PART 2

基本開發工作篇

- Q3：我的程式需要爲檔案壓縮, 有現成的程式碼可以套用嗎？
- Q4：ASP.NET 開發伺服器可以直接上線使用, 而不安裝 IIS 嗎？
- Q5：ASP.NET 2.0 中不能使用自訂的命名空間 (namespace) 嗎？
- Q6：如何在 ASP.NET 中共用程式碼, 讓其他網頁中也可以使用？
- Q7：我需要的功能在內建的控制項做不到, 我該怎麼辦？
- Q8：爲甚麼我無法使用 Visual Studio 的除錯器?
- Q9：我要如何才能看到完整的錯誤訊息？
- Q10：預先編譯 (Pre-compile) 眞的可以提昇效率嗎？如何部署這樣的 ASP.NET 程式？
- Q11：要如何在同一台 IIS 中共用 ASP.NET 1.x 和 ASP.NET 2.0？
- Q12：如何將 ASP.NET AJAX 整合到現有的 ASP.NET 2.0 應用程式中？
- Q13：如何將 ASP.NET AJAX 1.0 的應用程式升級到 ASP.NET 3.5？

Q3 我的程式需要爲檔案壓縮，有現成的程式碼可以套用嗎？

適用範圍：☑ASP.NET 2.0 ☑ASP.NET 3.5

問題

我目前開發了一支 Web 程式, 需要在網路上傳輸許多的資料, 我想要在網路傳輸時, 能夠將資料的大小壓縮到最小, 另外我也有一支上傳文件資料的程式, 我想要在上傳後將檔案做壓縮, 以縮小佔用的磁碟空間, 請問我要如何做？

問題說明

在很多的應用中, 壓縮技術是很常被使用到的在 Microsoft .NET 1.x 中並沒有提供具有壓縮能力的類別可用, 在 Microsoft .NET 2.0 中才開始提供 GZip 與 Deflate 壓縮能力的類別使用, 這兩個類別都在 System.IO.Compression 命名空間中。

類別	說明
GZipStream	使用 GZip 演算法進行壓縮。
DeflateStream	使用 Deflate 演算法進行壓縮。

這兩個類別都支援壓縮與解壓縮工作, 並且由不同的壓縮方向與呼叫的方法來處理：

方向	說明
壓縮	列舉：CompressionMode.Compress 方法：Write()
解壓縮	列舉：CompressionMode.Decompress 方法：Read()

壓縮與解壓縮的方式其實和讀取與寫入資料流差不多, 只要掌握壓縮與解壓縮的方向與方法, 就可以寫出壓縮的程式碼, 如程式碼 1 所示。

程式碼 1：以 GZipStream 撰寫之壓縮程式碼

```
public byte[] GzipCompressData(byte[] SourceData)
{
   MemoryStream ms = new MemoryStream();
   GZipStream stream = new GZipStream(ms, CompressionMode.Compress);

   stream.Write(SourceData, 0, (int)SourceData.Length);
   stream.Flush();
   ms.Flush();
   result = ms.ToArray();

   stream.Close();
   ms.Close();
   return result;
}
```

而解壓縮的程式, 則可以這樣寫, 如程式碼 2 所示：

程式碼 2：以 GZipStream 撰寫的解壓縮程式

```
public byte[] DecompressGZipData(byte[] SourceData)
{
    byte[] result = null;
    byte[] buffer = null;
    MemoryStream source = new MemoryStream();
    int bufferSize = 10000;
    int readBytes = 0, offset = 0;

    source.Write(SourceData, 0, SourceData.Length);
    source.Flush();
    source.Position = 0;
    GZipStream stream =
```

```
        new GZipStream(source, CompressionMode.Decompress);
    MemoryStream ms = new MemoryStream();

    do
    {
        buffer = new byte[bufferSize];
        readBytes = stream.Read(buffer, 0, (int)buffer.Length);

        if (readBytes == 0)
            break;

        offset += readBytes;

        if (readBytes < buffer.Length)
            ms.Write(buffer, 0, readBytes);
        else
            ms.Write(buffer, 0, buffer.Length);
    }
    while (readBytes > 0);

    ms.Flush();
    result = ms.ToArray();

    stream.Close();
    ms.Close();

    return result;
}
```

而用 DeflateStream 撰寫的壓縮與解壓縮程式, 只要把程式碼列表中的GZipStream 改成DeflateStream, 就可以使用了。

解決方案

若要透過壓縮方式在網路上傳輸資料, 必須要注意幾件事情：

1. 兩邊都要使用相同的演算法。
2. 已經壓縮過的資料 (例如 GIF/JPEG 或已經是壓縮檔) 再壓縮, 縮小的程度也有限, 而且浪費 CPU 資源。

有了壓縮與解壓縮的方法, 再搭配這些事項, 就可以寫出傳送端的壓縮程式和接收端的解壓縮程式 (如程式碼 3 與 4) 。

程式碼 3：壓縮資料後輸出

```
// compress and write response (ASP.NET Page)
public void Page_Load(object sender, EventArgs e)
{
    // read data from database.
    byte[] data = DBService.GetFile(Request.QueryString["fileID"];
    // compress data.
    data = CompressGZipData(data);
    // Write to Client.
    Response.ContentType = "application/octet-stream";
    Response.BinaryWrite(data);
    Response.End();
}
```

程式碼 4：接收並解壓縮

```
// read program (Windows Forms Client)
public void cmdGetFile_Click(object sender, EventArgs e)
{
     HttpWebRequest request = WebRequest.Create(this.T_Url.Text) as
HttpWebRequest;
```

```
    HttpWebResponse response = request.GetResponse() as HttpWebResponse;
    Stream s = response.GetResponseStream();
    byte[] data = ReadStream(s); // custom function.

    data = DecompressGzipData(data);

    WriteFile(data, "FileName"); //  custom function.
}
```

考生停看聽

本問題所討論的內容, 可作爲準備 70-536: TS: Microsoft .NET Framework 2.0 Application Development Foundation 的下列主題：

- 在.NET Framework 應用程式中壓縮與解壓縮資料流資訊。
  - DeflateStream 類別
  - GZipStream 類別

Q4 ASP.NET 開發伺服器可以直接上線使用，而不安裝 IIS 嗎？

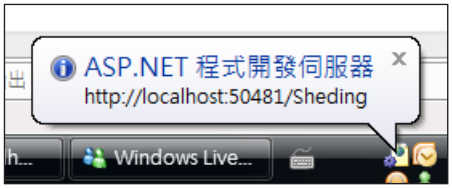
適用範圍：V ASP.NET 2.0　V ASP.NET 3.5

問題

我在 Windows XP Home 作業系統上開發 Web 應用程式, 因爲沒有 IIS, 我都是用檔案系統方式建立與發展應用程式, 在測試時都會啓動 ASP.NET 程式開發伺服器 (ASP.NET Development Server) , 平時在測試時也都很正常, 但是如果我想要開放給其他電腦的使用者連進來看, 可以嗎？

問題說明

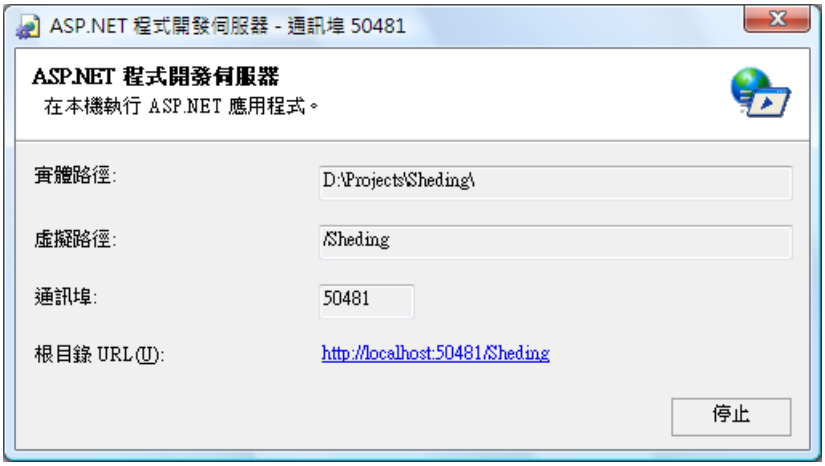
ASP.NET 程式開發伺服器是 Visual Studio 2005 才提供的新功能, 它可以允許開發人員在沒有 IIS 的情況下開發 Web 應用程式, 當開發人員在 Visual Studio 中使用檔案系統方式新增或開啓 Web 專案時, 就會內定使用 ASP.NET 程式開發伺服器來執行, 在開發人員執行 Web 應用程式時, ASP.NET 程式開發伺服器就會自動被執行起來。



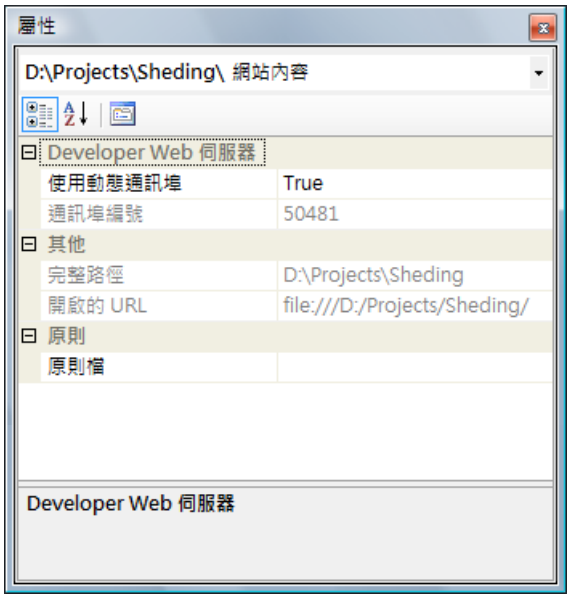
這個伺服器是用來提供本機測試與使用而設計的, 也因爲它只是供開發時的測試用, 因此它不像 IIS 一樣具有完全的功能, 它有幾個限制：

- 只能在本機上瀏覽與測試, 不支援由外部電腦瀏覽到 ASP.NET 程式開發伺服器。
- 只支援瀏覽應用程式範圍內的資料夾與檔案, 對於不在應用程式範圍內的檔案不提供服務。
- 不支援 IIS 上的特殊功能, 例如可程式化 IIS 元件、SMTP Service 等。

所以, 要讓 ASP.NET 程式開發伺服器開放給外部瀏覽是不可行的, 不過若想要修改 ASP.NET 程式開發伺服器使用的連接埠是可行的, 在預設的設定下, ASP.NET 程式開發伺服器使用動態連接埠 (Dynamic Port) , 也就是在啓動之時自動設定一個連接埠供瀏覽網站用, 在程式開發伺服器啓動時, 可以經由程式開發伺服器的**內容**頁, 看到目前使用的 URL 以及連接埠：

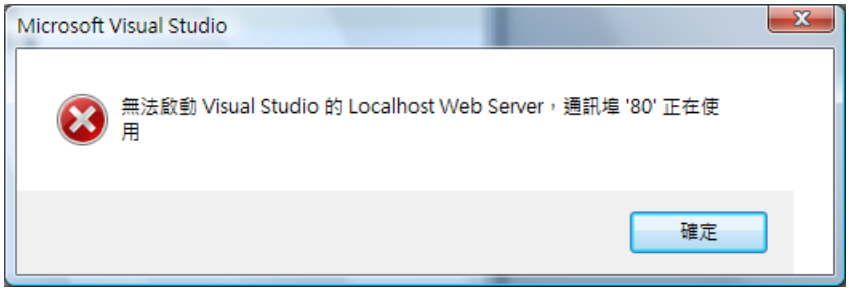


這個連接埠和啓動目錄的設定, 可以在專案的**屬性**頁設定, 將**使用動態連接埠**設定爲 false, 然後在**通訊埠編號**設定想要使用的連接埠即可：



請注意

若連接埠已經被其他程式使用時, 會造成啓動失敗, 並且出現下列的訊息：



所以在指定連接埠時應指定不被其他程式佔用的可用連接埠。

另外, ASP.NET 程式開發伺服器執行時使用的帳戶是目前登入的使用者的帳戶, 和 IIS 使用的帳戶是不同的, 因此若要在 ASP.NET 程式開發伺服器上測試一些必須要利用較高權限的帳戶才能執行的功能 (例如檔案複製) 時, 也許可以成功, 但部署到 IIS 時可能會因爲權限不足而失敗, 要利用 ASP.NET 程式開發伺服器來測試這類型的應用程式時, 要特別注意權限的問題。

解決方案

若想要讓其他電腦的使用者可以瀏覽這個網站, 或者想測試 IIS 的特殊功能, 請將網站移動到 IIS 伺服器上, 才可以開放與測試。

Q5 ASP.NET 2.0 中不能使用自訂的命名空間 (namespace) 嗎？

適用範圍： ☑ASP.NET 2.0 ☑ASP.NET 3.5

問題

我曾在 Visual Studio .NET 2003 (.NET 1.x) 以及 Visual Studio 2005 (.NET 2.0) 中開發 Web 應用程式, 在.NET 1.1 的時候, 每個 ASP.NET 應用程式都可以自行設定專案的命名空間 (namespace) , 這讓我在設計專案時可以識別元件之間的功能, 並且可以由外部程式來呼叫 ASP.NET 中的某些元件。但在 ASP.NET 2.0, 好像已經拿掉了這個功能, 請問在 ASP.NET 2.0 中, 是否有辦法設定命名空間？

問題說明

在 ASP.NET 1.x 的時代, 每個 ASP.NET 的網頁和元件, 都可以設定自己的命名空間, 當時在網頁的檔案中, 都會加入頁面中控制項的宣告與屬性設定的程式碼, 並且隱藏起來, 所以用 Visual Studio .NET 開發的開發人員應該經常在程式碼編輯器中看到這個區塊：

```
        this.BackOrderViewGrid.DataBind();
    }
}

Web Form 設計工具產生的程式碼

private void SaleViewGrid_ItemDataBound(object sender, System.Web.UI
{
    if (e.Item.ItemType == ListItemType.Item || e.Item.ItemType == I
```

這個區塊是由 Visual Studio 來維護, 專門用來放置頁面上所使用的控制項的建立與屬性設定的程式碼, 因爲這些程式碼是在同一個檔案中, 所以比較沒有命名空間設定上的問題。不過在 ASP.NET 2.0 (.NET 2.0) 時, 引進了一個新的功能－Partial Class, Partial Class 可以允許開發人員將同一個類別的程式碼, 切割在不同的檔案中, 這個功能在 C#和 VB.NET 皆有支援。

在 Visual Studio 2005 中, 不論是 Windows Forms 或是 ASP.NET 的 Web Form, 都大量的採用了這個新功能, 讓 Visual Studio 所維護的程式碼不會出現在開發人員的編輯器上, 在 Windows Forms 上因爲程式碼都是在新增和編輯表單時就自動加到程式檔 (*.Designer.cs 或*.Designer.vb) 中, 所以 Windows Forms 的專案是可以修改命名空間的。

但在 ASP.NET 2.0, 微軟將控制項對應的程式碼部份交給了 ASP.NET 的執行引擎 (aspnet_wp.exe 與 aspnet_isapi.dll) 來做, 也就是說, 控制項對應的程式碼是由 ASP.NET 的執行引擎來動態產生的, 在實際執行前的編譯時期, 才會由執行引擎來組合成單一的類別, 所以在 ASP.NET 2.0 程式碼中, 通常都只會看到這樣的宣告：

程式碼 1：預設的 ASP.NET 網頁程式碼原型

```
public partial class Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        …
    }
    …
}
```

而 Partial Class 的特性就是, 能夠將一個類別的程式碼放在多個檔案中, 這也表示, 開發人員只需要關注在自己的程式碼與事件處理上, 而不必關心控制項的宣告與設定的程式 (那個由執行引擎做掉了) 。然而, Partial Class 不支援跨命名空間, 若命名空間不同, 那麼 Partial Class 就會失效, 所以在 ASP.NET 2.0 中, 都只會存在一個預設的命名空間－ASP, 也就是說, 在執行引擎的編譯時期, 它所編譯的程式碼其實是這樣的：

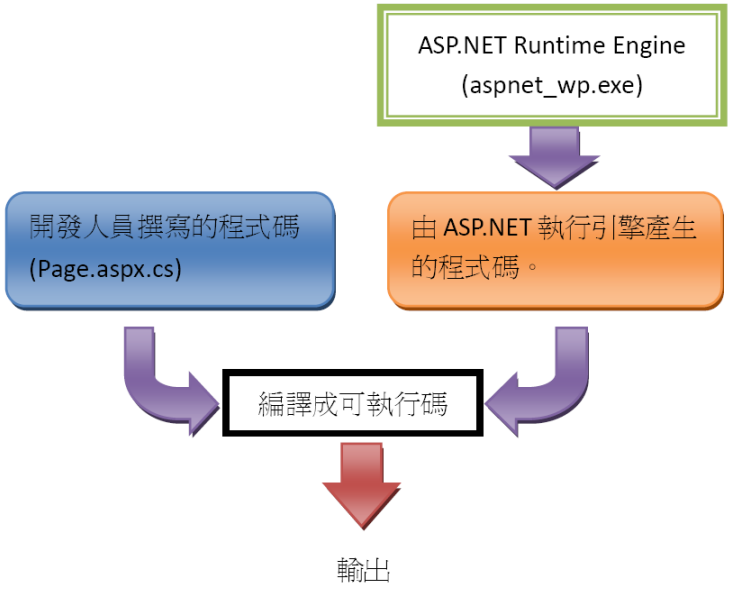
程式碼 2：執行引擎編譯時期的 ASP.NET 網頁程式碼原型－開發人員的程式碼

```
namespace ASP {
  public partial class Default : System.Web.UI.Page
  {
      protected void Page_Load(object sender, EventArgs e)
      {
          …
      }
      …
  }
}
```

程式碼 3：執行引擎編譯時期的 ASP.NET 網頁程式碼原型－由 ASP.NET 執行引擎產生的程式碼

```
namespace ASP {
  public partial class Default : System.Web.UI.Page
  {
      protected Button cmdButton1;
      protected Button cmdButton2;
      protected TextBox TextBox1;
      …
  }
}
```

在 Partial Class 無法跨命名空間的狀況下, Visual Studio 2005 的程式碼編輯器會將 namespace ASP 這個宣告隱藏起來, 所以開發人員既看不到, 也無法修改命名空間。這是因爲 ASP.NET 2.0 的程式碼模型的關係, 其編譯模式如圖所示。



解決方案

由於程式碼模型的限制關係, 故無法在 ASP.NET 2.0 中自行設定命名空間, 若想要將 ASP.NET 的功能開放給外部存取, 建議由類別庫來提供服務, 並且在設計時就妥善將功能由 ASP.NET 程式中切出, 並置入類別庫中, 而 ASP.NET 程式也可以藉由呼叫類別庫來取得所需要的功能。

相關問題

- Q6：如何在 ASP.NET 中共用程式碼, 讓其他網頁中也可以使用？

Q6 如何在 ASP.NET 中共用程式碼, 讓其他網頁中也可以使用？

適用範圍： V ASP.NET 1.0 V ASP.NET 1.1 V ASP.NET 2.0 V ASP.NET 3.5

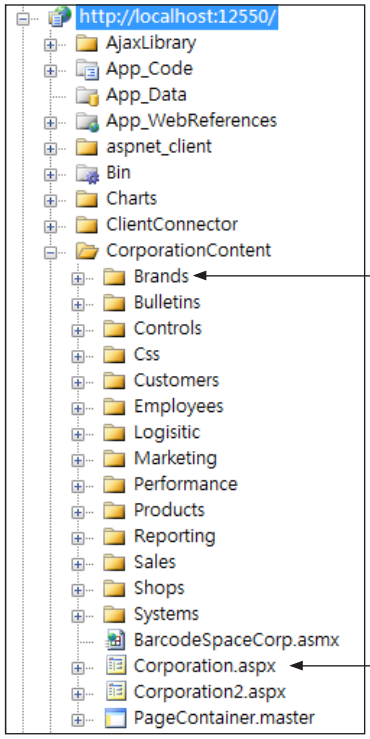
問題

我使用 Visual Studio 2005 開發 ASP.NET 網頁已有一段時間, 並且寫了許多的程式碼, 但現在因爲作業流程改變的關係, 我需要重新組織程式碼, 其中有些作業流程可能會需要由許多網頁共享, 我要如何把這個作業流程開放出來, 讓網頁可以共享這組程式碼？

問題說明

在 ASP.NET 中, 每個網頁都是一個獨立的類別, 並且擁有自己特有的程式碼與執行空間, 網頁之間在正常情況下是無法相互呼叫, ASP.NET 2.0 新增了一個 Cross-Page Post Back 的呼叫方式, 允許由網頁 A 提交它的 Post Back 訊息到網頁 B, 但這也只是允許讓網頁 B 處理網頁 A 的 Post Back 訊息, 若要眞正做到程式碼共享, 這個方法似乎還不夠好。

在 ASP.NET 1.x 時代, 網頁是具有命名空間的, 所以可以根據命名空間和類別名稱來尋找網頁類別, 用這種方法能做到由網頁類別開放程式碼, 但在 ASP.NET 2.0, 網頁不但沒有命名空間, 而且命名方式也做了改變, 例如下圖的目錄結構, 在 ASP.NET 1.x 和 ASP.NET 2.0 (Visual Studio .NET 和 Visual Studio 2005) 就是不同的命名方式：



路徑：/CorporationContent/Brands/BrandForm.aspx	
ASP.NET 1.x	ASP.NET 2.0
命名空間： ASP.CorporationContent.Brands 類別命名： BrandForm	類別命名： CorporationContent_Brands_BrandForm

路徑：/CorporationContent/Corporation.aspx	
ASP.NET 1.x	ASP.NET 2.0
命名空間： ASP.CorporationContent 類別命名： Corporation	類別命名： CorporationContent_Corporation

在這種情況下, 要靠網頁類別來共享程式碼是不太可行的, 因此 ASP.NET 2.0 新增了幾個可以共享應用程式資源的特殊資料夾, 開發人員可以利用這些資料夾來分享共用的資源, 這些資料夾稱爲『應用程式資料夾』。

資料夾	說明
App_Browsers	用來放置瀏覽器定義檔。
App_Code	用來放置要共享的程式碼類別, 這些類別會被編譯成 DLL 檔案, 若發現有變更時, 會自動重新載入整個 Web 應用程式。
App_Data	用來放置資料檔案, 這些檔案包含 SQL Server Express 資料檔 (MDF) 或是 XML 檔案, 若是要利用 SQL Server Express 使用 ASP.NET 2.0 的新功能, 例如會員服務或是角色管理時, 會在 App_Data 加入一個資料庫的檔案。
App_GlobalResources	用來放置適用於全域的資源檔 (*.resx 或 *.resource)。

資料夾	說明
App_LocalResources	用來放置只適用於特定頁面或使用者控制項的資源檔 (*.resx 或*.resource) 。
App_Themes	用來放置主題的定義資料 (包含 *.skin, *.css 或影像檔等) 。
App_WebReferences	用來放置加入 Web Service 參考時所產生的檔案 (WSDL, DISCO 等) 。
Bin	用來放置外部參考的類別庫, 以及在編譯後產生的 DLL 可執行檔案 。

App_Code 資料夾可以放置程式碼檔案, 也可以放置一些非程式碼的檔案 (如 WSDL 或 XSD 等) , 這些檔案也會被編譯到 DLL 中, 在大多數的設計案例中, App_Code 不會放置非程式碼的檔案, 筆者也不建議。盡可能讓這個資料夾單純化, 可以避免一些意外的問題。另外, 雖然 App_Code 預設並沒有硬性規定要放哪種程式語言, 但在同一個目錄下只能存在一種語言, 也就是 C#和 VB.NET 不能夠共存, 若想要讓 C#和 VB.NET 共存, 則要個別在 App_Code 中建立子目錄, 並且在 Web.config 檔案中設定：

取自 .NET Framework SDK 文件中的範例

```
<compilation debug="false">
    <codeSubDirectories>
        <add directoryName="VBCode" />
        <add directoryName="CSCode" />
    </codeSubDirectories>
</compilation>
```

這樣就可以在 App_Code 中放不同程式語言所撰寫的類別程式。

解決方案

將需要共用的應用程式碼放在單一個程式碼檔案 (*.cs 或*.vb) 中, 然後將它放到 App_Code, 就可以在應用程式中的所有網頁程式共享這個程式碼類別。

另一個可行的解決方案爲, 新增一個類別庫的專案, 將要共用的程式碼放到類別庫專案中, 再由 ASP.NET 專案引用這個類別庫, 即可使用。

經驗談

其實在大多數的情況下, 把共用程式碼寫到單一的類別庫是比較好的作法, 而且可以用在其他類型的用戶端程式 (例如 Windows Forms 或 Web Service 等) , 若寫在 App_Code 中, 就只能在 ASP.NET 程式中使用而已, 在可擴充性和可延展性的考量下, 應將共用程式碼寫在類別庫中較佳。

相關問題

- Q5：ASP.NET 中不能使用自訂的命名空間 (namespace) 嗎？

Q7 我需要的功能在內建的控制項做不到，我該怎麼辦？

適用範圍：V ASP.NET 1.0 V ASP.NET 1.1 V ASP.NET 2.0 V ASP.NET 3.5

問題

在 ASP.NET 2.0 中有提供了相當多的控制項可以用, 也大多能滿足我的需求, 但是就是有些控制項這邊少一點, 那邊少一點, 或者是我想要延伸控制項的功能, 那我應該要怎麼做？

問題說明

在 ASP.NET 2.0 中, 提供了大約 50 個左右的控制項, 可供開發人員選用。不過在某些情況下, 可能這些控制項都無法滿足開發人員的需求, 例如說想要展示圖表的控制項, 或者是想要和使用者互動更多的控制項等等, 可能以內建的方式做不到, 或者需要實作延伸功能時, 就需要自己設計。

在 ASP.NET 環境中, 可以利用三種方式來做控制項的設計與延伸。

方式	成本	說明
使用者控制項	低	以最快的方式組織一個或數個控制項，並且可以不必處理底層的機制，就可以應用在網頁中。
自現有控制項繼承	低／中	以最快的方式取得基礎的功能，然後在基礎上做延伸，若延伸的功能太多，則成本可能會提高。
自訂控制項	高	全部自己打造，要處理許多控制項本身的功能 (例如產生 HTML 碼的 Rendering 功能)，若對一些網頁基礎不了解，成本會較高。

一般來說, 使用者控制項 (User Control) 是最簡單、最快也最省事的控制項, 它可以在一個控制項內容中布置數個控制項, 組合成一個複合式控制項 (Composite Control) , 然後直接佈置到網頁上, 它的開發成本不但低, 開發所需要的時間也很短。例如若想要設計一個可以選取日期區間, 但是想用網路上有人設計好的日曆控制項時, 就可以利用使用者控制項來實作這種功能, 筆者手上有一個簡單的例子：

程式碼1：利用網路上的日曆控制項組合出來的控制項 (.ascx)

```
<%@ Control Language="C#" AutoEventWireup="true"
  CodeFile="DateRangeSelector.ascx.cs"
  Inherits="DateRangeSelector" %>
<link rel="stylesheet"
href="../../Scripts/calendar-win2k-cold-1.css" type="text/css" />
<script type="text/javascript" language="javascript"
  src="../../Scripts/calendar.js"></script>
<script type="text/javascript" language="javascript"
  src="../../Scripts/lang/calendar-en.js"></script>
<script type="text/javascript" language="javascript"
  src="../../Scripts/calendar-setup.js"></script>
<asp:Literal ID="ScriptSpace" runat="server"
  EnableViewState="false" />
<asp:TextBox ID="T_StartDate" runat="server" Columns="10"
  MaxLength="10" />
<img id="btn1" alt="" src="../../Scripts/img2.gif"
  style="cursor: pointer" />
<script type="text/javascript">

  Calendar.setup ({
      inputField : startDateTextBoxID,
      ifFormat   : "%Y/%m/%d",
      button     : "btn1",
      align      : "B1"
  });

</script>
至
<asp:TextBox ID="T_EndDate" runat="server" Columns="10"
  MaxLength="10" />
```

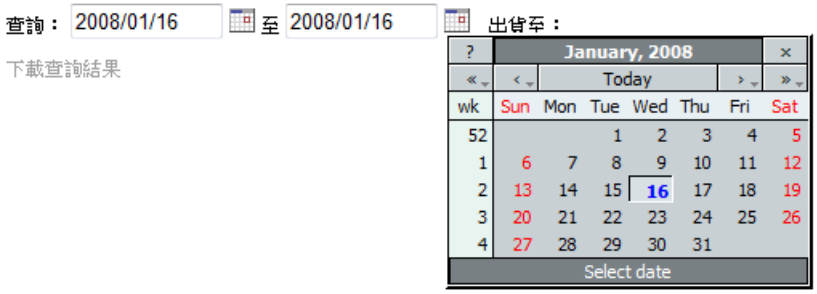
```
<img id="btn2" alt="" src="../../Scripts/img2.gif"
  style="cursor: pointer" />
<script type="text/javascript">

  Calendar.setup ({
      inputField : endDateTextBoxID,
      ifFormat   : "%Y/%m/%d",
      button     : "btn2",
      align      : "B1"
  });

</script>
```

程式碼 2：利用網路上的日曆控制項組合出來的控制項 (.ascx.cs)

```
public partial class DateRangeSelector : System.Web.UI.UserControl
{
  private DateTime _dateFrom = DateTime.Today;
  private DateTime _dateTo = DateTime.Today;

  protected void Page_Load(object sender, EventArgs e)
  {
    this.ScriptSpace.Text =
      "<script>" + "var startDateTextBoxID = \"" +
      this.T_StartDate.ClientID + "\";" +
      "var endDateTextBoxID = \"" + this.T_EndDate.ClientID +
      "\";" + "</script>";

    if (!Page.IsPostBack)
    {
      this.T_StartDate.Text =
      this._dateFrom.ToString("yyyy/MM/dd");
      this.T_EndDate.Text = this._dateTo.ToString("yyyy/MM/dd");
```

```
    }
  }

  public DateTime StartDate
  {
    get
    {
      DateTime returnDate =
      Convert.ToDateTime(this.T_StartDate.Text);
      return new DateTime(returnDate.Year, returnDate.Month,
      returnDate.Day, 0, 0, 0);
    }
    set
    {
      this._dateFrom = value;
    }
  }

  public DateTime EndDate
  {
    get
    {
      DateTime returnDate = Convert.ToDateTime(this.T_EndDate.Text);
      return new DateTime(
      returnDate.Year, returnDate.Month, returnDate.Day, 23, 59, 59);
    }
    set
    {
      this._dateTo = value;
    }
  }
}
```

這個例子是利用 Dynarch.com 所發展的 JavaScript Calendar 控制項所製作而成, 其程式碼與相關附件可由此下載：http://www.dynarch.com/projects/calendar/。這個控制項執行起來是這樣子：



像這種簡單功能的控制項, 就可以利用使用者控制項來實作, 快又簡單, 但它的缺點則是必須要跟著專案跑, 無法獨立出來單獨使用。

若是想要自己從頭打造, 則可以利用自訂控制項 (Custom Control) , 這種控制項要設計起來就沒這麼簡單, 除了要自己設計畫面外, 像是與 Visual Studio 開發環境的互動, 或是一些和應用程式環境的互動, 以及產生 HTML 或／和 Script 指令碼等, 都要自己一步一步的去做, 才可以打造出來, 它的優點是因爲它本身是個 DLL 檔案, 因此可以和多個專案共用, 而且可以單獨部署到其他電腦上。

若只是想要由現有的控制項去延伸一些功能, 則可以考慮由現有的控制項去繼承, 然後加上自己想要的功能, 在 ASP.NET 2.0 中, 絕大多數的控制項都可以繼承, 除了一些少部份的控制項 (類別有宣告 sealed 的) 無法利用這個方式來做, 那就只能自己打造了。

由於撰寫自訂控制項的主題已經可以寫成一本書了, 所以筆者就不再詳細說明, 可逕至書店找找, 有這類主題中文書可以看。

但如果需要的是具有高度複雜性, 而且在市場上有人提供設計好的控制項 (例如股市的 T 線圖表, 或是複雜度較高的圖表) 時, 爲了節省開發時間, 用外購的方式也許是不錯的方法, 不過這會動用到預算, 需要仔細思考以及收集資訊判斷後, 才能做出決定, 畢竟這些控制項的成本也不低。

解決方案

要視需求, 以及開發時程與成本, 來決定要如何發展控制項, 最簡單的是使用者控制項, 若需要複雜或是完全客制化的方案, 則可考慮使用自訂控制項, 但自訂控制項的開發成本較高, 時間也較久。因此如果是這類型的控制項, 向控制項的發展廠商來購買, 可能會是個比較好的解決方案。

考生停看聽

本主題所討論的內容, 可以用來準備下列考試中的主題。

Exam 70-528: TS: Microsoft .NET Framework 2.0 Web Application Development

- 建立自訂 Web 控制項
  - 建立複合式的 Web 控制項
    - 建立使用者控制項

Exam 70-547: PRO: Designing and Developing Web Applications by using .NET Framework

- 設計與開發使用者介面
  - 依設計規格選擇適合的控制項
    - 評估可使用的控制項, 包含 .NET Framework 內建的；自訂控制項；內部開發或是外部廠商的控制項。

相關問題

發展具有用戶端指令碼的控制項, 可參考

- Q14：如何在按下 Button 後, 彈出確認視窗或是資料輸入表單？
- Q16：如何在 ASP.NET 程式中輸出用戶端指令碼？ (變更題目名稱)
- Q23：如何使用用戶端指令碼動態產生控制項？
- Q24：如何動態產生控制項？
- Q25：如何動態產生圖片?

若要實作動態的使用者控制項載入, 可參考

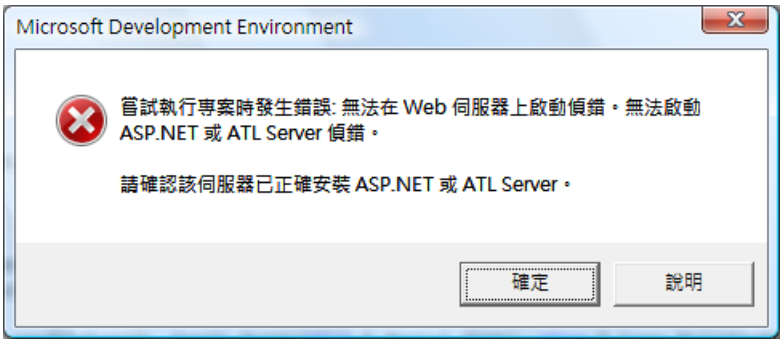
- Q74：如何動態載入自訂的使用者控制項？

Q8 爲甚麼我無法使用 Visual Studio 的除錯器?

適用範圍：VASP.NET 1.0 VASP.NET 1.1 VASP.NET 2.0 VASP.NET 3.5

問題

我使用 Visual Studio 2005 發展 Web 應用程式, 並且使用 IIS 當作應用程式的測試環境 (不是 ASP.NET Development Server) , 但是當我在啓動除錯器時, 都會出現類似於這樣的錯誤訊息：



請問我要如何解決？

問題說明

ASP.NET 應用程式的除錯器若搭配著 IIS 環境使用, 會有一些環境上的問題, 因爲 IIS 的環境是受限制的 (Restricted) , 因此除錯器若要在 IIS 上注射到應用程式的執行碼時, 就需要有一些足夠的權限。這種情況在 Windows Vista 中更爲明顯 (請參閱「Q2：如何在 Windows Vista 上啓用偵錯功能」) , 在 IIS 中啓動除錯器會失敗, 大多數的原因都是權限或設定上有問題。

在微軟官方的知識庫 (Knowledge Base) 中, 有一篇文章列出了使用 Visual Studio 啓動 ASP.NET 除錯器時, 經常會看到的錯誤訊息以及解決方法：

```
http://support.microsoft.com/kb/306172/en-us
```

說明中的處理方式, 部份亦適用於 Visual Studio 2005。另外, 在 ASP.NET 2.0 中, 預設 Web.config 中的除錯指令是關閉的：

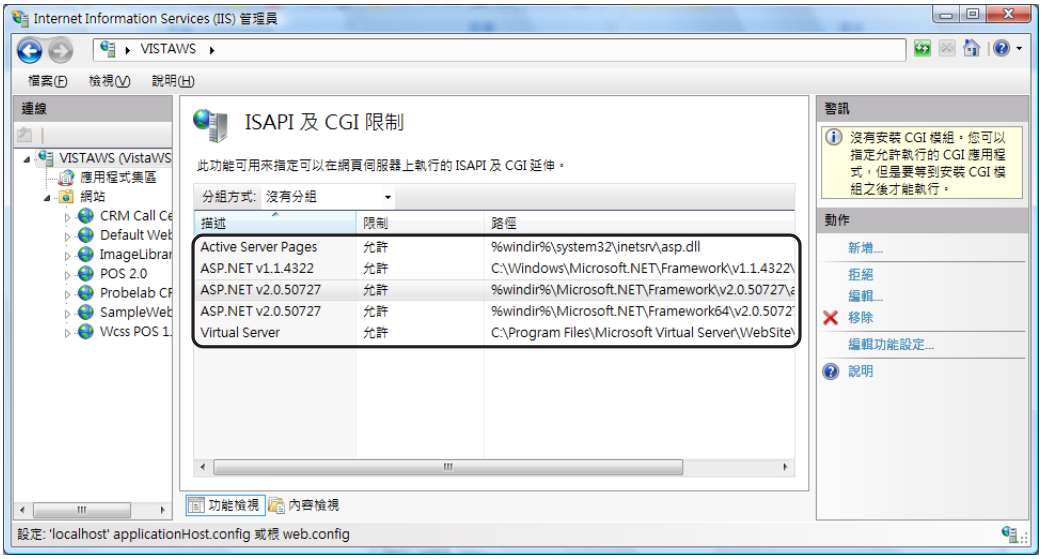
```
<configuration>
    <system.web>
        <compilation debug="false" />
    </system.web>
</configuration>
```

若要除錯這個專案, 則必須要將除錯指令打開, 亦即將 debug 的參數設爲 true, 如此除錯器才會生效。

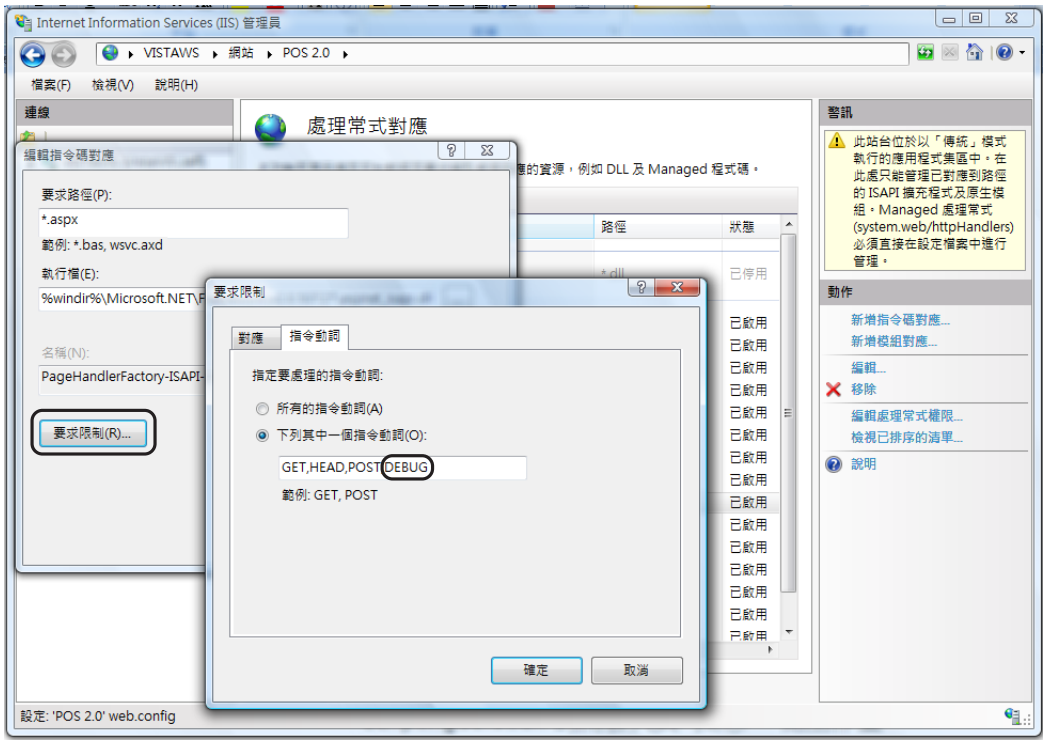
解決方案

若發生 Visual Studio 無法啓用 ASP.NET 除錯器時, 有幾個比較簡單的處理方法：

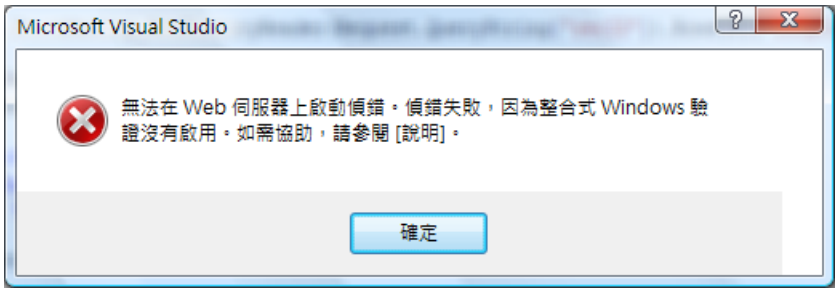
1. 重新建立 ASP.NET 在 IIS 上的 mapping, 這可以透過執行 **aspnet_regiis -i** 來做到 (aspnet_regiis.exe 的位置在 WINDIR\Microsoft.NET\Framework\版本號碼\的目錄中, 其中 WINDIR 表示 Windows 的安裝目錄, 而版本號碼會對應不同的 ASP.NET 版本, 例如 2.0 的版本號碼爲 2.0.50727) 。
2. 若是 Windows Vista (IIS 7.0)或 Windows Server 2003 (IIS 6.0), 則應檢查 ASP.NET 是否已被授權執行。請參考 Q2。



3. 檢查 IIS 的 ASP.NET 允許 DEBUG 動詞, 否則除錯器會失效。



4. 在 Windows Vista (IIS 7.0)中, 預設是不啓用 Windows 整合驗證模式, 但若不啓用, 除錯器會出現錯誤訊息：



故必須要啓用。

5. 如果以上都無法解決, 那就試著以修復方式安裝 Visual Studio 看看能否解決, 並且檢查電腦是否中毒或被植入惡意程式 (如木馬或殭屍程式) 或其他可能的影響因素。因爲每個人的環境都不同, 筆者的方法不一定能適用於所有的電腦。

經驗談

通常 Visual Studio 除錯器出錯的原因, 比例最高的仍然是 IIS 的設定有遺漏, 或是權限的問題, 若將這些問題排除的話, 除錯器會出現錯誤的機率其實不高, 筆者使用 Visual Studio 一段時間, 很少出現除錯器有問題的狀況, 除非專案檔沒有處理好 (這可以用另開專案, 然後將原專案檔移入新專案方式解決) , 或者是電腦本身不穩定 (中毒或安裝一堆怪程式, 讓電腦的設定亂掉了) , 最簡單的解決方法就是重灌 , 但如果問題不嚴重, 能不重灌就不重灌。

相關問題

- Q02：如何在 Windows Vista 上啓用偵錯功能

Q9 我要如何才能看到完整的錯誤訊息？

適用範圍：V ASP.NET 1.0 V ASP.NET 1.1 V ASP.NET 2.0 V ASP.NET 3.5

問題

我開發了一個小型的 Web 應用程式, 並且放到了網站上去跑, 但是會出現類似這樣的錯誤訊息：

```
[IndexOutOfRangeException: 索引在陣列的界限之外。]
   Part2_Default.Page_Load(Object sender, EventArgs e) +158
   System.Web.Util.CalliHelper.EventArgFunctionCaller(IntPtr fp, Object o, Object t, EventArgs e) +25
   System.Web.Util.CalliEventHandlerDelegateProxy.Callback(Object sender, EventArgs e) +48
   System.Web.UI.Control.OnLoad(EventArgs e) +133
   System.Web.UI.Control.LoadRecursive() +66
   System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint) +2604
```

雖然有說明錯誤原因, 但是我想要看是程式碼的哪裡有問題, 卻沒有出現這樣的資料, 我要如何做才可以看到出問題的程式碼位置？

問題說明

通常在執行 Web 應用程式的維護時, 我們會想要知道程式的錯誤在哪裡, 這樣在除錯與修補時的處理會比較快一些, 但是一般來說, 我們不會在正式環境中使用除錯程式碼, 主要原因之一是除錯的程式碼較大, 載入時間以及佔用記憶體的量會比較多, 但優點是可以立即反應出程式的錯誤位置, 以及可搭配除錯器來除錯。

以 Debug 模式編譯的程式碼, 都會額外產生一個記錄偵錯符號 (debug symbol) 的檔案, 以 PDB 爲副檔名, 這個檔案中存放所有專案中程式碼以及必要的系統程式的符號, 通常大小會比可執行檔要大, 所以需要的載入時間也會比較多, 但若以 Release 編譯, 則不會有這個檔案, 相對的, 執行碼會經過最佳化, 因此速度會比較快, 但也無法提供太多的錯誤資訊, 尤其是當錯誤是發生在較長程式碼的函式或類別時, 要找出錯誤的位置相對來說就會比較困難。

因此, 若想要輸出較完整的錯誤訊息, 則要以 Debug 的方式來建置或編譯專案才行。

解決方案

1. ASP.NET 1.x: 將專案以 Debug 模式編譯即可。
2. ASP.NET 2.0: 修改專案中的 Web.config 檔案, 啓用 debug 模式即可, 但若是有引用到元件, 則該元件也必須要以 Debug 模式編譯, 否則程式的位置輸出就只會到 ASP.NET 程式爲止, 不會輸出元件中錯誤的程式位置。

經驗談

在已部署的應用程式中啓用除錯，其實是相當常見的，尤其是專案剛部署到正式環境 (Production Environment) 試營運的初期, 以及人手不足讓測試工作需要交給客戶的情況下, 就需要在應用程式上線的狀況下啓用除錯, 影響其實不會很大, 缺點是除錯模式的程式碼未經最佳化, 執行起來會稍微慢一些, 不過在 CPU 運算能量高的伺服器上, 這一點點的效能損耗是可以接受的。

不過長期來看, 仍然應該要使用 Release 模式來建構並部署到正式環境中, 並且在程式碼中使用 Trace 來輸出資料, 讓開發人員可以利用夾擠方法來偵錯, 或者是輸出參數資料, 讓開發人員得以由參數資料來研判問題的發生地點, 這樣才是一個具高品質 Web 應用程式應該有的作法。

Q10 預先編譯 (Pre-compile) 眞的可以提昇效率嗎？如何部署這樣的 ASP.NET 程式？

適用範圍：V ASP.NET 2.0　V ASP.NET 3.5

問題

以往用 Visual Studio .NET 2003 開發 ASP.NET 應用程式, 建置好的專案都是一個 DLL 檔案, 但到了 Visual Studio 2005 時, 建置以後卻沒有產生 DLL 檔案, 而是需要連同原始檔案部署出去, 這樣會有程式碼被竊取的風險, 聽說 ASP.NET 2.0 有一個預先編譯的功能, 請問這個功能是否可以保護程式碼？效率是否有比較好？那麼又要如何部署預先編譯的專案？

問題說明

預先編譯 (Pre-compilation) 是 ASP.NET 2.0 才有的功能, 在 ASP.NET 2.0 的架構下, 整個程式碼模型有了大幅度的改變, 以往在 ASP.NET 1.x 的編譯法已經不再適用, 因此 ASP.NET 2.0 修改了編譯的方法, 讓開發人員在部署專案時, 可以先針對原始程式碼進行先期的編譯工作, 然後再部署到伺服器上。

ASP.NET 2.0 統一了命名空間, 並且納入了 partial class 的模型, 無法像 ASP.NET 1.x 時代, 可以直接設定命名空間或做網頁繼承 (Page-Inheritance) , 同時由於每個網頁都會被編譯成一個獨立的執行碼 (DLL) , 這個方法的好處是可以減少載入時間以及省下未用到的功能的記憶體, 但由於不再是單一個體的執行碼, 在後續維護上會有一些困難。另外, ASP.NET 在載入網頁時, 也會先對程式碼進行編譯, 以產生二進位 MSIL 執行碼, 再將 MSIL 轉換爲機器碼, 多了這一個編譯的步驟, 會產生額外的負載以及短暫時間的等待。

預先編譯的好處, 首先就是在伺服器上可以減少一個編譯網頁程式的動作, 直接部署 MSIL 執行碼, 伺服器省去了編譯的工作, 相對的速度也會加快些, 最顯而易見的好處, 就是它將原始碼編譯成 MSIL, 讓原始碼暴露的風險降低很多。

.NET Framework 2.0 提供了命令列的預先編譯工具：aspnet_compiler.exe, 開發人員可利用參數來控制它的編譯行爲。

aspnet_compiler.exe 語法：

```
aspnet_compiler [-?]
                [-m metabasePath | -v virtualPath [-p physicalPath]]
                [[-u] [-f] [-d] [-fixednames] targetDir]
                [-c]
                [-errorstack]
                [-nologo]
                [[-keyfile file | -keycontainer container ] [-aptca]
                 [-delaysign]]
```

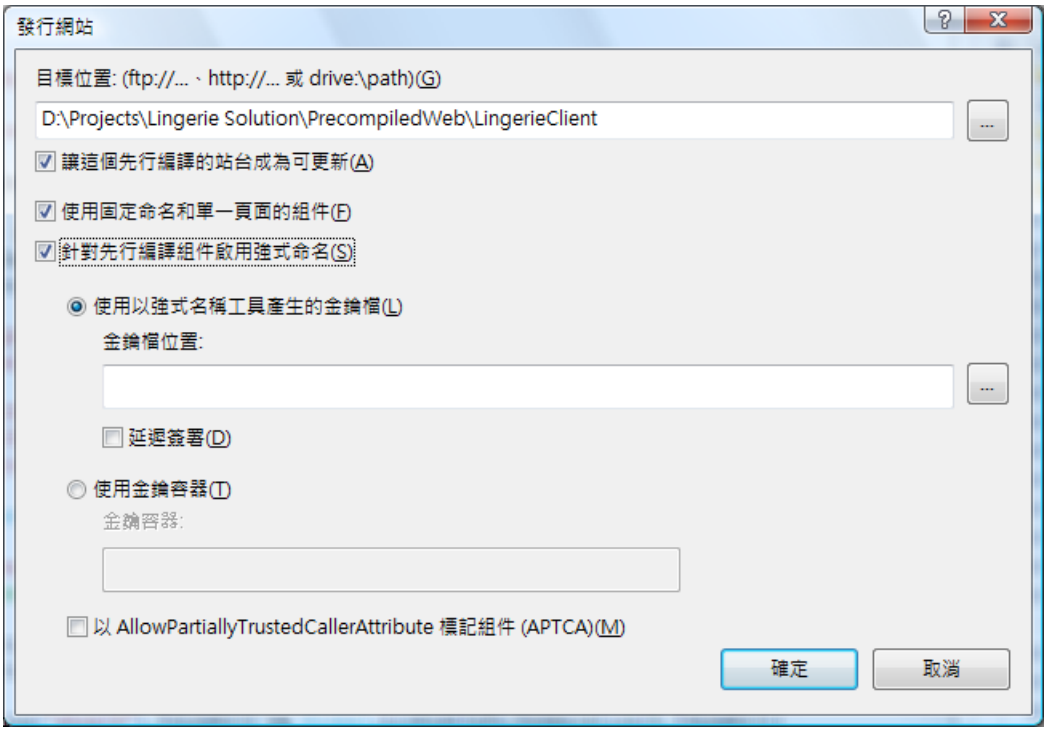
其中常用的參數有：

-v virtualPath	指定應用程式的 IIS 虛擬路徑。
-p physicalPath	指定應用程式的實體路徑。
-u	指定以更新模式來編譯應用程式。
-f	指定覆寫目的路徑指定的目錄與子目錄中的現有檔案。
-d	覆寫應用程式的 Web.config，以啓用偵錯功能。
targetDir	編譯後輸出的目錄路徑。
-c	指示全部重建。
-errorstack	若無法編譯，則輸出錯誤的堆疊。
-fixednames	固定編譯輸出的組件名稱。

若需要簽署預先編譯的組件，則需要使用下列參數：

-keyfile file	指定數位簽章檔案的位置。
-keycontainer container	指定具有數位簽章的容器。
-aptca	在組件上簽署數位簽章，並且套用部份信任 (Partial Trust) 層級的安全設定。
-delaysign	指定允許延遲組件的簽署工作，這個選項允許程式碼可在簽署作業完成前即可執行。

若對命令列工具不熟, 開發人員也可以使用 Visual Studio 2005 中的**發布網站 (Publish Web Site)** 功能, 由 Visual Studio 下參數執行 aspnet_compiler.exe 來編譯網站專案程式碼。



其中, **目前位置**即爲 -v 及-p 參數, **讓這個先行編譯的站台成爲可更新** 爲-u 參數； **使用固定命名和單一頁面的組件** 爲-fixednames 參數； **針對先行編譯組件啓用強式命名**的各選項, 分別代表了-keyfile、-delaysign, -keycontainer、以及-aptca 等參數。

小常識　要用哪種 ASP.NET 預先編譯模型？

ASP.NET 的預先編譯可分爲三種方式, 第一種是即地編譯 (In-place compilation) , 亦即由 aspnet_compiler 產生 HTTP Request, 讓應用程式可以藉由處理 HTTP Request 的程序來編譯；第二種是二進位編譯法 (Binary Compilation) , 亦即全部的原始碼都編譯成組件檔, 這樣應用程式就不必再重新編譯, 但若應用程式需要更新時, 就一定要將編譯後的輸出檔全部都部署上去才可以；第三種則是可更新式編譯 (Updatable Compilation) , 這種編譯法會只編譯部份應用程式碼到組件中, 而原始的網頁檔則不會被編譯, 讓應用程式可以更新。

通常一個已經完成的應用程式, 會採用二進位編譯法, 讓應用程式得以正常運行, 而在發展測試中的應用程式, 會採用即地編譯 (或根本就不先編譯) 的方法, 以利在開發過程中的隨時更新, 若是一個剛開發完成而可能預期會有變化的時候, 則可以考慮使用可更新式編譯法。

解決方案

預先編譯不但可以保護原始程式碼不外流, 並且可以在部署前就先編譯好, 讓伺服器不必再進行第二次的編譯, 會讓速度加快一些。

若要使用預先編譯的方式部署 ASP.NET 應用程式, 可以透過命令列的 aspnet_compiler.exe 以及 Visual Studio 2005 的發行網站功能來執行預先編譯工作, 然後編譯的輸出可以透過手動或 Windows Installer 方式來部署。

考生停看聽

本問題的討論內容, 可用來準備 Exam 70-528: TS: Microsoft .NET Framework 2.0 Web Client Development 的下列主題：

- 追蹤, 設定與部署應用程式
  - 利用發行網站工具執行 Web 應用程式的預先編譯工作。

Q11 要如何在同一台 IIS 中共用 ASP.NET 1.x 和 ASP.NET 2.0？

適用範圍：V ASP.NET 1.0 V ASP.NET 1.1 V ASP.NET 2.0

問題

我用 Visual Studio 2005 開發 Web 應用程式已經有一段時間了, 寫了不少的應用程式, 但現在有個問題, 公司有個用 ASP.NET 1.1 開發的應用程式, 要我去維護它, 那我要如何在 IIS 上面設定, 讓 IIS 可以讓這兩種平台的應用程式都可以跑呢？

問題說明

雖然.NET Framework 2.0 已經發表這麼久了, 仍然有一些以 ASP.NET 1.1 或 1.0 爲平台的應用程式沒有升級, 爲了要維護這些應用程式, 勢必要讓兩個版本的核心和應用程式都可以跑才行。

自.NET Framework 1.1 起, .NET Framework 不同版本之間的核心程式碼是可以並存的, 安裝時會裝在不同目錄, 而且各自的應用程式都可以持續使用, 這個能力稱爲 side-by-side (並存執行) , 除了可以允許不同版本的應用程式可執行外, 也允許應用程式決定要使用的.NET Framework 版本, 這在 Windows 應用程式中會比較常用到, 但在 Web 應用程式, 一切都得聽令於 IIS 中的 ASP.NET Runtime。

所幸微軟已經事先想到這個問題, 在.NET Framework 2.0 中, 附有一個加掛的設定工具, 可以讓系統管理員或開發人員設定網站所要使用的 ASP.NET 的版本, 而這個設定工具在安裝.NET Framework 2.0 時就會安裝進去, 若是在安裝.NET Framework 2.0 之後才安裝 IIS 的, 也只要利用 **aspnet_regiis -i** 來建立 ASP.NET 的設定, 即可使用這個工具。

當 IIS 的電腦有安裝.NET Framework 1.1 與 2.0 時, ASP.NET 版本即有 **1.1.4322** (ASP.NET 1.1) 以及 **2.0.50727** (ASP.NET 2.0) 兩個選項, 只要選取網站所需要的.NET Framework 版本, 並且按下**套用**或**確定**儲存, 該網站即可使用所設定版本的 ASP.NET Runtime 來處理應用程式的工作。

但請注意, ASP.NET 的 side-by-side 只針對網站或應用程式等級的虛擬目錄, 無法讓同一個網站中同時執行不同版本的.NET Framework。

解決方案

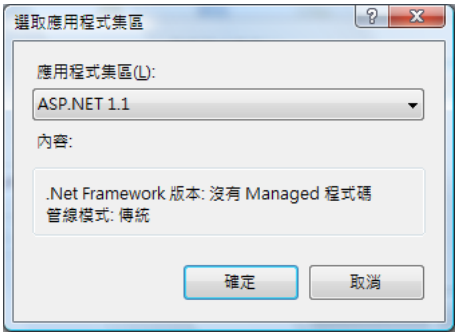
在 IIS 中設定網站所使用的 ASP.NET 版本, 並且儲存設定即可, 若沒有看到所需.NET Framework 版本, 則請安裝所需版本的 .NET Framework 可轉散佈套件。

小常識 如何在 IIS 7.0 中設定 ASP.NET 1.1 與 ASP.NET 2.0 並存？

由於 IIS 7.0 已經內建有.NET Framework 2.0, 所以直接由**程式與功能**來安裝 ASP.NET 2.0 即可, 但是若想要在 IIS 7.0 中設定讓 ASP.NET 1.1 與 2.0 並存, 則建議在安裝 IIS 7.0 之前, 先行安裝好.NET Framework 1.1 以及.NET Framework 1.1 Service Pack 1, 然後再安裝 IIS 7.0, 並且要安裝 IIS 6.0 的管理相容性以及 IIS Metabase 等, 讓.NET Framework 1.1 的 ASP.NET 安裝工具可以在 IIS 7.0 中建立關聯與資訊 (必要的元件要打勾) 。

接著在.NET Framework 1.1 的安裝目錄中, 下 **aspnet_regiis -i -enable** 將 ASP.NET 1.1 安裝到 IIS 7.0 中 (安裝副檔名對應, 以及 ASP.NET 1.1 專屬的應用程式集區) , 這樣 ASP.NET 1.1 就可以在 IIS 7.0 執行了。

最後, 將需要使用 ASP.NET 1.1 的網站的應用程式集區設定爲 ASP.NET 1.1 專屬的集區。



設定完成後, ASP.NET 1.1 應用程式即可在 IIS 7.0 中正常執行。

Q12 如何將 ASP.NET AJAX 整合到現有的 ASP.NET 2.0 應用程式中？

適用範圍： V ASP.NET 2.0　V ASP.NET AJAX

問題

微軟發表了 ASP.NET 的 AJAX Extension Library, 我用它來寫了簡單的測試網站, 感覺很好用, 但是我是用新增 ASP.NET AJAX 網站的專案範本來做的, 如果我要將它和現有的網站整合在一起時, 我要怎麼做？

問題說明

ASP.NET AJAX 於 2007 年三月份在衆所期待下推出, 它讓 ASP.NET 的開發人員在很簡單的情況下就可以發展出 Ajax-Enabled 的應用程式, 而且不必寫到任何一行 JavaScript 程式碼, 也不必用到 XmlHttp 物件, 是功能相當強大的 AJAX Extension 模組, 就連 Silverlight 也要透過 ASP.NET AJAX 才能和伺服器溝通。

在 ASP.NET AJAX Extension 發行時, 也附帶了一個 Visual Studio 2005 專用的 ASP.NET AJAX 專案範本, 若是要發展新的應用程式時, 直接由範本新增即可, 但是對於現有的應用程式來說, 豈不是要流口水？…當然不, ASP.NET AJAX 只要修改一下 Web.config 的設定, 就可以在現有的應用程式中使用 ASP.NET AJAX 提供的功能。

至於怎麼使用 ASP.NET AJAX…坊間有很多專書, 筆者在後面也會有幾個問題的篇幅來討論 ASP.NET AJAX 部份功能的使用 ☺。

解決方案

在 Web.config 中加入 ASP.NET AJAX Extension 所需要的設定資料, 並且在伺服器上安裝 ASP.NET AJAX Extension 即可。

首先, 爲了要讓 ASP.NET AJAX 的設定生效, 所以需要在 Web.config 中加入一些<configSections>設定區段, 如程式 1 所列。

程式 1 ：ASP.NET AJAX 所需要的 Configuration Section 擴充指令。

```
<configSections>
  <sectionGroup name="system.web.extensions"
    type="System.Web.Configuration.SystemWebExtensionsSectionGroup,
    System.Web.Extensions, Version=1.0.61025.0, Culture=neutral,
    PublicKeyToken=31bf3856ad364e35">
    <sectionGroup name="scripting"
      type="System.Web.Configuration.ScriptingSectionGroup,
      System.Web.Extensions, Version=1.0.61025.0, Culture=neutral,
      PublicKeyToken=31bf3856ad364e35">
        <section name="scriptResourceHandler"
          type="System.Web.Configuration.ScriptingScriptResourceHandlerSection,
          System.Web.Extensions, Version=1.0.61025.0,
          Culture=neutral, PublicKeyToken=31bf3856ad364e35"
          requirePermission="false"
          allowDefinition="MachineToApplication"/>
        <sectionGroup name="webServices"
          type="System.Web.Configuration.ScriptingWebServicesSectionGroup,
          System.Web.Extensions, Version=1.0.61025.0,
          Culture=neutral, PublicKeyToken=31bf3856ad364e35">
        <section name="jsonSerialization"
          type="System.Web.Configuration.ScriptingJsonSerializationSection,
            System.Web.Extensions, Version=1.0.61025.0,
            Culture=neutral, PublicKeyToken=31bf3856ad364e35"
            requirePermission="false" allowDefinition="Everywhere" />
        <section name="profileService"
          type="System.Web.Configuration.ScriptingProfileServiceSection,
            System.Web.Extensions, Version=1.0.61025.0,
```

```
            Culture=neutral, PublicKeyToken=31bf3856ad364e35"
            requirePermission="false"
            allowDefinition="MachineToApplication" />
        <section name="authenticationService"
          type="System.Web.Configuration.ScriptingAuthenticationServiceSection,
            System.Web.Extensions, Version=1.0.61025.0,
            Culture=neutral, PublicKeyToken=31bf3856ad364e35"
            requirePermission="false"
            allowDefinition="MachineToApplication" />
      </sectionGroup>
    </sectionGroup>
  </sectionGroup>
</configSections>
```

這些設定必須要放在<appSettings>之前, 否則會發生錯誤。

接下來, 要修改<system.web>中的設定, 重點有三個：

1. 爲了要讓<asp:ScriptManager>可以使用, 必須要允許 ScriptManager 所在的元件可以使用 asp 這個前置名稱, 因此要在<system.web>的<pages>中加上程式 2 所列的設定：

程式 2：將 asp 前置名稱設定關聯至 System.Web.Extensions.dll

```
<pages>
  <controls>
    <add tagPrefix="asp" namespace="System.Web.UI"
        assembly="System.Web.Extensions, Version=1.0.61025.0,
        Culture=neutral, PublicKeyToken=31bf3856ad364e35"/>
  </controls>
</pages>
```

2. 讓 Web 應用程式使用 System.Web.Extension.dll 的功能, 所以要在<system.web>的<compilation>中加入設定, 如程式 3。

程式 3：加入 System.Web.Extension 的參考

```
<compilation>
  <assemblies>
  <add assembly="System.Web.Extensions, Version=1.0.61025.0,
    Culture=neutral, PublicKeyToken=31bf3856ad364e35"/>
  </assemblies>
</compilation>
```

3. 讓 ASP.NET AJAX 可以處理由 scripting 發出的 request, 像是 asmx 或是一些資源要求指令等, 所以要在<system.web>的<httpHandlers>中加入設定, 如程式 4。

程式 4：加入 HTTP Handler 的設定, 讓 ASP.NET AJAX 能處理特定的 Scripting Request。

```
<httpHandlers>
  <remove verb="*" path="*.asmx"/>
  <add verb="*" path="*.asmx" validate="false"
    type="System.Web.Script.Services.ScriptHandlerFactory,
      System.Web.Extensions, Version=1.0.61025.0, Culture=neutral,
      PublicKeyToken=31bf3856ad364e35"/>
  <add verb="*" path="*_AppService.axd" validate="false"
    type="System.Web.Script.Services.ScriptHandlerFactory,
      System.Web.Extensions, Version=1.0.61025.0, Culture=neutral,
      PublicKeyToken=31bf3856ad364e35"/>
  <add verb="GET, HEAD" path="ScriptResource.axd" validate="false"
    type="System.Web.Handlers.ScriptResourceHandler,
      System.Web.Extensions, Version=1.0.61025.0, Culture=neutral,
      PublicKeyToken=31bf3856ad364e35"/>
</httpHandlers>
```

4. 加入 ASP.NET AJAX 的元件爲 HTTP 處理模組, 要在<system.web>中的 <httpModules>中加入設定, 如程式 5。

程式 5：設定 ASP.NET AJAX 的元件爲 HTTP Module

```
<httpModules>
  <add name="ScriptModule" type="System.Web.Handlers.ScriptModule,
    System.Web.Extensions, Version=1.0.61025.0, Culture=neutral,
    PublicKeyToken=31bf3856ad364e35"/>
</httpModules>
```

<system.web>必要的設定至此就全部設定完成。

接著, 加入 ASP.NET AJAX 的設定, 這些設定是由開發人員需要利用 ASP.NET AJAX 的特殊功能時設定之用, 在<system.web>的下方, 加入程式 6 的設定。

程式 6：ASP.NET AJAX 設定資訊。

```
<system.web.extensions>
  <scripting>
    <webServices>
    <!-- Uncomment this line to customize maxJsonLength and add a
      custom converter -->
      <!--
      <jsonSerialization maxJsonLength="500">
        <converters>
          <add name="ConvertMe"
            type="Acme.SubAcme.ConvertMeTypeConverter"/>
        </converters>
      </jsonSerialization>
      -->
```

```
      <!-- Uncomment this line to enable the authentication service.
           Include requireSSL="true" if appropriate. -->
      <!--
      <authenticationService enabled="true"
        requireSSL = "true|false"/>
      -->
      <!-- Uncomment these lines to enable the profile service. To
           allow profile properties to be retrieved
           and modified in ASP.NET AJAX applications, you need to add
           each property name to the readAccessProperties and
           writeAccessProperties attributes. -->
      <!--
      <profileService enabled="true"
        readAccessProperties="propertyname1, propertyname2"
        writeAccessProperties="propertyname1, propertyname2" />
      -->
    </webServices>
    <!--
    <scriptResourceHandler enableCompression="true"
      enableCaching="true" />
    -->
  </scripting>
</system.web.extensions>
```

最後, 若要在 IIS 7.0 中設定 ASP.NET AJAX, 則需要在<system.webServer>中, 加入 IIS 7.0 對應的設定資料, 如程式 7。

程式 7：IIS 7.0 需要的 ASP.NET AJAX 設定資料

```
<validation validateIntegratedModeConfiguration="false" />
<modules>
  <add name="ScriptModule"
    preCondition="integratedMode"
    type="System.Web.Handlers.ScriptModule, System.Web.Extensions,
      Version=1.0.61025.0, Culture=neutral,
      PublicKeyToken=31bf3856ad364e35"/>
</modules>
<handlers>
  <remove name="WebServiceHandlerFactory-ISAPI-2.0"/>
  <add name="ScriptHandlerFactory" verb="*" path="*.asmx"
    preCondition="integratedMode"
    type="System.Web.Script.Services.ScriptHandlerFactory,
      System.Web.Extensions, Version=1.0.61025.0, Culture=neutral,
      PublicKeyToken=31bf3856ad364e35"/>
  <add name="ScriptHandlerFactoryAppServices" verb="*"
    path="*_AppService.axd" preCondition="integratedMode"
    type="System.Web.Script.Services.ScriptHandlerFactory,
      System.Web.Extensions, Version=1.0.61025.0, Culture=neutral,
      PublicKeyToken=31bf3856ad364e35"/>
  <add name="ScriptResource" preCondition="integratedMode"
    verb="GET, HEAD" path="ScriptResource.axd"
    type="System.Web.Handlers.ScriptResourceHandler,
      System.Web.Extensions, Version=1.0.61025.0, Culture=neutral,
      PublicKeyToken=31bf3856ad364e35" />
</handlers>
```

設定完成後, 記得存檔, 然後重新瀏覽網站, 設定值就會生效, 至此, 現有的應用程式即支援 ASP.NET AJAX, 開發人員即可在專案中使用 ASP.NET AJAX 的各項功能。

在此, 筆者提供一個簡單又快速的方法：

1. 用 ASP.NET AJAX 專案範本建立一個新專案。

2. 打開新專案的 Web.config, 以及現有專案的 Web.config。

3. 複製必要的資料過去 (記得不要貼錯邊了) 。

4. 重新瀏覽網站, 讓設定生效。

以上的設定內容, 在 ASP.NET AJAX 的線上文件中可以找的到, 讀者也可以自 http://ajax.asp.net 中下載線上文件到本機電腦上直接瀏覽。

Q13 如何將 ASP.NET AJAX 1.0 的應用程式升級到 ASP.NET 3.5？

適用範圍： VASP.NET 2.0

問題

我使用了 ASP.NET 2.0 及 ASP.NET AJAX 1.0 來開發 AJAX 的網站應用程式, 最近爲了因應公司將開發技術升級到.NET Framework 3.5 (ASP.NET 3.5) , 主管要求要升級我的網站到 ASP.NET 3.5, 但在 ASP.NET 3.5 中, 已經內建了 ASP.NET AJAX, 那我要怎麼處理升級的問題？

問題說明

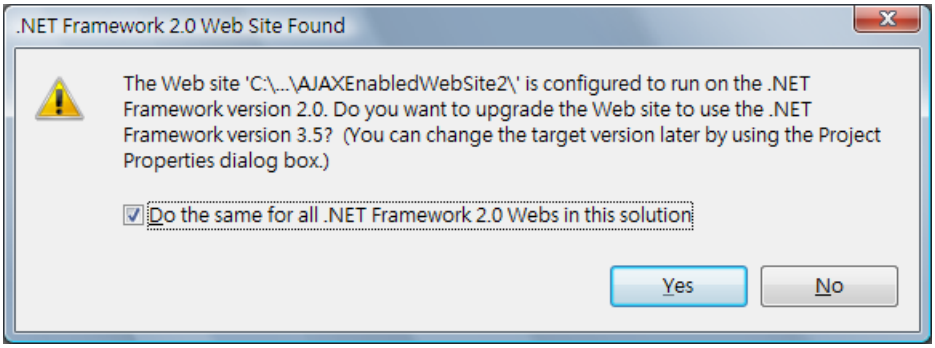
Visual Studio 2008 英文版本已經在 2007 年 11 月下旬時發表, 與它一起發表的還有 Microsoft .NET Framework 3.5, 這個版本的更新幅度比起 3.0 版本要大, 其中主要的地方就是語言的升級 (LINQ 的支援) 以及資料存取的平台 (ADO.NET Entity Framework) , ASP.NET 3.5 雖然也有更新, 但幅度就沒有這麼大。

即便如此, ASP.NET AJAX 由於要和.NET Framework 的版號同步, 因此也被升級到 3.5, 並內建在 ASP.NET 3.5 中, 這也就是說, ASP.NET AJAX 1.0 可能無法在 ASP.NET 3.5 中使用了, 那麼在原有的 ASP.NET AJAX 1.0 發展的應用程式, 是否就不能升級到 ASP.NET 3.5？

答案當然是否定的, 但是要做一些設定便是。若是有 Visual Studio 2008 時, 升級的工作將會更加簡單, 同樣的, 也可以讓 Visual Studio 2008 使用 ASP.NET AJAX 1.0 來開發網站 (但這不是本篇文章的重點, 筆者會在文末提供參考資料) 。

解決方案

如果是使用 Visual Studio 2008 時, 若開啓支援 ASP.NET 2.0 的網站, 它會彈出訊息視窗, 詢問是否要升級。



如果選擇升級的話, Visual Studio 2008 會自動把 Web.config 做修訂, 將版本 2.0 的組件都升級到 3.5, 包含原有的 ASP.NET 組件, 以及 ASP.NET AJAX 的組件。

如果應用程式是使用 Web Application Project 來開發的話, 則可能需要檢查一下網站的參考是否仍保留了 ASP.NET AJAX 1.0 版本的組件, 如果有, 則需要做手動的更新。

接著, 下載支援.NET Framework 3.5 的 ASP.NET AJAX Control Toolkit, 並更新網站的 AJAX Control Toolkit 的參考 (將現有的 AJAX Control Toolkit 1.0 的參考移除, 再加入新的 AJAX Control Toolkit 3.5 的參考) , 1.0 版本的版號爲 1.0.10920.0, 支援 3.5 版的 AJAX Control Toolkit 的版本號碼爲 3.0.11119.9。

最後, 在 Web.config 中加上版本的繫結轉送 (binding redirect) 設定, 讓協力廠商的 AJAX 元件可以直接存取.NET 3.5 的 AJAX 程式碼：

```
<configuration>
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="System.Web.Extensions"
          publicKeyToken="31bf3856ad364e35"/>
```

```
        <bindingRedirect oldVersion="1.0.0.0-1.1.0.0"
          newVersion="3.5.0.0"/>
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="System.Web.Extensions.Design"
          publicKeyToken="31bf3856ad364e35"/>
        <bindingRedirect oldVersion="1.0.0.0-1.1.0.0"
          newVersion="3.5.0.0"/>
      </dependentAssembly>
    </assemblyBinding>
  </runtime>
</configuration>
```

到此, ASP.NET AJAX 1.0 的應用程式即告升級完成。

參考資料

讓 Visual Studio 2008 和 ASP.NET AJAX 1.0 互動的方法, 包含將 ASP.NET AJAX 1.0 升級, 以及讓 Visual Studio 2008 使用 ASP.NET AJAX 1.0 的方法, 可參考下列兩篇文章：

- Using Visual Studio 2008 to target ASP.NET AJAX 1.0

```
http://blogs.msdn.com/webdevtools/archive/2007/07/30/using-vs-2008-to-target-asp-net-ajax-1-0.aspx
```

- Upgrading ASP.NET AJAX 1.0 Web Site and Web Applications to .NET Framework 3.5

```
http://blogs.msdn.com/webdevtools/archive/2007/07/28/upgrading-asp-net-ajax-1-0-websites-and-web-applications-to-net-framework-3-5.aspx
```

相關問題

- Q12：如何在將 ASP.NET AJAX 整合到現有的 ASP.NET 2.0 應用程式中？