

IBM DB2 Alphablox



# 入門

8.4 版



IBM DB2 Alphablox



# 入門

8.4 版

**請注意：**

使用本資訊及其支援的產品之前，請先閱讀第 29 頁的『注意事項』中的資訊。

**第二版 (2006 年 3 月)**

除非新版中另有指示，否則本版適用於 IBM DB2 Alphablox for Linux, UNIX and Windows 8.4 版 (產品號碼 5724-L14) 及所有後續的版本與修訂版。

當您將資訊傳送給 IBM 時，您將授與 IBM 非專屬權利，IBM 得以其認為不損及您任何權利義務的適當方法使用或分送您所提供的資訊。

**Copyright © 1996 - 2006 Alphablox Corporation. All rights reserved.**

**© Copyright International Business Machines Corporation 1996, 2006. All rights reserved.**

---

## 目錄

### 第 1 章 指導教學：建置第一個應用程式 . . . 1

定義應用程式 . . . . .	1
存取資料 . . . . .	2
建立您的應用程式首頁 . . . . .	2
設定預設的首頁 . . . . .	3
建立您的第一個分析檢視畫面 . . . . .	3
第一個分析檢視頁面的結構 . . . . .	6
建立您的第二個分析檢視畫面 . . . . .	8
摘要 . . . . .	9

### 第 2 章 指導教學：建置第一個含有 Blox 元件的 Portlet . . . . . 11

安裝 Portlet 範例 . . . . .	11
執行 Portlet 範例 . . . . .	12
檢查包含 Blox 元件的 Portlet JSP 頁面結構 . . . . .	12
建立包含 Blox 元件的專屬 Portlet JSP 頁面 . . . . .	14
建立 Portlet 專案以使用 Blox 元件 . . . . .	15
使用 Rational Application Developer 配置 Portlet 專案 . . . . .	16
後續步驟 . . . . .	17
Portlet 開發要訣 . . . . .	18

### 第 3 章 指導教學：利用 Rational Developer 工具建置應用程式 . . . . . 19

準備開發環境 . . . . .	19
------------------	----

安裝 DB2 Alaphblox Toolkit . . . . .	19
在 WebSphere 整合測試環境中安裝 DB2 Alaphblox . . . . .	20
建立 WebSphere 伺服器實例 . . . . .	21
建立 WebSphere 5.1 伺服器替代變數 . . . . .	21
配置 WebSphere 5.1 伺服器實例 . . . . .	22
將來賓使用者新增至 DB2 Alaphblox 管理者群組 . . . . .	22
建立 DB2 Alaphblox 應用程式 . . . . .	23
建立含 DB2 Alaphblox 內容的 JSP 檔 . . . . .	23
存取 DB2 OLAP Server 及 Essbase 資料來源 . . . . .	24

### 第 4 章 指導教學：使用 DB2Cube Views 建置 DB2 Alaphblox 資料方塊 . . . 25

定義 DB2 關聯式資料來源 . . . . .	25
定義 Alaphblox Cube Server Adaptor 資料方塊 . . . . .	26
定義 DB2 Alaphblox 資料方塊 . . . . .	26
啓動 DB2 Alaphblox 資料方塊 . . . . .	27

注意事項 . . . . .	29
商標 . . . . .	30

索引 . . . . .	31
--------------	----



---

## 第 1 章 指導教學：建置第一個應用程式

此教學指導會教您有關 Blox 元件的重要基礎概念，並提供讓您遵循得以快速建置 DB2® Alphablox 應用程式的步驟。

若要建置 DB2 Alphablox 應用程式，您將會使用 JavaServer Pages (JSP) 技術，但是此教學指導並不要求您具備 JSP 的知識。此教學指導說明開始建置分析應用程式所需瞭解的 JSP 方面知識。若要瞭解其他關於 JSP 技術的資訊，有很多極好的書籍可供參考。

### 先決要件

在此教學指導中的步驟，假設您使用的是在 WebSphere® 應用程式伺服器上執行的 DB2 Alphablox。如果您使用 BEA WebLogic 或 Apache Tomcat 作為應用程式伺服器，則可能需要修改下列其中幾個步驟，才能適合您伺服器中的特殊差異。

同時，在開始此教學指導之前，您需要安裝 DB2 Alphablox 資料方塊 (qcc\_2003) 及其基礎關聯式資料來源 (qcc2003-rdb)。您可以在「DB2 Alphablox 安裝 CD」的 `sampledata/qcc/acs` 目錄中，找到此教學指導中關於配置 DB2 Alphablox 資料方塊及關聯式資料庫所需的範例資料庫檔案及指示 (在 `readme.txt` 檔案中)。

---

## 定義應用程式

若要利用 J2EE 開發方法來建立應用程式，就必須以包含應用程式描述子檔案 (`web.xml`) 的 `WEB-INF` 目錄來建立目錄結構。在 DB2 Alphablox 中建立此結構的最簡單方法，是利用「DB2 Alphablox 管理頁面」內的「應用程式」頁面來建立新應用程式。

若要建立應用程式及資料夾，請：

1. 遵循管理手冊中「應用程式定義」一節中所說明的步驟，在 DB2 Alphablox 中建立稱為 `MyApp` 的新應用程式。
  - 在名稱欄位中輸入 `MyApp`。
  - 在顯示名稱欄位中輸入 `My App` (含一個空格)，其定義會顯示在「應用程式」頁面清單中的應用程式標籤。
2. 按一下頁面左上方的**應用程式**標籤頂端。開啓可用應用程式的清單。
3. 按一下應用程式清單中的新建應用程式名稱 (`My App`)。因為您尚未建立任何檔案，所以檔案目錄是空的。
4. 瀏覽至您應用程式伺服器上的新應用程式資料夾 (若為 WebSphere，則資料夾位於 WebSphere 的 `installedApps` 目錄中)。請注意應用程式包括含有 `web.xml` 檔的 `WEB-INF` 目錄和 `tlds` 目錄。`web.xml` 檔案定義應用程式資訊。`tlds` 目錄包含 Blox 標示庫描述子檔案 (`blox.tld`)，其定義您將在本指導教學中建立分析檢視畫面時所使用的 Blox 標示。此目錄包含其他 Blox 標示庫的其他標示庫描述子 (TLD) 檔案 (本指導教學中不使用)，包括 `bloxform.tld`、`bloxlogic.tld`、`bloxreport.tld` 及 `bloxui.tld`。

您已建立了 DB2 Alphablox 應用程式架構，它現在具有新增的分析檢視畫面。

---

## 存取資料

在此指導教學中，您將使用在開始此教學指導之前所建立的 DB2 Alphablox 資料方塊資料來源。關聯式資料庫及 Alphablox Cube Server 必須正在執行中。同時，所使用的 DB2 Alphablox 資料方塊必須也在執行中。

若要驗證所需的資料來源可用並在運作中，請：

1. 開啓瀏覽器至「DB2 Alphablox 管理頁面」。
2. 按一下**管理**標籤。
3. 按一下**資料來源**。
4. 在「資料來源」頁面上，在頁面左側的功能表中尋找 qcc2003-rdbdata 來源。如果它不在該處，則您需要複查「DB2 Alphablox 安裝 CD」sampledata/qcc/acs 目錄中，readme.txt 檔案裡的步驟。
5. 按一下可用資料來源清單中的 qcc2003-rdb，然後按一下**測試已選取的資料來源**按鈕。如果資料庫正在執行中，而且已適當配置資料來源定義，則會顯示成功訊息。如果顯示錯誤訊息，則您需要檢查資料來源定義 (選取資料來源並按一下**編輯**按鈕)，或檢查是否已適當地配置資料庫，且使用者對其具有存取權。
6. 從資料來源清單中選取 qcc2003-acs 資料來源，然後按一下**測試已選取的資料來源**按鈕。如果目前已適當配置 Alphablox Cube Server 資料來源，則會顯示成功訊息。如果顯示錯誤訊息，請複查資料來源定義。

順利測試資料來源之後，您可以開始學習如何建置幾個簡單的 DB2 Alphablox 分析檢視畫面。

---

## 建立您的應用程式首頁

對於這項作業，您可使用基本網頁技巧來建立簡單的首頁。

爲了此指導教學的使用，您僅需要建立一個非常簡單的 Web 應用程式結構。在首頁上您需要建立兩個鏈結，以存取在此教學指導中所建立的兩個分析檢視畫面。

若要建立簡單的首頁，請：

1. 使用文字編輯器或您喜愛的開發工具，將其命名爲 index.html。爲了此指導教學的使用，您可以讓其結構保持非常簡單。
2. 編輯檔案以指定應用程式標題 (例如 My DB2 Alphablox Application) 及兩個鏈結可連接到您將建立的分析檢視畫面。

在 index.html 檔案中，複製並貼上 (或是鍵入) 下列程式碼。

```
<html>
<head>
<title>My DB2 Alphablox Application</title>
</head>
<body>
<h2>My DB2 Alphablox Application</h2>
<p>
<a href="PresentBloxView.jsp">Simple PresentBlox View</a>
</p>
<p>
```



```
<a href="ChartView.jsp">Customized Chart View</a>
</p>
</body>
</html>
```

3. 將此檔案儲存至定義應用程式時所建立的 MyApp 目錄。
4. 開啟 Web 瀏覽器至 DB2 Alphablox 首頁。根據預設值，瀏覽器會顯示**應用程式**標籤。
5. 按一下 **My App** 鏈結，以驗證該檔案在目錄清單中。按一下 **index.html** 鏈結，以檢視首頁。如果您的 DB2 Alphablox 安裝在 IBM WebSphere Application Server 或 BEA WebLogic 伺服器上，則必須先啟用目錄清單，然後才能執行此步驟。至於在 Apache Tomcat 上的 DB2 Alphablox 安裝，則不需要其他的步驟。如果您看到 HTTP 403 錯誤 (依預設不容許目錄清單)，您可以修改伺服器以容許目錄清單，或只要將檔案名稱 myapp.html 附加至 URL 尾端 (例如，http://localhost:9080/MyApp/myapp.html)。使用完整的鏈結位址時應該顯示新頁面。

現在，您已經有一個含有連到兩個分析檢視畫面首頁的鏈結 (您下一步將建立分析檢視畫面)。

---

## 設定預設的首頁

在先前步驟中，直接存取首頁的方法有，按一下 MyApp 應用程式目錄中的檔案名稱鏈結，或在 URL 中指定首頁檔案 index.html。在此步驟中，您會指定應用程式的預設首頁。

若要設定預設應用程式首頁：

1. 開啟瀏覽器至 DB2 Alphablox 首頁。
2. 按一下**管理**標籤。
3. 按一下**應用程式**鏈結。
4. 在應用程式清單中選取 **MyApp**，並按一下清單下面的**編輯**按鈕
5. 在 MyApp 的「編輯應用程式」頁面上，在**首頁 URL** 欄位中輸入 index.html。
6. 按一下**儲存**按鈕，以儲存您所進行的變更。
7. 按一下「應用程式」主標籤 (左上方的資料夾標籤)，回到「應用程式」頁面。
8. 按一下 **My App** 應用程式名稱。應用程式現在會直接開啓已定義的首頁。

現在已定義了首頁，即使使用者無法在他們的 URL 中指定首頁，伺服器也會知道從何處啓動應用程式，並會引導應用程式在此頁面開啓。因此，當使用者在瀏覽器網址列中輸入 http://yourServerName/MyApp/ 時，應用程式將會自動開啓已定義的首頁，即 index.html。

在即將進行的主題中，您將使用由「DB2 Alphablox 查詢建置器」應用程式所產生的簡單 Blox 標示來開始建立 DB2 Alphablox 分析檢視畫面。

---

## 建立您的第一個分析檢視畫面

在這項作業中，您將會建立 DB2 Alphablox 分析檢視畫面，以顯示合併成 PresentBlox 元件的網格及圖表。

**先決要件：**請閱讀 *Developer's Guide* 中的簡介章節，以瞭解 DB2 Alphablox 及其元件。這可讓您對 DB2 Alphablox 的功能，以及在利用 JSP 技術建置分析應用程式時如何使用 Blox 自訂標示庫，有基本的瞭解。

針對您將建立的第一個分析檢視畫面，將會使用「DB2 Alphablox 查詢建置器」應用程式產生為 PresentBlox 元件定義的 JSP 自訂標示。PresentBlox 元件是用於呈現資料方面最常使用的 Blox 元件。PresentBlox 元件可以包括網格及圖表檢視、資料佈置畫面、工具列，以及其他巢狀元件。網格及圖表檢視代表相同資料的同步替代檢視。

若要使用「查詢建置器」來建立第一個分析檢視畫面，請：

1. 開啟瀏覽器至「DB2 Alphablox 管理頁面」。根據預設值，會看見「應用程式」標籤。
2. 按一下「DB2 Alphablox 查詢建置器」，以開啟應用程式。
3. 按一下**連線設定**按鈕。即會出現「資料庫連線」視窗。
4. 選取 qcc2003-acs 資料來源，然後按**連接**按鈕。即會關閉視窗，您會看到「查詢建置器」頁面的「資料庫狀態」區段中會顯示「已連接」。

**註：**您無需填寫**綱目、型錄、使用者名稱及密碼**欄位，因為會自動從您選取之資料來源的資料來源定義中擷取這些值。此外，請勿選取**執行預設查詢**勾選框。

5. 在頁面上尋找位於「狀態」區段下面的「查詢」區段。請注意下列特性：
  - 針對此教學指導，您使用的是 DB2 Alphablox 資料方塊 (qcc2003-acs)。在「查詢」區段的標頭下，您會看到可用 (目前在執行中) 資料方塊的清單。應會列出 qcc2003-acs 資料方塊。
  - 與此資料來源相關聯的預設查詢字串顯示在文字視窗中。因為資料來源是 DB2 Alphablox 資料方塊，所以會將依字母順序列出的第一個資料方塊，新增至查詢陳述式 (例如，"select from [datasourceName]")。

**註：**啟用 DB2 Alphablox Cube Server 中的連線儲存區功能之後，您將無法使用**取得預設查詢**按鈕來取得 DB2 Alphablox 資料方塊的預設查詢。

- - 當資料顯示在頁面底部的 PresentBlox 時，可以使用標準使用者介面來交換軸、收合或展開，以及在軸之間移動維度等。
  - 若要檢視查詢的結果，請按一下「執行查詢」按鈕。即會在頁面底部的 PresentBlox 顯示結果集。
6. 在預設查詢 (例如，select from [qcc2003-acs]) 的方括弧之間鍵入 qcc2003-acs，然後按**執行查詢**按鈕。

PresentBlox 區段會根據查詢顯示預設的檢視畫面。

7. 使用 PresentBlox 使用者介面建立新的檢視畫面。

例如，您可以將 Products 維度拖曳到「資料佈置」畫面左側的「橫列」軸。然後，您可以將 Time (Calendar) 維度拖曳到「直欄」軸。之後您可能要將 Measures 維度拖曳到「頁」軸，以讓 Measures 功能表顯示在顯示區上方的「頁面」畫面中，該畫面會顯示頁面過濾器。選取其他兩個維度，將它們新增至「頁」軸，以讓三個頁面過濾器顯示在顯示畫面的上方。如果您是使用 DB2 Alphablox 使用者介面的新手，請從 PresentBlox 工具列選取**說明**，以瞭解如何使用介面。

8. 建立完新的檢視畫面之後，請注意「查詢」區段中顯示的查詢陳述式。如果查詢不是動態產生的 (未勾選「自動更新查詢」選項)，您可以按一下**取得現行查詢**按鈕取得現行查詢。文字方塊會顯示開發現行資料檢視畫面所需的查詢陳述式。您可以從要用來開發 DB2 Alphablo 應用程式的「查詢」欄位中複製及貼上查詢陳述式。
9. 按一下**產生 Blox 標示**按鈕。即會開啟對話框，顯示重新產生 PresentBlox 佈置及結果集的標示 (及巢狀標示)。
10. 在文字編輯器中開啟新檔，並將下列文字新增至檔案，然後將檔案儲存為 PresentBloxView.jsp。下一主題將說明此頁面上之程式碼的許多重要元素。

```
<%@ taglib uri="bloxtld" prefix="blox" %>
//從下面產生的 Blox 標示中將標示資訊複製及
//貼在此行的正下方，並從查詢陳述式移除所有
//換行。您還可以在新增程式碼之後，從
//「DB2 Alphablo 查詢建置器」移除此註解。
<html>
<head>
<blox:header/>
</head>
<body>
<h2>Simple PresentBlox View</h2>
<p>
<blox:display bloxRef="MyPresentBloxView"/>
</p>
</body>
</html>
```

11. 將儲存的檔案 (PresentBloxView.jsp) 複製到您伺服器上的 MyApp 目錄，然後藉由從 MyApp 首頁按一下此頁面的鏈結 (Simple PresentBlox View) 來測試新的分析檢視畫面。或者，您可以直接在瀏覽器網址中輸入 URL (例如，<http://localhost:9080/MyApp/PresentBloxView.jsp>)。您將看到的檢視畫面應與在「DB2 Alphablo 查詢建置器」中處理的 PresentBlox 類似。

以下所示為完整版本之「簡式 PresentBlox 檢視畫面」頁面 (PresentBloxView.jsp) 的程式碼範例：

```
<%@ taglib uri="bloxtld" prefix="blox" %>
<blox:present
id="queryBuilder4_present"
height="500"
visible="false"
width="100%">
<blox:grid/>
<blox:chart/>
<blox:page/>
<blox:data
dataSourceName="qcc2003-acsc"
onErrorClearResultSet="true"
query="SELECT DISTINCT({[qcc_2003].[Time (Calendar)].[All Time (Calendar)], [qcc_2003].[Time (Calendar)].[All Time (Calendar)].[2000], [qcc_2003].[Time (Calendar)].[All Time (Calendar)].[2001], [qcc_2003].[Time (Calendar)].[All Time (Calendar)].[2002], [qcc_2003].[Time (Calendar)].[All Time (Calendar)].[2003]}) ON AXIS(0), DISTINCT( {[qcc_2003].[Products].[All Products], [qcc_2003].[Products].[All Products].[100 Truffles], [qcc_2003].[Products].[All Products].[200 Chocolate Blocks], [qcc_2003].[Products].[All Products].[300 Chocolate Nuts], [qcc_2003].[Products].[All Products].[400 Specialties]}) ON AXIS(1) FROM [qcc_2003] WHERE ([qcc_2003].[Measures].[Sales], [qcc_2003].[Time (Fiscal)].[All Time (Fiscal)], [qcc_2003].[Date Opened].[All Date Opened], [qcc_2003].[Has Nuts].[All Has Nuts],
```

```

[qcc_2003].[Chocolate Type].[All Chocolate Type],
[qcc_2003].[Ounces Per Package].[All Ounces Per Package],
[qcc_2003].[Pieces Per Package].[All Pieces Per Package],
[qcc_2003].[Date Introduced].[All Date Introduced],
[qcc_2003].[Seasonal].[All Seasonal],
[qcc_2003].[Scenario].[Actual],
[qcc_2003].[Locations].[All Locations]]"
selectableSlicerDimensions="[qcc_2003].[Measures],
[qcc_2003].[Locations],[qcc_2003].[Chocolate Type]"
useAliases="true"/>
<blox:toolbar/>
<blox:dataLayout/>
<bloxui:calculationEditor />
</blox:present> <html>
<head>
<blox:header/>
</head>
<body>
<h2>Simple PresentBlox View</h2>
<blox:display bloxRef="queryBuilder4_present"/>
</p>
</body>
</html>

```

在上述的程式碼範例中，為便於閱讀，查詢陳述式中包含換行。若要正確執行此頁面，需要將屬性值 (如上述的 `query` 及 `selectableSlicerDimensions` 屬性) 中的換行移除。在「DB2 Alphablox 查詢建置器」所產生的程式碼中，所包括的許多標示是不必要的。不過，看見此處所有的標示可讓您瞭解 Blox 標示作為母項標示及巢狀標示是如何運作的。此外，您還可以查看在「查詢建置器」的 PresentBlox 中進行操作所新增之標示的標示屬性用法。若要學習使用 Blox 標示、標示屬性，以及如何開發 DB2 Alphablox 應用程式的詳細資料，請參閱「DB2 Alphablox 資訊中心」。

---

## 第一個分析檢視頁面的結構

此主題彙總了您在前一個主題中所建立之第一個分析檢視畫面 (PresentBloxView.jsp) 的結構。

以下是在前一主題所建立之頁面中某些重點部分的簡要概觀。如需對程式碼的更完整瞭解，您必須閱讀「DB2 Alphablox 資訊中心」中感興趣的主題。

PresentBloxView.jsp 頁面以下行開頭：

```
<%@ taglib uri="bloxtld" prefix="blox" %>
```

此命令行是 JSP taglib 指引，通知伺服器您打算使用 Blox 標示庫。`uri` 是標示庫描述子檔案的指標。`prefix` 值 (定義為 `blox`) 會告知伺服器尋找此頁面上以 `blox` 開頭的所有標示，然後依照標示庫描述子檔案中的定義，使用 Blox 標示庫來處理內容。

在 taglib 指引的下面，PresentBlox 元件由下列標示及其標示屬性指定：

```

<blox:present
id="queryBuilder4_present"
height="500"
visible="false"
width="100%">
<blox:grid/>
<blox:chart/>
<blox:page/>
<blox:data
dataSourceName="qcc2003-acb"
onErrorClearResultSet="true"

```

```

query="SELECT DISTINCT({[qcc_2003].[Time (Calendar)].[All Time
(Calendar)], [qcc_2003].[Time (Calendar)].[All Time (Calendar)].[2000],
[qcc_2003].[Time (Calendar)].[All Time (Calendar)].[2001],
[qcc_2003].[Time (Calendar)].[All Time (Calendar)].[2002],
[qcc_2003].[Time (Calendar)].[All Time (Calendar)].[2003]})
ON AXIS(0), DISTINCT( {[qcc_2003].[Products].[All Products],
[qcc_2003].[Products].[All Products].[100 Truffles],
[qcc_2003].[Products].[All Products].[200 Chocolate Blocks],
[qcc_2003].[Products].[All Products].[300 Chocolate Nuts],
[qcc_2003].[Products].[All Products].[400 Specialties]}) )
ON AXIS(1) FROM [qcc_2003] WHERE ([qcc_2003].[Measures].[Sales],
[qcc_2003].[Time (Fiscal)].[All Time (Fiscal)],
[qcc_2003].[Date Opened].[All Date Opened],
[qcc_2003].[Has Nuts].[All Has Nuts],
[qcc_2003].[Chocolate Type].[All Chocolate Type],
[qcc_2003].[Ounces Per Package].[All Ounces Per Package],
[qcc_2003].[Pieces Per Package].[All Pieces Per Package],
[qcc_2003].[Date Introduced].[All Date Introduced],
[qcc_2003].[Seasonal].[All Seasonal],
[qcc_2003].[Scenario].[Actual],
[qcc_2003].[Locations].[All Locations])"
selectableSlicerDimensions="[qcc_2003].[Measures],
[qcc_2003].[Locations],[qcc_2003].[Chocolate Type]"
useAliases="true"/>
<blox:toolbar/>
<blox:dataLayout/>
<bloxui:calculationEditor />
</blox:present>

```

`<blox:present>` 標示指定您想在這裡顯示 `PresentBlox`，且 `id` 屬性值為 `queryBuilder4_present`。為了便於 Scripting，`id` 屬性可讓您識別此特定 Blox。請注意，`<blox:present>` 標示包括了設為 `false` 的 `visible` 屬性。如果設為 `false`，則一直要等到編譯器在頁面中遇到 `<blox:display>` 標示時，才會呈現 `PresentBlox`。

每個使用標示定義的 Blox 都有許多可用的屬性，但是除非您需要指定與預設值不同的屬性值，否則無需將這些屬性值併入標示中。在此範例中，`height` 值為 500，而 `width` 值為 100%。高度及寬度可以按照像素或百分比進行定義。

將 `dataSourceName` 屬性定義為 `qc2003-acs`，就會併入巢狀 `DataBlox`。如果未指定資料來源，則顯示頁面時會看到「無可用資料」訊息。`query` 屬性併入了由「DB2 Alphablox 查詢建置器」所產生的 MDX 查詢陳述式。當您瞭解與 DB2 Alphablox Cube Server 搭配使用的 MDX 查詢語言後，您就可以簡化查詢，讓它縮短。不過，這裡的 `query` 陳述式是完整的，並將產生您在「查詢建置器」應用程式中所建立的檢視畫面。

頁面的 `<head>` 區段中有一個特殊的 Blox 標示，用於在呈現頁面之前將重要的程式碼新增至頁面中：

```
<blox:header/>
```

DB2 Alphablox 使用此標示來自動將所需的 HTML、JavaScript™ 及 CSS 程式碼新增至頁面的標頭區段。當伺服器呈現頁面時，此標示會將 CSS 鏈結新增至已定義的 HTML 主題中，並產生用於阻止快取的 `meta` 標示。請記得在每一個使用 Blox 元件的 JSP 頁面輸入此標示。如果不加入此標示，頁面將無法正常呈現。

該頁面的 `<body>` 區段中包含下列行：

```
<blox:display bloxRef="MyPresentBloxView"/>
```

此 Blox 標示會導致上述指定的 `PresentBlox` 在這裡呈現。這樣可讓您在同一位置對 Blox 元件或其巢狀元件進行變更，您還可以在此位置讓其他 Blox 標示及 JSP Scriptlet 控制



Blox 元件行爲。在複雜的 JSP 檔中，當您熟悉 DB2 Alphablox 後，可以選擇讓 DataBlox 元件與 PresentBlox 標示分開，並於 Blox 元件在 Web 瀏覽器中呈現之前，新增修改 Blox 元件及其行爲的 JSP Scriptlet。

總而言之，此頁面上所使用的最重要 Blox 標示包括：`<blox:header/>`、`<blox:present>` 及 `<blox:display>`。這三個標示及巢狀標示會指定分析檢視畫面，無需在頁面上使用任何 Java™ 程式碼。呈現邏輯的複雜性由這些 Blox 標示來管理。您可以透過新增巢狀 Blox 及修改屬性值，根據您的事業需求自訂檢視畫面。

若爲更複雜的檢視畫面及應用程式，可能需要其他的 JavaBeans 元件及 Java 程式碼。

---

## 建立您的第二個分析檢視畫面

在這項作業中，您將建立一個分析檢視畫面，它會根據此教學指導中第一個分析檢視畫面的程式碼，只顯示包含頁面過濾器的圖表。

許多應用程式資料只使用格線或圖表來呈現。雖然 PresentBlox 元件可以將 GridBlox 及 ChartBlox 合併爲巢狀元件，同時顯示在 PresentBlox 元件中，但您可以停用或隱藏不想顯示的巢狀元件。使用 PresentBlox 元件來建立包含頁面過濾器的圖表檢視比較簡單，您可以藉由修改某些巢狀 Blox 標示上的標示屬性來完成此動作。

在這項作業中，您將重覆使用第一個分析檢視畫面的 PresentBlox 標示，以建立第二個自訂的分析檢視畫面。

若要建立顯示包含頁面過濾器之圖表的分析檢視畫面，請：

1. 在編輯器中開啓先前建立的 PresentBloxView.jsp 檔案，並將其儲存爲 CustomizedChartBlox.jsp。
2. 在以下列出之 PresentBlox 標示的巢狀元件中，將 visible 屬性新增至每個標示，並將值設爲 false：新增此屬性會導致不顯示這些巢狀元件，只顯示包含頁面過濾器列的圖表檢視。

```
<blox:grid visible="false"/>
<blox:chart visible="false"/>
<blox:toolbar visible="false"/>
<blox:dataLayout visible="false"/>
```

3. 修改 PageBlox 標示 (`<blox:page>`)，新增 labelPlacement 屬性並將值設爲 top。根據預設值，頁面過濾器的標籤會顯示在功能表的左側。新增此屬性並將其設爲 top 會置換預設行爲，並會將標籤置於頁面過濾器的上方。

```
<blox:page labelPlacement="top" />
```

4. 將 `<blox:present>` 標示中的 id 屬性名稱改成下列值：chartview。每一個顯示或呈現至頁面的 Blox 元件都需要一個唯一的 ID，以避免發生錯誤。

5. 儲存變更，並將檔案新增至伺服器上的 MyApp 目錄：

6. 開啓 Web 瀏覽器，並從首頁按一下鏈結或使用 URL (例如，<http://localhost:9080/MyApp/CustomizedChartView.jsp>)直接進行存取來測試新頁面

您已利用「DB2 Alphablox 查詢建置器」應用程式來協助入門，建立了兩個分析檢視畫面。同時，您瞭解到藉由在現有標示上修改或新增屬性，可以控制 Blox 及其巢狀元件的外觀及行爲。

---

## 摘要

完成教學指導的所有作業之後，您學會了如何使用「DB2 Alphablox 管理頁面」、「DB2 Alphablox 查詢建置器」及 Blox 標示來建置基本的 DB2 Alphablox 應用程式。如果您之前已經查看 *Developer's Reference*，會知道可以使用許多內容及方法來在 JSP 頁面上定義及操作 Blox。那麼，接下來您應該執行哪些步驟呢？

如果您想在 DB2 Alphablox 檢視畫面中查看公司資料，現在您知道如何從頭建立應用程式，以及如何將分析檢視畫面新增至應用程式。如果您想要立即查看某些資料，則只要對 MyApp 應用程式進行幾個簡單的修改即可。若不要使用 qcc2003-acs 資料來源來顯示您公司資料庫的資料，您可以建立指向公司資料庫的資料來源、修改 dataSourceName 屬性以指向新定義的資料來源，然後新增適當的 query 屬性。使用「DB2 Alphablox 查詢建置器」，您可以經常快速地建立可用的查詢陳述式及 Blox 標示。如需建立資料來源的詳細資料，請參閱 *管理手冊*。若要學習如何建立適當的查詢及如何建置 DB2 Alphablox 應用程式，請參閱 *Developer's Guide* 中的 Retrieving Data 及 *Developer's Reference* 中的 DataBlox 一節。





---

## 第 2 章 指導教學：建置第一個含有 Blox 元件的 Portlet

在此教學指導中，您將學到如何將 Blox 元件新增至 Portlet 中。您將執行下列作業：

1. 安裝預建的 Portlet 範例。在此步驟，您可以很快看到 Blox 元件是如何新增至 Portlet JSP 頁面，並且在入口網站網頁中顯示的方式。
2. 撰寫一個含有 GridBlox 的專屬 JSP 頁面。

此教學指導中的作業不提供關於一般 Portlet 開發的詳細資料。此教學指導著重於 DB2 Alphablox 特定作業上，並假設您稍微了解入口網站環境及 Portlet 開發的一般概念。

### 先決要件

- 必須在 WebSphere Portal 5.1 版伺服器下安裝 DB2 Alphablox。如需關於安裝的詳細資料，請參閱安裝手冊。
- 必須要啟動您的 WebSphere Portal 伺服器。
- 必須要有對 WebSphere Portal 伺服器的管理存取權。
- 您必須熟悉 WebSphere Portal 中的管理功能及使用者介面。
- 您應該對 Java 及 JSP 有基本瞭解。
- 您應該已安裝 JSP 編輯器。

雖然可以使用任何 JSP 編輯器或甚至文字編輯器來進行此教學指導，當您開發自己的 Portlet 時，請使用 WebSphere Portal 所建議的開發工具，例如，Rational® Application Developer。

此教學指導使用以 DB2 Alphablox 所安裝的預先定義資料來源。您可使用此資料來源來快速開發基本的應用程式。您不需要擔心配置此教學指導的自訂資料來源。

---

### 安裝 Portlet 範例

學習如何將 Blox 新增至 Portlet 的最佳方法，是安裝 DB2 Alphablox 中提供的 Portlet 範例，並將其載入到入口網站網頁。這可讓您檢查 JSP 程式碼的基本結構，讓其與您入口網站上的輸出相符。

安裝 DB2 Alphablox 中所提供的 Portlet 範例：

1. 開啓瀏覽器，以管理使用者身分登入您的入口網站（URL 格式為 `http://<yourPortalServer>:<port>/wps/portal`）。
2. 按一下**管理**按鈕。
3. 在「Portlet 管理」區段下，按一下 **Web 模組**。右邊即會顯示「管理 Web 模組」頁面。
4. 按一下**安裝**。系統會提示您輸入要安裝的「Web 模組」。
5. 按一下**瀏覽**按鈕，瀏覽至您 DB2 Alphablox 安裝目錄下的 `installableApps` 目錄。
6. 選擇 `AlphabloxSamplePortlets.war`，再按**下一步**。Portlet 應用程式表格中會顯示 DB2 Alphablox Portlet 範例應用程式，其中含有名稱為「DB2 Alphablox JSP 頁面 Portlet 範例」的 Portlet。

7. 按一下**完成**。

現在已安裝「DB2 Alphablox Portlet 範例」應用程式及內含的 Portlet。檢查您 WebSphere Portal 安裝下的 installedApps 目錄。新建目錄的名稱以 DB2 Alpha 開頭，以動態產生的 Portlet ID 結尾，形式為 \_PA\_x\_x\_xx.ear。

## 執行 Portlet 範例

若要在您的入口網站網頁上執行 Portlet 範例：

1. 前往入口網站網頁。
2. 建立或編輯現有的頁面。您可以建立新的頁面來測試此 Portlet 範例，或是按一下現有的頁面以編輯該頁面。
3. 按一下入口網站佈置頁面上的其中一個**新增 Portlet** 按鈕。
4. 在「搜尋」欄位中鍵入 DB2，然後按**搜尋**。即會出現 **DB2 Alphablox JSP 頁面 Portlet 範例** 勾選框。
5. 選取該勾選框，再按一下**確定**。
6. 按一下**完成**。

重新整理入口網站網頁時，在入口網站網頁中會出現一個 PresentBlox。此 PresentBlox 包含：

- 上方的功能表列
- 功能表列下方的兩個工具列面板
- 左邊的資料佈置面板 - 可讓您在不同的軸之間移動維度
- 格線 - 以列表格式顯示資料
- 右邊的三維長條圖

此 present.jsp 檔案是範例中內含之 BloxJSPPagePortlet servlet 中所指定要載入的預設頁面。此 servlet 範例的原始程式碼可在 WEB-INF/src/ 目錄中找到。

您已成功安裝 Portlet 範例，並將含 Blox 的 Portlet 新增至入口網站網頁。下一個作業是檢查此 JSP 頁面中的程式碼結構。

## 檢查包含 Blox 元件的 Portlet JSP 頁面結構

在這項作業中，您會檢視包含 Blox 之 JSP 頁面的程式碼結構。所有的 JSP 頁面都必須有相同的主要元素。

若要開啓 JSP 檔案：

1. 瀏覽至 WebSphere Portal 安裝下的 installedApps/ 目錄，並找到以 DB2 Alpha 開頭的新建應用程式資料夾。
2. 瀏覽至 PA\_x\_x\_xx.war/jsp/html/ 目錄。
3. 在 JSP 或 Java 編輯器中開啓 present.jsp。
4. 查驗下列程式碼並注意主要元素：

```
<%@ page contentType="text/html"%>
<%@ taglib uri="bloxtld" prefix="blox"%>
<%@ taglib uri="/WEB-INF/tld/portlet.tld" prefix="portletAPI" %>
<portletAPI:init/>
<%
String bloxName = portletResponse.encodeNamespace("presentBlox");
```

```

%>
<head>
<blox:header />
</head>
<blox:present id="presentBlox" bloxName="<%= bloxName %>" width="800">
<blox:data dataSourceName="canned" />
</blox:present>

```

此程式碼區段包含六個主要元素：

1. 第一行告知瀏覽器輸出為 HTML：

```
<%@ page contentType="text/html"%>
```

2. 下一組程式碼指定此頁面中使用的兩個 JSP 標示庫：

```

<%@ taglib uri="bloxtld" prefix="blox"%>
<%@ taglib uri="/WEB-INF/tld/portlet.tld" prefix="portletAPI" %>

```

uri 是標示庫描述子檔案所在目錄位置的指標。prefix 值 (定義為 blox 及 portletAPI) 會告知伺服器：

- 尋找此頁面上以 blox 開頭的所有標示，然後使用標示庫描述子檔案所定義的「Blox 標示庫」來處理內容。
- 尋找此頁面上以 portletAPI 開頭的所有標示，然後使用標示庫描述子檔案所定義的「Portlet 標示庫」來處理內容。

3. 接著，新增 Portlet 起始設定標示：

```
<portletAPI:init/>
```

此標示可提供對 PortletRequest、PortletResponse 及 PortletConfig 物件的存取權。利用 PortletResponse，透過呼叫 encodeNamespace() 方法，來確保您的 Blox 名稱與同一頁面上所執行其他 Portlet 的不同物件沒有衝突。

4. 下一個標示會針對新增至該頁面的 Blox 名稱空間進行編碼：

```

<%
String bloxName = portletResponse.encodeNamespace("presentBlox");
%>

```

這可讓您在稍後建立一個 Blox，並將此唯一名稱指定給它。

5. 下一個程式碼區塊新增 Blox 呈現及伺服器用戶端通訊時所需的 Blox 標頭標示：

```

<head>
<blox:header />
</head>

```

DB2 Alphablox 使用此標示，自動將所需的 HTML、JavaScript 及 CSS 程式碼新增至頁面的標頭區段。當伺服器呈現頁面時，結果此標示會將 CSS 鏈結加入已定義的 HTML 主題中，並產生用於阻止快取的 meta 標示。此標示必須新增至每一個包含 Blox 元件的 JSP 頁面，或無法正常呈現的元件。

6. 使用「Blox 標示庫」中提供的標示新增 PresentBlox：

```

<blox:present id="presentBlox" bloxName="<%= bloxName %>" width="800">
<blox:data dataSourceName="canned" />
</blox:present>

```

- 進行名稱空間編碼之後，此程式碼會新增一個 id 為 presentBlox 且 bloxName 為 xx\_x\_x\_xxx\_presentBlox 的 PresentBlox。
- 此 Blox 的寬度為 800 像素，高度為 400 像素 (預設值)。
- 將 dataSourceName 標示屬性設為 canned，就會併入巢狀 DataBlox。

PresentBlox 的 id 標示屬性為必要屬性。其指定要在 JSP 頁面上使用的 Java Scripting 名稱。bloxName 屬性指定伺服器上的物件名稱。已編碼的 bloxName 可確保實例在伺服器上是唯一的。

**註：**如果未指定資料來源，則一般會在格線中看到**無可用資料**訊息。會在安裝期間預先定義預錄的資料來源，並且包含少量的範例資料。不需要安裝和配置實際的外部資料庫，且可用於疑難排解及學習。我們不需要指定查詢，所以程式碼中沒有查詢屬性。

在入口網站網頁上會使用其他 Portlet 來顯示此 JSP 頁面，所以此頁面沒有最外層的 <html> 標示。不需要其他 <html> 或 <body> 標示。

現在您已檢查過 JSP 結構，並學會將基本的程式碼加入 Portlet JSP 頁面，因此在下一個作業中，您將建立一個含有不同 Blox 的新 JSP 頁面，並指定一些常用的 Blox 屬性，讓頁面更適合您的入口網站網頁。

---

## 建立包含 Blox 元件的專屬 Portlet JSP 頁面

在這項作業中，您將建立包含 GridBlox 的新 JSP 頁面，並設定其部分內容。目的是讓您在建立適合一般入口網站網頁之 GridBlox 的同時，熟悉一般的 Blox 標示建構。

GridBlox 的預設大小為 400x400 像素。另外，還附有功能表列及工具列。您將設定一些常用的 GridBlox 內容，讓格線的高度僅佔 100 像素，且關閉其功能表列及工具列。完成這些作業的方法是設定下列 GridBlox 內容：

- height：設為 100 像素
- menubarVisible：設為 false
- toolbarVisible：設為 false

請確定您符合第 11 頁的第 2 章，『指導教學：建置第一個含有 Blox 元件的 Portlet』中指定的需求，且已安裝了第 11 頁的『安裝 Portlet 範例』一節中所述的 Portlet 範例應用程式。

遵循下面的步驟：

1. 在顯示 PresentBlox 的瀏覽器視窗中，按一下 Portlet 的編輯按鈕 (帶有鉛筆圖示的按鈕)。即會呈現下拉選項清單。
2. 在下拉清單中，選取「格線 Blox」，再按一下**確定**。

重新整理頁面後，您應看到 400x400 的 GridBlox。依預設會開啓此 GridBlox 的功能表列及工具列。關閉功能表列及工具列後，便可將其大小變更為 400x100。

3. 瀏覽至先前安裝 DB2 Alphablox Portlet 範例」應用程式之下的 PA\_x\_x\_xx.war/jsp/html/目錄。
4. 在 JSP 編輯器中開啓 grid.jsp。此頁面看起來幾乎與 present.jsp 相同，但下列除外：

- <blox:present> 標示現在表示為 <blox:grid>，且 id 的值不同：

```
<blox:grid id="gridBlox" bloxName="<%= bloxName %>" width="400">
<blox:data dataSourceName="canned" />
</blox:grid>
```

- bloxName 值不同：

```
<%  
String bloxName = portletResponse.encodeNamespace("gridBlox");  
%>
```

5. 將此 GridBlox 的高度設為 100 像素，方法是新增 height 屬性並將其值設為 100：

```
<blox:grid id="gridBlox" bloxName="<%= bloxName %>" width="400" height="100">  
<blox:data dataSourceName="canned" />  
</blox:grid>
```

6. 關閉頂端的功能表列及工具列，方法是將 menubarVisible 及 toolbarVisible 屬性設為 false：

```
<blox:grid id="gridBlox" bloxName="<%= bloxName %>" width="400" height="100"  
menubarVisible="false" toolbarVisible="false">  
<blox:data dataSourceName="canned" />  
</blox:grid>
```

請確定您輸入的屬性名稱正確無誤，包括大小寫（兩個屬性名稱中均使用大寫字體“V”）。亦請確定屬性都加在結束角括弧（“>”）的前面。

7. 將名稱空間變更為 myFirstGrid：

```
<%  
String bloxName = portletResponse.encodeNamespace("myFirstGrid");  
%>
```

變更此 GridBlox 的名稱空間是為了確保您剛才所進行的變更會在將此 JSP 載入您的入口網站時反映出來。由於先前您已載入此頁面，所以對於此階段作業，已經有一個在伺服器上執行之此 GridBlox 的實例。除非變更名稱空間，否則即使是重新整理頁面，您剛才所進行的變更也不會反映出來。變更名稱空間是一種在開發環境中快速測試變更的方法。另外，還可以開啓新的瀏覽器視窗，在伺服器上建立新階段作業的新物件。

8. 儲存檔案。

You are now ready to test this file on WebSphere Portal.

若要測試您剛才所進行的變更，請：

1. 回到入口網站網頁。
2. 按一下瀏覽器的「重新整理」按鈕，以重新載入頁面。

您應看到一個 400x100 的 GridBlox，無功能表列或工具列。

**註：**此 JSP 下拉選項清單並不是 WebSphere Portal 的內建功能，而是由此 Portlet 範例建立。檢查 WEB-INF/src/ 目錄中的 edit.jsp 檔案及 Java 原始檔。

---

## 建立 Portlet 專案以使用 Blox 元件

在開發工具中建立 Portlet 專案時，請確定已將 Blox 元件所需的 AlphabloxServer servlet 對映及標示庫參照新增至您專案的 web.xml 檔案中，並已將「DB2 Alphablox 標示庫」的 .tld 檔案複製到您的專案。

1. 修改專案的 web.xml 檔案，以併入下列命令行：

- 若為 servlet 定義及 servlet 對映：

```
<servlet>  
<servlet-name>AlphabloxServer</servlet-name>  
<servlet-class>com.alphablox.server.webapps.server.AlphabloxServer  
</servlet-class>
```

```

</servlet>
<servlet-mapping>
<servlet-name>AlphabloxServer</servlet-name>
<url-pattern>/abx/*</url-pattern>
</servlet-mapping>

```

- 若為標示庫參照：

```

<taglib>
<taglib-uri>bloxtld</taglib-uri>
<taglib-location>/WEB-INF/tlds/blox.tld</taglib-location>
</taglib>
<taglib>
<taglib-uri>bloxformtld</taglib-uri>
<taglib-location>/WEB-INF/tlds/bloxform.tld</taglib-location>
</taglib>
<taglib>
<taglib-uri>bloxlogictld</taglib-uri>
<taglib-location>/WEB-INF/tlds/bloxlogic.tld</taglib-location>
</taglib>
<taglib>
<taglib-uri>bloxreporttld</taglib-uri>
<taglib-location>/WEB-INF/tlds/bloxreport.tld</taglib-location>
</taglib>
<taglib>
<taglib-uri>bloxportlettld</taglib-uri>
<taglib-location>/WEB-INF/tlds/bloxportlet.tld</taglib-location>
</taglib>
<taglib>
<taglib-uri>bloxuitld</taglib-uri>
<taglib-location>/WEB-INF/tlds/bloxui.tld</taglib-location>
</taglib>

```

2. 將「DB2 Alphablox 標示庫」的 .tld 檔案複製到您專案的 WEB-INF/tlds/ 目錄。這些檔案位於下列目錄之下：

```
<db2alphablox_dir>/bin/
```

其中，<db2alphablox\_dir> 是您的 DB2 Alphablox 安裝目錄。

## 使用 Rational Application Developer 配置 Portlet 專案

WebSphere Portal 建議使用 Rational Application Developer 進行 Portlet 開發。特別是，WebSphere Portal 5.1 版需要使用 Rational Application Developer 6.0 版。Rational Application Developer 提供精靈來引導您完成 Portlet 專案的設定及建立。它會根據您的選擇，設定適當的結構及部署描述子檔案，並自動建立您控制器所需的 Java 類別及用於 Portlet 檢視畫面的 JSP 頁面。當您遵循「新增 Portlet 專案」精靈進行操作時，請確定下列配置正確：

- 選擇「WebSphere Portal 5.1 版 Stub」，作為您的目標伺服器。這是在精靈的第一個畫面上指定，方法是按一下顯示進階 >> 按鈕。
- 將 DB2 Alphablox 的 servlet 對映新增至在 WebContent/WEB-INF/ 下建立的 web.xml 檔案。完成此作業的方法是在您專案的 web.xml 中輸入下列 servlet 定義及對映程式碼：

```

<servlet>
<servlet-name>AlphabloxServer</servlet-name>
<servlet-class>com.alphablox.server.webapps.server.AlphabloxServer
</servlet-class>
</servlet>

```



```

<servlet-mapping>
<servlet-name>AlphabloxServer</servlet-name>
<url-pattern>/abx/*</url-pattern>
</servlet-mapping>

```

- 將「Alphablox 標示庫」新增至您專案的 web.xml 中。遵循下面的步驟：
  1. 開啓 web.xml 時，按一下**變數**標籤。
  2. 捲動到底部的「標示庫參照」區段。
  3. 按一下**新增**。
  4. 對於 **URL** 及**位置**，根據下表輸入值：

URL	位置
bloxtld	/WEB-INF/tlds/blox.tld
bloxformtld	/WEB-INF/tlds/bloxform.tld
bloxlogictld	/WEB-INF/tlds/bloxlogic.tld
bloxreporttld	/WEB-INF/tlds/bloxreport.tld
bloxportlettld	/WEB-INF/tlds/bloxportlet.tld
bloxuitld	/WEB-INF/tlds/bloxui.tld

對於每組 URL 及「位置」，您都需要按一下**完成**來新增標示庫參照，然後按一下**新增**來新增下一組。

**註：**如需不同標示庫及其用法的詳細資料，請參閱「使用 JavaServer Pages 及 Blox」。

最後，請記住將「DB2 Alphablox 標示庫」的 .tld 檔案複製到您專案的 WEB-INF/tlds/ 目錄。這些檔案位於下列目錄之下

```
<db2alphablox_dir>/bin/
```

其中，<db2alphablox\_dir> 是您的 DB2 Alphablox 安裝目錄。

---

## 後續步驟

您已瞭解如何建立含 Blox 元件的 Portlet 及基本的程式碼結構。若要顯示您資料庫中的資料，則需要：

1. 將新的資料來源定義到指向您資料來源的 DB2 Alphablox。
2. 修改 DataBlox 的 dataSourceName 屬性，以指向所定義的資料來源。
3. 將帶有查詢字串的 query 屬性新增至 DataBlox。

如需建立資料來源的詳細資料，請參閱**管理手冊**。若要瞭解如何建立適當的查詢，請參閱 Developer's Guide 中的 Retrieving Data 及 Developer's Reference 中的 DataBlox 一節。

一旦將您自己的資料顯示在 Blox 檢視畫面中，就可以開始探索許多可使用 Blox 標示來設定的內容。

---

## Portlet 開發要訣

當您使用 DB2 Alphasblox 文件中提供的資訊並繼續進行 Portlet 開發時，有一些您需要瞭解的 Portlet 特定主題、概念及一般準則。下列清單彙總了這些一般開發準則，以及文件中您所感興趣之特定章節的指標。

- 一律使用 Portlet 的名稱空間編碼您的 Blox 名稱。名稱空間可確保現行 J2EE 階段作業中的 Blox 名稱是唯一的。
- 一律使用像素設定 Blox 的寬度及高度。在入口網站環境之外，您可以將寬度及高度設為百分比值，例如，"50%" 或 "100%"。頁面上有多個 Portlet 同時存在，所以百分比值不會在入口網站環境中生效。
- 請勿使用相對的 URL 在 Portlet 中呼叫資源。URL 應使用 PortletResponse 類別的 `encodeURL()` 方法來編碼。在文件中顯示使用相對 URL 之處，都應一律對 URL 進行編碼。
- 若要確保 Blox Portlet 使用的主題與入口網站主題相似，請使用「入口網站主題公用程式」。此公用程式可從兩個來源取得：

- DB2 Alphasblox 首頁。

在**管理**標籤下，按一下**一般**鏈結。「公用程式」會在「入口網站」區段下列出。請參閱**管理手冊**及線上說明，以取得詳細資料。

- AlphasbloxAdminPortlets.war 檔案。

此為 Portlet 版本的「入口網站主題公用程式」。您可以安裝此 Portlet，以便在整個入口網站中執行，而無需個別登入 DB2 Alphasblox 首頁。

此公用程式會將入口網站環境的樣式與 DB2 Alphasblox 中的樣式合而為一，如此 Blox 顯示的顏色與字型就會與頁面上其他 Portlet 相似。

- 查看 *Developer's Guide* 中的 Blox Portlet Tag Library 主題，以尋找在規劃階段您需要考量的問題。
- DB2 Alphasblox 具有一個功能強大的「Blox UI 模型」。「UI 模型」包含 ClientLink 物件，在按下頁面上的 Blox 元件時讓您載入特定的 URL。若為 Portlet 開發，請使用「Blox Portlet 標示庫」建立 ClientLink。標示會處理 URL 格式，且是動態處理，如此在重新整理頁面之後，鏈結便不會變得陳舊。請參閱 *Developer's Guide* 中的 Planning for Portlet Development 主題，以取得詳細資料。
- 若為 Portlet 及動作 URI，請使用「Blox Portlet 標示庫」建立 Portlet 鏈結或動作鏈結。然後，您可以利用入口網站的 Portlet API 來處理動作或 Portlet 鏈結。請參閱 *Developer's Guide* 中的 Blox Portlet Tag Library 主題，以取得詳細資料。



---

## 第 3 章 指導教學：利用 Rational Developer 工具建置應用程式。

您可搭配使用 Rational Developer 工具與 DB2 Alphablox Toolkit 更快速地開發其中使用 DB2 Alphablox 元件及 DB2 Alphablox Java API 的應用程式。

Rational Developer 工具 (Rational Application Developer 或 Rational Web Developer) 可與 DB2 Alphablox Toolkit (Eclipse 型外掛程式集) 搭配使用，來開發及測試使用 DB2 Alphablox 技術建置的應用程式。此教學指導會引導您將 Rational 整合開發環境配置為可使用方法和完整標示，以及使用 DB2 Alphablox Toolkit 新增的自訂加強功能。

開始使用 DB2 Alphablox Toolkit 配置您的 Rational Developer 工具，來開發並測試 DB2 Alphablox 應用程式。

---

### 準備開發環境

您應先確定具備所有所需的軟體來準備開發環境，才可以安裝並開始使用 DB2 Alphablox Toolkit。

準備開發環境：

1. 在您的工作站上安裝 Rational Developer 工具 (Rational Application Developer 或 Rational Web Developer)。
2. 安裝任何所需的更新資料，以確定您至少有 Rational Application Developer 或 Rational Web Developer 6.0.0.1 版。
3. 如果您計劃使用 WebSphere 5.1 整合測試環境，請確定您已更新 Rational Developer 版本以包含此選用的安裝。

**註：**預設的 Rational Developer 安裝工具不包括 WebSphere 5.1 整合測試環境。

現在，您可以開始執行本指導教學中的作業。

---

### 安裝 DB2 Alphablox Toolkit

將 DB2 Alphablox Toolkit 與 Rational Developer 工具搭配使用，可讓您更輕鬆地開始開發包含 DB2 Alphablox 內容的 Web 型應用程式。

DB2 Alphablox Toolkit 需要使用 Rational Application Developer 或 Rational Web Developer 6.0.0.1 版。

將 DB2 Alphablox Toolkit 安裝到 Rational Developer 工具，避免執行許多必要的手動步驟，才能使用內容完成 Java 方法和 JSP 自訂標示。此外，還可利用自訂精靈與提要，來引導您建立包含 DB2 Alphablox 內容的應用程式及 WebSphere 伺服器實例。

若要將 DB2 Alphablox Toolkit 安裝到 Rational Application Developer 或 Rational Web Developer：

1. 將 DB2 Alphablox 安裝光碟放入工作站的光碟機中。
2. 在 plugin 目錄，尋找名為 UpdateSite 的子目錄。

3. 將 UpdateSite 目錄複製到您硬碟上方便使用的位置。例如，將目錄複製到 C:\DB2Alphablox\UpdateSite。
4. 啟動 Rational Developer 工具。
5. 在功能表列上，選取說明 > 軟體更新 > 搜尋並安裝。
6. 在開啓的「安裝」視窗中，選取搜尋要安裝的新增功能選項，然後按下一步。
7. 在「要造訪的更新站台」視窗上，按一下新本端站台按鈕，並瀏覽至 UpdateSite 目錄位置。
8. 按下一步，選取 **DB2 Alphablox Toolkit** 功能，再按下一步。
9. 在後續視窗中，接受授權合約，並選取要安裝 DB2Alphablox Toolkit 功能的位置。
10. 完成安裝之後，重新啟動 Rational Developer 工具

重新啟動 Rational Developer 工具之後，即可使用 DB2 Alphablox Toolkit 功能。

---

## 在 WebSphere 整合測試環境中安裝 DB2 Alphablox

在這項作業中，您要在 WebSphere 整合測試環境將 DB2 Alphablox 安裝到 Rational Developer 工具。

**必備項目：**支援的 WebSphere 整合測試環境必須已安裝在 Rational Developer 工具中。您需要存取 DB2 Alphablox 安裝光碟。檢視並執行 *DB2 Alphablox 安裝手冊*「前置安裝作業」一節中所說明的必要前置安裝步驟。

DB2 Alphablox 可安裝在獨立式 WebSphere 應用程式伺服器上，或安裝到 Rational Developer 工具中可用的 WebSphere 整合測試環境中。下面概述的步驟摘要說明了安裝中的差異，這些差異必須在將 DB2 Alphablox 安裝在 Rational Developer 工具中可用的 WebSphere Runtime 上時執行。

若要在 WebSphere 整合測試環境中安裝 DB2 Alphablox

遵循 *DB2 Alphablox 安裝手冊*「安裝」一節中的步驟，安裝 DB2 Alphablox，但需要注意下列例外狀況：

1. 在安裝程式的「配置 WebSphere」視窗的 **WebSphere 根目錄**欄位中，指定您要用作整合測試環境伺服器的 WebSphere Runtime 位置。

例如，若要在預設安裝時將 DB2 Alphablox 安裝到 WebSphere 6 整合測試環境，請選取 base\_v6 目錄的路徑。對於一般 Rational Application Developer 安裝，路徑會與下列類似：

```
C:\Program Files\IBM\Rational\SDP\6.0\runtimes\base_v6
```

2. 在「WebSphere 設定」視窗中，指定 WebSphere 管理者的名稱和密碼。在 WebSphere 整合測試環境中，不會使用這些項目值，但「DB2 Alphablox 安裝程式」會需要使用這些值。

**重要：**不要執行 *DB2 Alphablox 安裝手冊*中所說明的後置安裝步驟。

將 DB2 Alphablox 安裝進 WebSphere Runtime 後，您可以建立 WebSphere 伺服器實例，測試在 Rational Developer 工具中的 DB2 Alphablox 應用程式。

---

## 建立 WebSphere 伺服器實例

若要執行包含 DB2 Alphablox 內容的應用程式或 JSP 檔，您必須建立一個可存取所需 DB2 Alphablox 服務及 Java 類別的 WebSphere 伺服器實例。

**必備項目：**配置 Rational Developer 工具。安裝 DB2 Alphablox Toolkit。在 WebSphere 整合測試環境中安裝 DB2 Alphablox。

若要建立 WebSphere 伺服器實例，以測試包含 DB2 Alphablox 內容的應用程式及 JSP 檔：

1. 開啟 Rational Developer 工具。
2. 按一下**伺服器**標籤 (如果看不到該標籤，請選取視窗 > 顯示檢視畫面 > 伺服器)。
3. 用滑鼠右鍵按一下檢視畫面視窗內部，並選取**新增 > 伺服器**。即會開啟「新增伺服器」視窗。
4. 輸入 localhost 作為主機名稱，並選取伺服器類型。按**下一步**。
5. 輸入伺服器埠號，然後按**下一步**。
6. 新增要在此伺服器實例中執行的可用專案至已配置的專案清單中。按一下**完成**。便在**伺服器**檢視畫面中開啟新的伺服器實例。

如果您所建立的是 WebSphere 5.1 伺服器實例，則需要執行下一步「建立 WebSphere 5.1 伺服器替代變數」。至於 WebSphere 6 伺服器實例，則繼續執行「新增來賓至 DB2 Alphablox 管理者群組」。

---

## 建立 WebSphere 5.1 伺服器替代變數

在這項作業中，您將會修改 WebSphere 5.1 伺服器實例的替代變數，才能在 WebSphere 整合測試環境中適當地執行 DB2 Alphablox。

1. 開啟**伺服器**檢視畫面，並按兩下您要修改的 WebSphere 5.1 伺服器實例。
2. 按一下伺服器實例的**配置**標籤。
3. 選取**啟用管理主控台**，並取消選取**啟用通用測試用戶端**。
4. 按一下伺服器實例的**變數**標籤。即會開啟**替代變數**視窗。
5. 使用**新增**按鈕 (位於**節點設定**清單旁)，新增兩個變數。
  - a. 新增名為 WS\_EAR\_AlphabloxPlatform 的變數，並將值設為 \$(APP\_INSTALL\_ROOT)/localhost/AlphabloxPlatform.ear。
  - b. 新增名為 WS\_EAR\_AlphabloxStudio 的第二個變數，並將值設為 \$(APP\_INSTALL\_ROOT)/localhost/ApplicationStudio.ear
6. 儲存變更。

若為 WebSphere 5.1 伺服器實例，您需要遵循「配置 WebSphere 5.1 伺服器實例」中的步驟，才能完成修改伺服器實例。

---

## 配置 WebSphere 5.1 伺服器實例

若為 WebSphere 5.1 伺服器實例，您就必須配置伺服器實例以執行 DB2 Alphablox 應用程式及檔案。

**必備項目：**建立所需的 WebSphere 5.1 伺服器替代變數。

若要配置 WebSphere 5.1 伺服器實例：

1. 開啓 Rational Developer 工具中的**伺服器**檢視畫面，並啓動您要配置的伺服器實例。
2. 用滑鼠右鍵按一下伺服器實例，並選取**執行管理主控台**。
3. 請勿在 **ID** 欄位中輸入值。按一下**確定**。
4. 按一下配置表下面的**確定**按鈕。
5. 開啓**企業應用程式**導覽檢視畫面，再按一下 **ApplicationStudio** 應用程式名稱。
6. 將**應用程式二進位檔**設為 \$(WS\_EAR\_ApplicationStudio)。
7. 選取使用二進位檔 **meta** 資料。
8. 按一下配置表下面的**確定**按鈕。
9. 按一下 WebSphere 管理主控台上的**儲存**按鈕。
10. 在「儲存至主要配置」視窗中，按一下**儲存**。
11. 在 WebSphere 管理主控台的**應用程式 > 企業應用程式**下，啓動 AlphabloxPlatform 及 ApplicationStudio 應用程式。

已配置伺服器實例。接下來，您需要執行「將來賓新增至 DB2 Alphablox 管理者群組」步驟。

---

## 將來賓使用者新增至 DB2 Alphablox 管理者群組

將來賓使用者新增至 DB2 Alphablox 管理者群組，可讓您在使用 Rational Developer 工具時存取 DB2 Alphablox 管理頁面。

**必備項目：**建立 WebSphere 伺服器實例。配置 WebSphere 伺服器實例。

將來賓使用者新增至 DB2 Alphablox 管理者群組：

**重要：**在您正式作業的 WebSphere 伺服器上，不可將管理者權限授與來賓使用者。

1. 使用下列 Telnet 命令，登入 DB2 Alphablox 主控台：`telnet localhost portNumber`。其中，*portNumber* 是您在 DB2 Alphablox 安裝期間指定的埠。
2. 在 Telnet 主控台提示下，鍵入下列 DB2 Alphablox 主控台命令：`set Administrators guest`，然後按 Enter 鍵。
3. 在 Telnet 主控台中，鍵入 `save`，然後按 Enter 鍵。
4. 關閉 Telnet 階段作業。

現在可以開始使用您的 WebSphere 伺服器實例。

---

## 建立 DB2 Alphablox 應用程式

當您在 Rational Developer 中建立新的應用程式時，您需要新增應用程式所需的 DB2 Alphablox 內容，才能適當地執行。

**必備項目：**將 DB2 Alphablox Toolkit 安裝在 Rational Developer 工具上。

1. 在 Rational Developer 工具功能表列，選取**檔案 > 新增專案**。即會開啓**新增專案**精靈。
2. 展開 **Web** 選項，選取**動態 Web 專案**，然後按**下一步**。
3. 鍵入專案名稱，再按一下**顯示進階**按鈕。即會出現其他選項。
4. 選取適當的 servlet 版本及目標伺服器。
5. 按**下一步**。便會開啓「功能選項」視窗。
6. 選取 **DB2 Alphablox 內容**選項，然後按一下**完成**

您的新 Web 應用程式專案現在已具有 DB2 Alphablox 功能。DB2 Alphablox 標示庫及 Blox Java API 均可用，且應用程式描述子檔案 (web.xml) 已經過修改，併入了所需的 DB2 Alphablox 資訊。現在，您可以開始將包含 DB2 Alphablox 內容的 JSP 檔新增至您的專案。

---

## 建立含 DB2 Alphablox 內容的 JSP 檔

在這項作業中，您要搭配使用 Rational Developer 工具與 DB2 Alphablox Toolkit，建立可存取 Blox 標示庫的新 JSP 檔。

**必備項目：**將 DB2 Alphablox Toolkit 安裝在 Rational Developer 工具上。

1. 在 Rational Developer 的**專案瀏覽器**檢視畫面中，選取**檔案 > 新增 > JSP 檔**。
2. 在**檔名**欄位中鍵入檔案名稱。
3. 按一下**配置進階**選項，並按**下一步**。
4. 新增將在 JSP 檔中使用的 DB2 Alphablox 標示庫。
  - a. 按一下**新增**按鈕，以開啓「新增標示庫」視窗。
  - b. 選取您計畫要在 JSP 檔中使用的 DB2 Alphablox 標示庫。
  - c. 按**下一步**。
5. 從**編碼**清單選取 **ISO 10646/Unicode(UTF-8)**。需要 UTF-8 編碼才能適當地執行 DB2 Alphablox 應用程式。
6. 按一下**完成**。新的 JSP 檔即會顯示在專案清單中。
7. 按兩下該檔名，在 JSP 編輯器視窗中開啓檔案。
8. 將游標置於 HTML <head> 標示內，但必須在前述步驟所新增的 DB2 Alphablox JSP taglib 指引之後。
9. 在新的一行上鍵入下列內容，以輸入 Blox 標頭標示：`<blox:header/>` 執行 JSP 檔時，新增所需的 DB2 Alphablox JavaScript 及 CSS 檔案，需要有 Blox 標頭標示。

新的 JSP 檔現在已經可以存取您所選取的 DB2 Alphablox 標示庫。您可以使用 Rational Developer 工具的「內容輔助」功能，插入所選標示庫的 Blox 標示及標示屬性。

---

## 存取 DB2 OLAP Server 及 Essbase 資料來源

爲了存取 Rational Developer 工具中的 DB2 OLAP Server™ 或 Essbase 資料來源；使用 WebSphere 5.1 伺服器實例時，您必須建立可載入所需用戶端程式庫的啟動批次檔存取。

**必備項目：**安裝 DB2 Alphablox Toolkit。將 DB2 Alphablox 安裝到 WebSphere 5.1 整合測試環境。配置 WebSphere 5.1 伺服器實例。在開發機器上安裝 IBM® DB2 OLAP Server 或 Hyperion Essbase。

由於 Rational Developer 工具中 WebSphere 5.1 整合測試環境的限制，您必須建立啟動批次檔，讓 Rational Developer 工具存取 DB2 OLAP Server 或 Hyperion Essbase 資料來源。當您使用此批次檔來啟動 Rational Developer 工具時，所需的 Essbase 用戶端程式庫已加入 Java 程式庫路徑。

建立批次檔，以便使用 WebSphere 5.1 伺服器實例來啟動存取 DB2 OLAP Server 或 Essbase：

1. 使用文字編輯器並建立新的文字文件。
2. 新增一行程式碼以呼叫 DB2 Alphablox aassetup.bat 檔案 (此檔案位於 DB2 Alphablox 安裝目錄)。例如，下列程式碼將會執行位於指定 DB2 Alphablox 安裝中的 aassetup.bat 檔案：

```
call C:\alphablox\analytics\bin\aassetup.bat
```

3. 加入第二行來呼叫啟動 RAD 的 RAD rationalsdp.exe 檔。例如，下列程式碼會執行 Rational Developer 工具：

```
call C:\Program Files\IBM\Rational\SDP\6.0\rationalsdp.exe
```

4. 將此檔案儲存爲 startRAD.bat，位置在工作站桌面 (或其他方便的位置)。

當您按兩下 startRAD.bat 檔案，DB2 Alphablox aassetup.bat 會執行並設定必要的路徑及環境變數，然後就會啟動 Rational Application Developer。與 DB2 Alphablox 搭配使用的 Essbase 用戶端程式庫版本，必須符合您將用作資料來源的 DB2 OLAP Server (或 Hyperion Essbase) 版本。若要修改與 DB2 Alphablox 搭配使用的 Essbase 用戶端程式庫，則執行 DB2 OLAP Server / Essbase 用戶端程式庫公用程式 (ChangeEssbase.bat)，此程式位於 *db2\_alphablox\analytics\bin* 目錄，其中 *db2\_alphablox* 是 DB2 Alphablox 安裝的根目錄。當使用您所建立的 startRAD 批次檔來啟動 RAD 時，此公用程式會修改其中一個執行的批次檔。

上述的完整範例 startRAD.bat 檔案如下所示：

```
call C:\alphablox\analytics\bin\aassetup.bat  
call C:\Program Files\IBM\Rational\SDP\6.0\rationalsdp.exe
```

每當您想要存取 DB2 OLAP Server 或 Essbase 資料來源，就必須使用 startupRAD.bat 檔案來啟動 Rational Developer 工具。



---

## 第 4 章 指導教學：使用 DB2Cube Views 建置 DB2 Alphablox 資料方塊

此教學指導會引導您如何使用 DB2 Cube Views 範例資料庫來建立 DB2 Alphablox 資料方塊。

此教學指導中的作業不提供關於建置自訂 DB2 Alphablox 資料方塊的詳細資料。反而其目標是要讓您知道如何快速建立 DB2 Alphablox 資料方塊，用以探索 DB2 Alphablox Cube Server 功能。此結果資料來源也可以用來測試並建置 DB2 Alphablox 應用程式。

先決要件：

- 安裝 DB2 Alphablox。如需關於安裝的詳細資料，請參閱安裝手冊。
- 取得對支援 DB2 Cubes Views 實作的存取權，該實作中已安裝 DB2 Cube Views CVSAMPLE 範例資料庫。如需關於 DB2 Cube Views 支援版本的詳細資料，請參閱安裝手冊。

在此教學指導中，您將會學到如何以 DB2 Cube Views CVSAMPLE 範例資料庫為基礎，來建置 DB2 Alphablox 資料方塊。教學指導期間，您將會學到下列作業：

---

### 定義 DB2 關聯式資料來源

在這項作業中，您將會在 DB2 Alphablox 中定義 DB2 資料庫的資料來源定義。

必備項目：DB2 Alphablox 必須可存取所需的 DB2 JDBC 驅動程式。

DB2 Alphablox 資料方塊需要將基礎關聯式資料來源預先定義為 DB2 Alphablox 資料來源。使用 meta 資料及關聯式資料庫中可用的資料，來產生 DB2 Alphablox 資料方塊。

若要定義 DB2 資料庫的 DB2 Alphablox 資料來源：

1. 以 admin 使用者 (或具有管理者權限的使用者) 身分，登入 DB2 Alphablox 管理頁面。
2. 按一下**管理**標籤。
3. 按一下**資料來源**鏈結。
4. 按一下**建立**按鈕。
5. 從**配接卡**功能表，為您的資料庫伺服器選取適當的 IBM DB2 JDBC 驅動程式。

選取 **IBM DB2 JDBC 類型 4 驅動程式**或 **IBM DB2 UDB on iSeries 驅動程式**。

6. 在**資料來源名稱**欄位中，鍵入 CVSAMPLE，作為要用於您資料來源的名稱。
7. 在**伺服器名稱、埠號及資料庫名稱** (這個應為 CVSAMPLE) 中，輸入適當的值。

**註：**如果您需要判定這些欄位值是否正確的說明，請聯絡資料庫管理員。

8. 輸入**預設使用者名稱及預設密碼**。

使用者名稱及密碼必須對關聯式資料庫有效。當 DB2 Alphablox 資料方塊存取關聯式資料庫時，一律使用預設使用者名稱及密碼。您所使用的資料庫使用者應具有對資料庫的讀取存取權。

**註：**當指定的關聯式資料來源正在用於移入 DB2 Alphablox 資料方塊時，**使用 DB2 Alphablox 使用者名稱及密碼**的值會被忽略。存取控制清單 (ACL) 可用於限制對 DB2 Alphablox 資料方塊的存取權。如需 ACL 的相關資訊，請參閱**管理手冊**。

- 當資料來源正在用於移入 DB2 Alphablox 資料方塊時，**最大列數**及**最大直欄數**值會被忽略。您也可輸入值，其他應用程式使用該資料來源時會使用這些值。
- 除非您想要將 JDBC 記載資訊寫入 DB2 Alphablox 日誌檔，否則請將**啟用 JDBC 追蹤**值設為**否**。僅當您遇到問題並需要進行除錯以瞭解原因時，才啟用 JDBC 追蹤。
- 按一下**儲存**按鈕，以儲存資料來源定義。

您已定義了 CVSAMPLE 的 DB2 Alphablox 資料來源定義。現在您可以建立 DB2 Alphablox 資料方塊定義，用以存取 DB2 資料來源的 Cube Views™ 資料方塊 meta 資料。

---

## 定義 Alphablox Cube Server Adaptor 資料方塊

在這項作業中，您將會定義使用 Alphablox Cube Server Adapter 的 DB2 Alphablox 資料來源定義。

**必備項目：**建立 DB2 CVSAMPLE 資料庫的 DB2 Alphablox 資料來源定義。

若要定義 DB2 Alphablox Cube Server Adapter 資料來源，請：

- 以 admin 使用者 (或具有管理者權限的使用者) 身分，登入 DB2 Alphablox 管理頁面。
- 按一下**管理**標籤。
- 按一下**資料來源**鏈結。
- 按一下**建立**按鈕。
- 從**配接卡**功能表，選取 **Alphablox Cube Server Adapter** 選項。
- 在**資料來源名稱**欄位中，輸入 DB2AlphabloxCubes，作為要用於您資料來源的名稱。
- 按一下**儲存**按鈕，以儲存您的資料來源定義。

您已定義用於存取 DB2 Alphablox 資料方塊的 DB2 Alphablox 資料來源。接著，必須定義您可以存取的 DB2 Alphablox 資料方塊。

---

## 定義 DB2 Alphablox 資料方塊

在這項作業中，您將會以 DB2 Cube Views CVSAMPLE 資料方塊中的 meta 資料為基礎，定義 DB2 Alphablox 資料方塊。

**必備項目：**定義 DB2 關聯式資料資料來源。定義 DB2 Alphablox Cube Server Adapter 資料來源。

若要定義 DB2 Alphablox 資料方塊的一般內容，請：



1. 以 admin 使用者 (或具有管理者權限的使用者) 身分，登入 DB2 Alphablox 管理頁面。
2. 按一下**管理**標籤。
3. 按一下**資料方塊**鏈結。
4. 按一下**建立**按鈕。 便會開啓「資料方塊管理」視窗。
5. 定義新的資料方塊：
  - a. 在 **DB2 Alphablox 資料方塊名稱**欄位中，鍵入 CVSales。
  - b. 選取位於 **DB2 Alphablox 資料方塊名稱**欄位旁的**啓用**選項。 選取這個選項可讓您的資料方塊在重新啓動伺服器時自動啓動。
  - c. 從**關聯式資料來源**功能表，選取 MyDB2 (您在本指導教學中建立的關聯式資料來源)。
  - d. 不要選取**安全角色**選項。 此選項可用來限制使用者對特定資料方塊的存取能力。
6. 啓用 DB2 Cube Views 設定並指定要使用的 meta 資料：
  - a. 選取**啓用 DB2 Cube Views 設定**選項。
  - b. 從**資料方塊模型**功能表，選取 **CVSAMPLE.Sales**。
  - c. 從**資料方塊**功能表，選取**一般銷售**資料方塊。
  - d. 選取**使用商業名稱**圓鈕，以指定名稱。 選取這個選項後，即可使用一般較為有意義且易讀的成員名稱。
  - e. 按一下**匯入資料方塊定義**按鈕。 使用此選項可讓您匯入資料方塊定義並預先將度量值及維度移入 DB2 Alphablox 資料方塊中。匯入的資料方塊定義會根據支援的 DB2 Cube Views meta 資料，反映出盡可能與 DB2 Alphablox 相符的 DB2 Cube Views 資料方塊。當您對使用資料方塊有更多經驗時，就可以修改度量值及維度來滿足您的需要。
  - f. 按一下**顯示匯入日誌**按鈕，以查看其中指定了與匯入作業相關之資訊及除錯訊息的日誌。 本指導教學中包含此步驟的目的只是爲了協助您熟悉此功能。
  - g. 選取**啓動、重新建置及編輯時匯入資料方塊定義**選項。 此選項會讓您的 DB2 Alphablox 資料方塊在每次啓動、重新建置或開啓 DB2 Alphablox 資料方塊以進行編輯時，都使用最新的 DB2 Cube Views 定義。進一步熟悉 DB2 Alphablox 及 DB2 Cube Views 之後，您就可以匯入資料方塊定義並進行自訂。
7. 按一下**儲存**按鈕，以儲存 DB2 Alphablox 資料方塊定義。

您已建立了 DB2 Alphablox 資料方塊定義。現在您可以啓動新的 CVSales 資料方塊。

---

## 啓動 DB2 Alphablox 資料方塊

在這項作業中，會使用 DB2 Alphablox 管理頁面來啓動 CVSales 資料方塊。

**必備項目：**定義 DB2 關聯式資料資料來源。定義 DB2 Alphablox Cube Server Adapter 資料來源。定義 DB2 Alphablox 資料方塊。

若要啓動 CVSales 資料方塊，請：

1. 以 admin 使用者 (或具有管理者權限的使用者) 身分，登入 DB2 Alphablox 管理頁面。
2. 按一下**管理**標籤。
3. 在**執行時期管理**區段下，按一下**資料方塊**鏈結。

4. 從 **DB2 Alphablox 資料方塊**清單，選取您要啟動的 DB2 Alphablox 資料方塊。
5. 按一下**啟動**按鈕。 啟動資料方塊後，狀態欄位會顯示**執行中**。

DB2 Alphablox 範例資料方塊正在執行中。您現在可以使用剛才建立的 DB2 Alphablox 資料方塊來開始建置應用程式。您可以使用「查詢建置器」應用程式針對新的資料方塊執行快速檢查、選取您的資料方塊作為資料來源並對資料方塊執行 MDX 查詢。

---

## 注意事項

本資訊是針對在美國所提供之產品與服務開發出來的。

在其他國家中，IBM 不見得有提供本書中所提的各項產品、服務或功能。若要取得您所在區域目前是否可使用這些產品與服務的相關資訊，請向 IBM 代表查詢。本書在提及 IBM 的產品、程式或服務時，不表示或暗示只能使用 IBM 的產品、程式或服務。只要未侵犯 IBM 的智慧財產權，任何功能相當的產品、程式或服務都可以取代 IBM 的產品、程式或服務。不過，其他非 IBM 產品、程式、或服務在運作上的評價與驗證，其責任屬於使用者。

在這本書中可能包含著 IBM 所擁有之專利或專利申請案。本書使用者並不享有前述專利之任何授權。您可以用書面方式，將授權查詢傳送給：

*IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.*

若要查詢有關二位元組 (DBCS) 資訊的特許權限事宜，請聯絡您國家的 IBM 智慧財產部門，或者用書面方式寄到：

*IBM World Trade Asia Corporation, Licensing, 2-31 Roppongi 3-chome, Minato-ku, Tokyo 106-0032, Japan*

下列段落若與當地之法令抵觸，則作廢：IBM 僅以「現狀」提供本出版品，而不為任何明示或默示之保證 (包括但不限於產品未涉侵權、可售性或符合特定效用的保證)。倘若干地區在特定交易中並不許可相關明示或默示保證之棄權聲明，則於該等地區之特定交易，此項聲明不適用之。

本資訊中可能包含技術上或排版印刷上的錯誤。因此，IBM 會定期修訂；並將修訂後的內容納入新版中。同時，IBM 得隨時修改及 (或) 變更本出版品中所提及的產品及 (或) 程式，不必另行通知。

本資訊中任何對非 IBM 所擁有的網站之敘述僅供參考，IBM 對該等網站並不提供保證。該等網站上的資料，並非本 IBM 產品所用資料的一部分，如因使用該等網站而造成損害，其責任由貴客戶自行負責。

IBM 得以其認定之各種適當方式使用或散佈由貴客戶提供的任何資訊，而無需對您負責。

本程式之獲授權者若希望取得相關資料，以便使用下列資訊者可洽詢 IBM。其下列資訊指的是：(1) 獨立建立的程式與其他程式 (包括此程式) 之間更換資訊的方式 (2) 相互使用已交換之資訊方式。若有任何問題，請聯絡：

*IBM Corporation, J46A/G4, 555 Bailey Avenue, San Jose, CA 95141-1003 U.S.A.*

上述資訊之取得有其特殊要件，在某些情況下必須付費方得使用。

IBM 基於雙方之「IBM 客戶合約」、「IBM 國際程式授權合約」或任何同等合約之條款，提供本出版品中所述之授權程式與其所有適用的授權資料。

此間所含之任何效能資料，皆是得自於經控制的環境之下；因此，於其他不同作業環境之下所得的結果，可能會有很大的差異。有些測定已在開發階段系統上做過，惟此並不保證在一般系統上會出現相同結果。再者，部份度量是利用插補法而得的估計值，其實際結果可能會有所不同。本文件的使用者應根據其特有的環境，驗證出適用的資料。

本書所提及之非 IBM 產品資訊，係一由產品的供應商，或其出版的聲明或其他公開管道取得。IBM 並未測試過這些產品，也無法確認這些非 IBM 產品的執行效能、相容性、或任何對產品的其他主張是否完全無誤。如果您對非 IBM 產品的性能有任何的疑問，請逕向該產品的供應商查詢。

所有與 IBM 未來方向或目的相關的聲明僅代表一種目標，隨時都可能變更或撤銷，不另行通知。

本資訊包含用於日常商業營運的資料及報告範例。爲了儘可能詳細清楚地說明，範例中使用了個人、公司、品牌及產品的名稱。所有這些名稱均屬虛構，如與實際商業企業所使用的名稱及地址雷同，純屬巧合。

本資訊包含原始語言的範例應用程式，說明了在各種操作平台上的程式設計技術。基於開發、使用、銷售或散佈符合作業平台 (用於執行所撰寫的範例程式) 應用程式設計介面的應用程式等目的，得以任何形式複製、修改及散佈這些範例程式，不必向 IBM 付費。這些範例並未在所有情況下完整測試。因此，IBM 不保證或暗示這些程式的穩定性、服務能力或功能。基於開發、使用、銷售或散佈符合 IBM 應用程式設計介面的應用程式等目的，得以任何形式複製、修改及散佈這些範例程式，不必向 IBM 付費。

---

## 商標

下列專有名詞是 International Business Machines Corporation 在美國及 (或) 其他國家的商標或註冊商標：

DB2	DB2 OLAP Server	DB2 Universal Database
IBM	WebSphere	

Alphablox 及 Blox 是 Alphablox Corporation 在美國或 (及) 其他國家的商標或註冊商標。

Java 以及所有以 Java 爲基礎的商標爲 Sun Microsystems, Inc. 在美國及 (或) 其他國家的商標。

Linux<sup>®</sup> 是 Linus Torvalds 在美國及 (或) 其他國家的商標。

其他公司、產品及服務名稱，可能是其他公司的商標或服務標誌。

---

## 索引

索引順序以中文字，英文字，及特殊符號之次序排列。

### 〔九劃〕

查詢

產生使用查詢 4

查詢建置器

使用 4

### 〔十一劃〕

教學課程

應用程式開發 1

## P

Portlet

範例 11

portlet

JSP structure 12



## 讀者意見表

為使本書盡善盡美，本公司極需您寶貴的意見；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
	※評估意見為"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。

IBM DB2 Alphablox  
入門

GC40-2038-01

8.4 版

折疊線

110 台北市信義區松仁路 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
台灣北區郵政管理局 登記證
北台字第 00176 號

(免貼郵票)

寄件人 姓名：  
地址：

寄

折疊線







程式號碼: 5724-L14

GC40-2038-01

