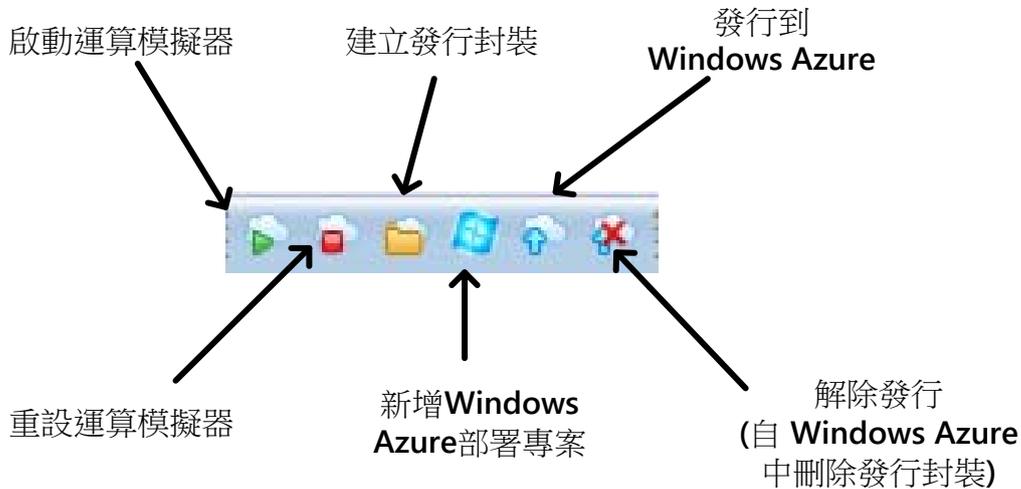


圖 12.15 Eclipse 中的 Windows Azure 工具列

經過一連串的啟動後，Compute Emulator 視窗會自動開啟，並顯示 Worker Role 已經啟動。
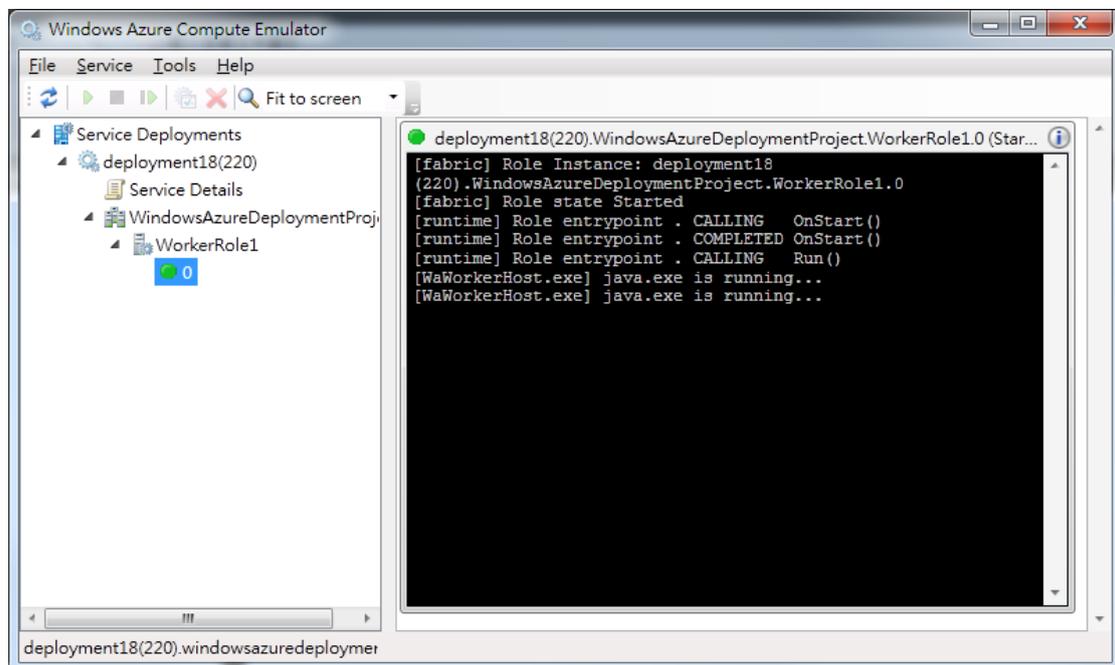


圖 12.16 啟動模擬器

---

[3] 這個工具列只有在 Windows Azure Tools for Eclipse 已安裝時才會出現。

此時再開啟瀏覽器並輸入 http://127.0.0.1:8080/JavaHelloAzure，即可看到 index.jsp 的執行結果：
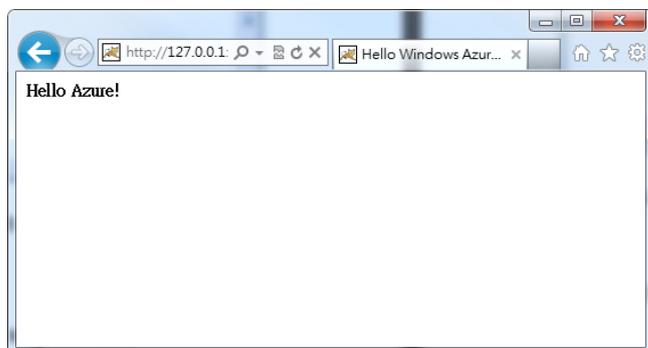


圖 12.17　測試結果

當看到執行結果時，表示 JSP 已成功執行。

除了工具的使用外，微軟也為 Java 提供了 SQL Server JDBC Driver 以及 Windows Azure Storage Library for Java[4]等開發環境支援，讓 Java 開發人員也能善用 Windows Azure 的環境來開發雲端應用程式。

若要發行到 Windows Azure 時，只要在工具列上按『Publish To Windows Azure Cloud』或是在部署專案上按右鍵，選擇 Windows Azure→Deploy to Windows Azure Cloud：
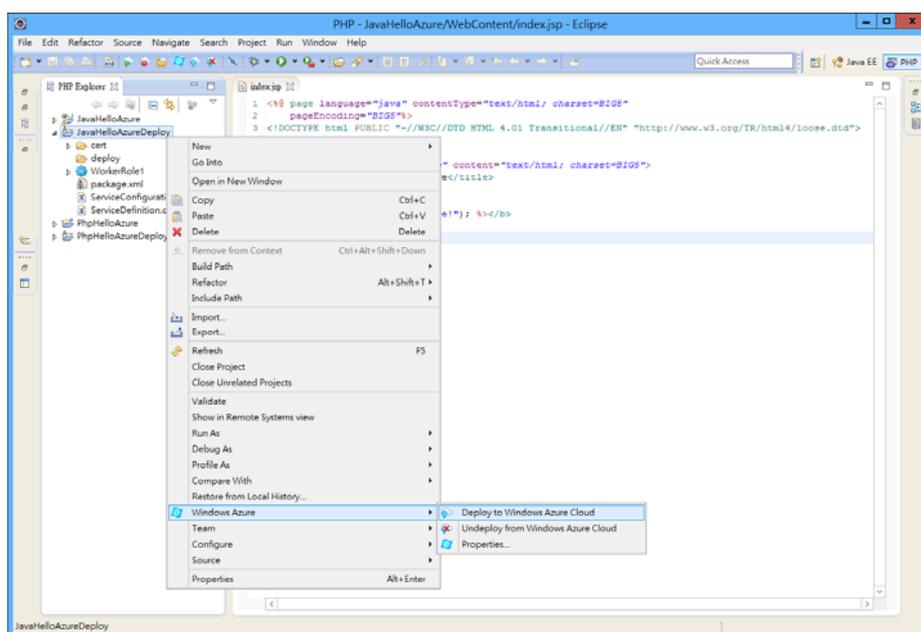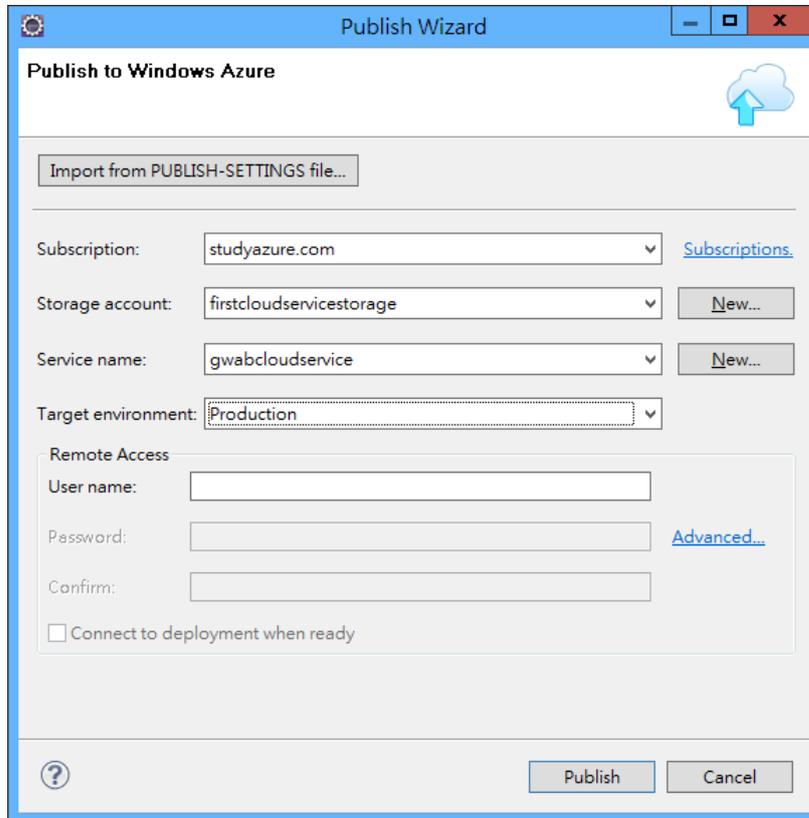


圖 12.18a　發行到 Windows Azure

---

4　https://github.com/WindowsAzure/azure-sdk-for-java

即會出現發行精靈，請設定好要發行的訂閱，儲存帳戶以及發行的組態等資訊後，按下 Publish 按鈕即可發行。



---

**NOTE**

若還沒有設定訂閱資訊時，可按下 *Import from PUBLISH-SETTINGS files*··· 按鈕，可連到 *Windows Azure* 管理介面並下載發行設定檔，其流程和使用 *Visual Studio* 下載發行設定檔相似。

---

**REFERENCES**

更多的 Java 相關 Windows Azure 平台開發的支援與資源，可參考 Windows Azure 官方網站的 Java Developer Center
(https://www.windowsazure.com/en-us/develop/java/)。