

PART

3

使用者介面篇

- Q14：如何在按下 Button 後, 彈出確認視窗或是資料輸入表單？
- Q15：為何在重新整理時, 都會彈出警告視窗？
- Q16：如何在 ASP.NET 程式中輸出用戶端指令碼？
- Q17：如何在程式中等待用戶端指令的執行結果？
- Q18：如何由遠端抓取需要登入帳戶才能存取的圖片, 並顯示在 Image 控制項？
- Q19：我要如何要求使用者依據指定的格式輸入電話之類的資料？
- Q20：我之前有寫好的用戶端驗證程式碼, 要如何整合到 ASP.NET 的驗證功能中？
- Q21：如何使用 ASP.NET Ajax 函式庫執行表單驗證？
- Q22：我有寫好的用戶端指令碼, 要如何跟 ASP.NET AJAX 函式庫整合在一起使用？
- Q23：如何使用用戶端指令碼動態產生用戶端的控制項？
- Q24：如何在伺服器端動態產生控制項？
- Q25：如何動態產生圖片？
- Q26：如何在網頁中讓使用者可以直接編修 HTML 格式的文件或內容？
- Q27：如何將 ASP.NET 2.0 的會員資料存在現有的資料庫中？
- Q28：如何動態切換 Master Page？
- Q29：我想要在 Content Page 中調整 Master Page 的內容, 請問我要如何做？
- Q30：如何讓 Menu 和 SiteMap 共用同一個 XML 資料？

Q14

如何在按下 Button 後, 彈出確認視窗或是資料輸入表單?

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

問題

我寫了一個資料輸入用的網頁, 並且在上面設計好了控制項和資料庫讀寫的方法了, 不過我想要在使用者按下確定的按鈕時, 出現一個確認視窗, 讓使用者可以做最後的確認動作, 請問我要如何做?

問題說明

在 Web 應用程式中, 有很多地方都會需要使用者的資料輸入, 以及在網頁上提供修改和刪除的功能, 像是一般的顧客資料輸入; 產品資料維護或是訂單處理等各式各樣的表單存在於 Web 應用程式中。

大多數的情況下, 使用者輸入資料後傳送到 Web Server 是不太需要再確認的, 但有些情況會例外:

- 刪除資料時。
- 資料的正確性較要求時。
- 需要給使用者有取消的機會時。

若以 ASP.NET 的程式發展方式, 通常都會在使用者送出資料後, 記錄在伺服器端, 然後顯示是否確定要儲存的訊息, 這是比較簡單的作法, 不過這個方法會有個問題, 就是多一次的 Round-trip (伺服器的來回傳輸), 在一個有很多人使用的網站應用程式上, 如果有 1000 個使用者, 可能就會產生 $1000 * n$ (n 為 Round-trip 的次數) 的網路流量, 假設一個 Round-trip 的傳輸量是 30K 的話, 至少就會有約 30MB 的資料傳輸量花在 Round-trip 上, 對於網站的頻寬會是個很大的影響因素。

就算對頻寬沒有太大要求, 對使用者的操作經驗也會是個比較大的影響, 因為每次的送出都是一個 Post Back, 這個動作會刷新網頁 (Page Refreshing), 讓畫面會出現閃動的情形, 會讓使用者認為處理的工作不是即時性的, 若處理的時間拉長的話, 那使用者的感受會更差 (認為應用程式很慢)。

所以現在較新的 Web 發展模式, 都強調盡可能的減少用戶端和伺服器的來回次數及資料量, 而且對於使用者感受的部份會比較要求, 像是即時回應 (Real-time response) 的能力, 以及快速反應的能力等, 當然有些要求需要配合系統、網路和硬體, 但是如果能夠在程式中就處理掉, 將可省下許多成本或時間。其中, Client Script (用戶端指令碼) 的應用就很重要。

Client Script 是內建在瀏覽器中, 用來操作 HTML 物件, 以及動態處理使用者介面與資料功能的程式, 它利用瀏覽器的 DOM (文件物件模型) 來對網頁進行控制, 包括物件的位移, 視窗的管理或是表單資料的處理等等。而目前最常用的 Client Script 語法為 JavaScript, 因此本書中用到 Client Script 的部份, 都會使用 JavaScript 來撰寫。

一般要在用戶端顯示一個訊息, 用戶端的指令碼可以這樣寫：

```
alert("訊息內容");
```

這個指令碼會彈出一個訊息方塊。

而要彈出確認視窗的話, 用戶端的指令碼可以這樣寫：

```
confirm("訊息內容");
```

這個指令碼會彈出一個訊息方塊, 方塊中會有確定和取消兩個按鈕, 由使用者按下的按鈕來決定 confirm() 的回傳值。

按鈕類型	回傳值
確 定	true
取 消	false

因此, 若要確認一個動作, 可以這樣寫：

```
if (confirm("..."))
{
    ... // 確定執行的程式碼
}
else
{
    ... // 取消的處理程式碼
}
```

ASP.NET 的按鈕在輸出到用戶端時, 都會輸出成一個<input type="submit" /> (提交表單) 的 HTML 控制項, 當按下按鈕的時候, 瀏覽器會送出表單, 這個表單包含了 ASP.NET 中的資料, 像是 ViewState、控制項資料以及表單中的控制項 ID 等等, 若要停止送出表單的動作, 可以利用 `event.returnValue = false;` 的方式, 來取消送出表單的動作, 所以指令碼可以這樣寫：

```
<input type="submit" value=" 確定 "
onclick="event.returnValue=confirm( '確定要送出 ?' );" />
```

或是

```
<input type="submit" value=" 確定 "
onclick="return confirm( '確定要送出 ?' );" />
```

這樣就可以在使用者取消的時候停止送出表單的動作。

若想要在 Server 端加上這個指令碼, 可以透過下列的方式：

```
// for ASP.NET 1.x
cmdSubmit.Attributes.Add(
    "onclick", "return confirm( '確定要送出 ?' );");

// for ASP.NET 2.0
cmdSubmit.OnClientClick =
    "return confirm( '確定要送出 ?' );";
```

除了可以彈出訊息視窗外, DOM 也提供了一個可以開啓視窗來瀏覽其他網頁的方法—`window.open()`。它可以開一個新的視窗, 並且也可以讓開發人員自行設定視窗的屬性以及是否要和母視窗 (`opener`) 做關聯。`window.open()`的參數格式爲：

```
window.open(strURL, strName, strFeatures);
```

參數說明如下：

- `strURL`：要開啓的網頁路徑 (可以是相對或絕對路徑)。
- `strName`：這個子視窗的名稱, 或控制指令。常用的控制指令有：

指令	說明
<code>_blank</code>	開啓新的, 且沒有命名的視窗。
<code>_parent</code>	指定的 URL 會被載入到目前的頁框 (Frame) 的父頁框中, 如果這個頁框沒有父頁框, 那麼動作就會和 <code>_self</code> 相同。
<code>_search</code>	將指定的 URL 用瀏覽器的搜尋工作窗格來顯示, 這個功能在 IE 7.0 不支援 (由右上角的搜尋輸入方塊取代)。
<code>_self</code>	將指定的 URL 開啓在目前的視窗中。
<code>_top</code>	如果頁面中有頁框存在時, 會用指定的 URL 取代掉頁框的內容, 如果沒有頁框定義時, 動作就會 <code>_self</code> 相同。

- **strFeatures**：設定視窗的外觀以及功能, 一次可以設定多個功能選項, 每個選項間以逗號分開, 常用的選項有：

功能選項	可用值	說明
menubar	yes, no, 0, 1	指定是否顯示功能表列。
toolbar	yes, no, 0, 1	指定是否顯示工具列。
status	yes, no, 0, 1	指定是否顯示狀態列。
width	數 值	指定視窗寬度。
height	數 值	指定視窗高度。
scrollbars	yes, no, 0, 1	指定是否要顯示捲軸。
resizable	yes, no, 0, 1	指定是否可變更視窗大小。
left	數 值	指定視窗左上角的 x 軸位置。
top	數 值	指定視窗左上角的 y 軸位置。
location	yes, no, 0, 1	指定是否顯示網址列 (IE 7 則為瀏覽列)。

備註：表中的 0 等同於 no, 1 等同於 yes。

例如, 若想要打開一個沒有功能表、狀態列、工具列、捲軸、並且不可調整大小, 以及寬 500 像素, 高 500 像素的視窗, 可以利用下列的指令來做到：

```
window.open("CustomerForm.aspx", "CustomerForm",
    "menubar=no, toolbar=no, status=no, scrollbars=no,
    resizable=no, height=500px, width=500px");
```

有了這些知識, 就可以解決這個問題了。

解決方案

利用用戶端指令碼的方式, 顯示確認視窗, 例如：

```
// 在程式碼端加入指令碼。

public void Page_Load(object sender, EventArgs e)
{
    cmdSubmit.OnClientClick = "return confirm( '確定要送出 ?' );";
}

<!-- 直接在 HTML 中加入指令碼。 -->
<asp:Button ID="cmdSubmit" runat="server"
    OnClientClick=" return confirm( '確定要送出?' );"
    OnClick="cmdSubmit_Click" />
```

若想要顯示新的視窗, 則可以利用下列的方法：

```
// 在程式碼端加入指令碼。

public void Page_Load(object sender, EventArgs e)
{
    cmdSubmit.OnClientClick = @"window.open( 'Form.aspx' , 'Form' ,
        'toolbar=no, status=no, menubar=no, height=400px, width=500px,
        scrollbars=yes, resizable=no' );";
}
```

考生停看聽

本問題中討論的內容, 可用來準備 Exam 70-528: TS: Microsoft .NET Framework 2.0 Web Client Development 的主題：

- Web 網頁程式設計
 - 使用 Client Script 來避免 Round-trip。

相關問題

- Q16：如何在 ASP.NET 程式中輸出用戶端指令碼？

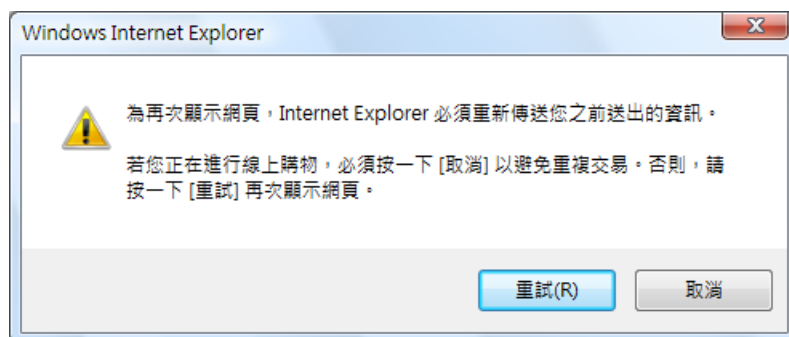
Q15

為何在重新整理時，都會彈出警告視窗？

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

❓ 問題

我用 ASP.NET 2.0 設計 Web 應用程式, 最近我發現一個問題, 就是使用者在工作完成後, 如果按下重新整理的按鈕 (或功能鍵 **F5**), 都會出現這個訊息：



這時若按重試, 則先前的資料寫入工作又會再做一次, 變成會有重覆資料寫入資料庫, 請問我要如何做, 才能夠避免這個問題？

❗ 問題說明

這是一個經常在發展 Web 應用程式時出現的問題, 主要的原因是發生在瀏覽器對 HTTP POST 的重試行為, 因此在 PHP 或 JSP 的應用程式中也會發生這樣的狀況, 尤其是以流程導向 (Process-Oriented) 作業的應用程式更容易發生這樣的問題。

ASP.NET 是以表單做為伺服器端和用戶端交換資料的基礎方法, 所以 ASP.NET 需要所有的控制項都要在表單的範圍內, 這樣才可以存取控制項中的資料。平常在 ASP.NET 的網頁中都會看到這樣的一個宣告：

```
<form runat="server">  
...  
</form>
```

這行宣告會在輸出到用戶端時, 轉換成一個標準的 HTML FORM 格式, 例如：

```
<form name="aspnetForm" method="post" action="Form.aspx"  
  id="aspnetForm">  
...  
</form>
```

而 ASP.NET 可以透過表單的 HTTP POST 的資料, 來存取與控制控制項的資料與屬性, 同時, 也可以判斷是否是 Post Back, `Page.IsPostBack` 會根據 HTTP POST 的資料來決定是否為 Round-trip, 如果是, `Page.IsPostBack` 會設為 true, 反之則是第一次瀏覽, 而 `Page.IsPostBack` 會設為 false。

瀏覽器的「重新整理」問題導因於表單資料的重新傳送的功能, 在早期的網路由於頻寬和伺服器處理速度較慢的關係, 有時候伺服器會沒有回應或是沒有回傳結果, 此時使用者可以透過重新整理的功能, 再次傳送表單資料給伺服器處理, 然而在現今網路和伺服器功能大大提升的情況下, 重新整理功能反而會讓資料出現重覆現象, 例如顧客資料輸入並且處理完成後, 若按了重新整理的重試, 則顧客資料會又再寫入一次, 變成有重覆的資料出現, 若是庫存, 則會造成虛增現象, 因此開發人員需要避免這個狀況發生, 確保資料的正確性。

解決方案

這個問題的最佳處理方式, 也是最簡單的處理方式, 就是強迫使用者的瀏覽器捨棄掉重新整理的資訊, 而最快速的方法, 就是強迫瀏覽器重新導向到相同的頁面, 例如：

```
// Server-Side  
Response.Redirect("Form.aspx");  
  
// Client-Side (Scripting)  
window.location.href="Form.aspx";
```

這樣就可以讓瀏覽器捨棄掉前面的表單資訊, 也就不會出現重新整理的訊息了。

考生停看聽

本問題中討論的內容, 可用來準備 Exam 70-528: TS: Microsoft .NET Framework 2.0 Web Client Development 的主題：

- Web 應用程式設計
 - 利用 `Page.IsPostBack` 來預防在 Round-trip 中不必要的處理工作。
 - 使用 `Client Script` 來減少或預防 Round-trip。

Q16

如何在 ASP.NET 程式中輸出用戶端指令碼？

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

問題

1. 我寫了一些用戶端的指令函數, 但我想要在按下按鈕, 以及網頁載入時呼叫這些函數, 請問我要如何做？
2. 我想要在用戶端依照不同的使用者條件, 而做不同的互動流程 (例如不同的問卷填答流程), 請問要如何實作？

問題說明

ASP.NET 雖然提供了許多方便的控制項, 供開發人員選用, 不過由於它是基於 HTTP 通訊以及表單處理的方式, 讓開發人員得以用接近 Windows Forms 開發模型的方法來發展 Web 應用程式, 但本質上它還是脫離不了 HTTP POST 以及 Client-Server 通訊這二個必要的步驟。

平常瀏覽網頁的時候, 由瀏覽器送出各式各樣的 HTTP 訊息 (可能是 HTTP GET 或 HTTP POST), 經伺服器端處理後產生結果回傳, 瀏覽器在接到結果後會將目前的畫面做更新, 這個工作稱為全頁面刷新 (Full Page Refresh), 而這個時候就算再快的電腦, 也還是會有閃動的狀況, 只是速度的快慢而已, 這無法避免, 因為這就是 Web 的運作模式。

早期的瀏覽器廠商其實就注意到這個問題了, 因此才會有新的 HTML 規範出現 (當時 HTML 只有靜態功能), 逐漸的 HTML 開始可以動態化, 並且直接在瀏覽器中產生動態效果, Dynamic HTML 就此成形, 且一直沿用到今日。除了瀏覽器賦與的 HTML 即時繪製外, 能夠驅動這些 HTML 物件的, 就只有內建在瀏覽器中的 scripting language (指令碼語言), 也就是大家常說的 JavaScript 與 VBScript, 直到現在, 這兩種語言仍被大家所愛用, 其中 JavaScript 幾已獨霸於瀏覽器界, 而多數的 VBScript 轉戰到 Windows Scripting Host 來做指令化的系統管理工作。

因此,若想要在 ASP.NET 網頁上設計避免有閃動的 **Post-Back**, 以及在控制項中加入一些用戶端行為的工作, 就只能交給 **JavaScript** 來做, 所以網路上就有這麼一句話:「沒有 **JavaScript** 的 ASP.NET 是死的」。想要發展出具有高互動性的 ASP.NET 應用程式, **JavaScript** 可是一個不可忽略的基本功。

AJAX (**Asynchronous JavaScript and XML**) 則是最近相當火紅的 **Web 2.0** 技術之一, 也是一個成熟的 **JavaScript** 應用, 經由非同步的遠端呼叫以及通訊, 讓用戶端可以不必再經歷 **Post-Back** 以及畫面閃動, 而即時將資料繪製到網頁中, 這個技術稱為部份頁面刷新 (**Partial Page Refresh**), 在 ASP.NET **AJAX** 推出後, 更成為 ASP.NET 開發人員愛用的技術, 筆者會在隨後的數個問答中展示 ASP.NET **AJAX** 的活用技巧。

但在這之前, 我們可以先來了解一下 ASP.NET 如何處理用戶端指令碼的輸出工作。

ASP.NET 1.x 的時代, 用戶端指令碼其實並不受重視, 可以支援的方法少之又少, 最常用的就是三個輸出用戶端指令碼的方法:

<code>Page.RegisterClientScriptBlock()</code>	註冊並輸出用戶端指令碼。
<code>Page.RegisterStartupScript()</code>	註冊並輸出用戶端啟動指令碼。
<code>Page.RegisterOnSubmitStatement()</code>	註冊並輸出用戶端提交處理指令碼。

這三個方法其實說穿了, 它其實就只等於是簡單的 `<script>...</script>` 的包裝而已, 只是它的輸出位置會有些不同, `Page.RegisterStartupScript()` 會輸出到網頁的最後, 因此會被執行, 而 `Page.RegisterClientScriptBlock()` 會輸出到 `<head></head>` 之間的區塊, 而唯一比較不同的 `Page.RegisterOnSubmitStatement()` 則是輸出在表單中觸發 **Post-Back** (也就是提交表單) 時要處理的用戶端指令碼。如果本身就對 **DHTML** 以及 **scripting** 相當熟悉的開發人員, 這些方法會用到的機率其實近乎零, 因為只要用 `Response.Write()` 來輸出指令, 或者是應用一些可以完整輸出 (將給定的值完整的輸出到用戶端) 的控制項 (例如 **Literal** 控制項), 就可以自行輸出指令碼, 而不必藉由 **Page** 類別的函式代勞。

到了 ASP.NET 2.0, 用戶端指令碼的支援才受到重視, 改由 `ClientScriptManager` (`Page.ClientScript`) 來處理, 但其實本質也沒有太大變化, 不過由於 ASP.NET 2.0 多加入了 `Callback` 的能力, 所以 `ClientScriptManager` 也新增了一些函式來處理。不過以筆者自己的開發經驗來說, `ClientScriptManager` 所提供的函式仍然沒有太多的作用, 自行輸出指令碼其實也沒有太多的負擔, 而且還可以有更多的彈性, 而用戶端回呼能力 (`Callback`) 也將在 ASP.NET AJAX 推出後逐漸淡出。

而筆者自己使用的方法, 多是以 `Literal` 控制項為載具, 然後直接將指令碼設定到 `Literal` 控制項, 例如：

```
<!-- in HTML -->
<asp:Literal ID="ScriptSpace" runat="server"
    EnableViewState="false" />

// in code-behind class
this.ScriptSpace.Text =
    "<script> alert('Thanks for your cooperation.');";
```

這個方法可以讓開發人員自訂輸出的訊息以及一些必要程序的指令碼, 而不必透過 `RegisterClientScriptBlock()` 等函式, 也不必自己用 `Response.Write()` 一行一行的輸出 (多行的話就很不一樣了)。

另外, 若是想要在頁面中引用指令碼的話, 就不能使用這個方法, 一定要用 `ClientScript.RegisterClientScriptInclude()` 函式, 這個函式是 ASP.NET 2.0 才有的, 若是在 ASP.NET 1.x, 只能夠把必要的指令都放到 HTML 中事先宣告。

解決方案

先將想要執行的指令碼, 使用系統預設提供的輸出指令碼方法, 或是利用筆者的輸出方法, 將指令碼輸出到用戶端, 然後在要下指令的控制項的觸發事件 (例如 `Button` 的 `OnClientClick`), 設定呼叫用戶端指令碼的函式, 例如程式 1 所示。

程式1：將指令碼輸出到前端，並由按鈕呼叫執行。

```
<!-- in HTML -->
<asp:Button ID="cmdLoadList" runat="server"
    OnClientClick="return LoadList();"
    Text="填入資料到 ListBox" /><br />
<asp:ListBox ID="listData" runat="server" Rows="5" Width="400px" />
<asp:Literal ID="ScriptSpace" runat="server" />

// in code
protected void Page_Load(object sender, EventArgs e)
{
    string myScript = @"
<script>
var data = new Array(
    'data1', 'data2', 'data3', 'data4', 'data5',
    'data6', 'data7', 'data8', 'data9', 'data10' );

function LoadList()
{
    var listBox = document.getElementById('listData');

    for (var i=0; i<data.length; i++)
    {
        var opt = document.createElement('OPTION');
        opt.text = data[i];
        opt.value = data[i];
        listBox.options.add(opt);
    }
}
```

```
        return false;
    }
</script>
";
// 設定指令到 Literal 控制項中。
this.ScriptSpace.Text = myScript;
}
```

若想要避免畫面閃動, 就要利用 **JavaScript** 來填入資料, 如果是簡單的資料瀏覽, 可以先將資料清單輸出到用戶端, 然後透過 **JavaScript** 指令將資料填到表單的欄位中, 倘若資料量過大, 或者資料具有機密性, 不便輸出到用戶端時, 則可以思考使用 **AJAX** 能力來實作。

相關問題

- Q14：如何在按下 **Button** 後, 彈出確認視窗或是資料輸入表單？
- Q17：如何在程式中等待用戶端指令的執行結果？
- Q20：我之前有寫好的用戶端驗證程式碼, 要如何整合到 **ASP.NET** 的驗證功能中？
- Q23：如何使用用戶端指令碼動態產生控制項？

Q17

如何在程式中等待用戶端指令的執行結果？

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

我的程式有一個需求, 要在程式繼續進行下去之前, 先在用戶端取得一些資訊, 然後才能判斷接下來的動作, 請問我要如何做？

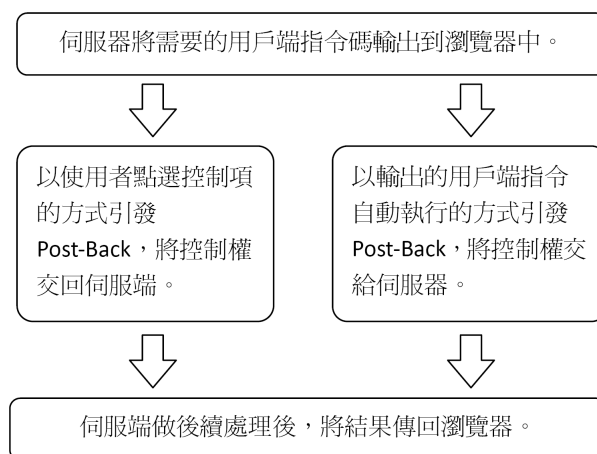
❗ 問題說明

曾經在 MSDN 開發人員論壇上看到這樣的問題, 需要由用戶端指令碼的執行結果來決定接下來伺服器端的執行工作, 當時筆者看到這個問題其實是蠻納悶的, 因為大多數的用戶端資訊, 在 `Request.Browser` 屬性就可以得到, 實在不必要花太多的心思去做這件事情。

筆者能想到唯一的理由, 就只有因為用戶端有安裝 `ActiveX Control`, 伺服器端想要由這個 `ActiveX Control` 得到一些資訊, 這樣的話就確實需要先由用戶端指令碼取回資訊, 才能夠決定伺服器的程式如何執行。這個工作不是做不到, 而且可以很簡單就做到, 只是需要有一些設定, 而且還要有一些必要的控制項才行。

在說明解決方法前, 我們要先知道要怎麼樣才能夠在用戶端和伺服器端間溝通。我們都知道, 伺服器端要能夠和用戶端瀏覽器溝通的方法, 就只有 `HTTP`, 而且需要由伺服器端先下指令給用戶端, 用戶端才能執行, 並且回傳執行結果。而用戶端回傳結果的方法, 不是 `HTTP GET` 就是 `HTTP POST`, 通常我們都會用 `HTTP POST` 來做, 也就是透過表單。因此, 在網頁中就必須要存在一個可以放置用戶端指令碼執行結果的控制項, 這個控制項除了 `HiddenField` 以外, 也許沒有其他的控制項可勝任。

再者, 要由伺服器端發出指令碼給用戶端, 就是用戶端指令碼輸出的議題了, 若讀者不知如何輸出指令碼, 可以參考「Q16：怎樣可以在 ASP.NET 程式中輸出用戶端指令碼？」。其動作示意圖如下：



🔧 解決方案

這個問題有兩種解法，一種是直接由控制項下指令 (輸出啟動用戶端指令碼的呼叫指令)，另一種則是在網頁載入時就下指令，要用哪一種方法，視實際的需求而定。

不論哪一種方法，網頁上都至少要有一個 **HiddenField** (或其他可以讓伺服器收到用戶端執行結果的控制項)，

首先，用控制項下指令的方法，如程式 1 所列。

程式 1：以控制項下指令的方法，執行用戶端指令碼並回傳結果

```

<!-- HTML -->
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title>執行用戶端指令碼</title>
  <script language="javascript" type="text/javascript">
    function executeScript()
    {
  
```

```
        document.getElementById("hiddenField").value = "10000";

        document.form1.submit();
    }
</script>
</head>
<body>
    <form id="form1" runat="server">
        <asp:Button ID="cmdExecuteScript" runat="server"
            Text="執行用戶端指令"
            OnClientClick="executeScript(); event.returnValue=false;" />
        執行回傳值: <asp:Label ID="labelResult" runat="server" />
        <asp:HiddenField ID="hiddenField" runat="server" />
        <asp:Literal ID="ScriptSpace" runat="server"
            EnableViewState="false" />
    </form>
</body>
</html>

// in code
protected void Page_Load(object sender, EventArgs e)
{
    if (Page.IsPostBack) // 請注意，沒有 "!"
        this.labelResult.Text = this.hiddenField.Value;
}
```

作法很簡單, 只要在下指令時, 執行用戶端的指令碼 (`executeScript()`), 在指令碼中將執行結果設定到 `HiddenField` 中, 然後自動送出表單, 如此在伺服器端就可以收到用戶端指令碼的執行結果, 因為用戶端指令碼的送出表單指令 (`document.form1.submit()`) 會產生回呼的狀態, 因此可以在伺服器端判斷是否為回呼, 若為回呼則將 `HiddenField` 的資料取出來即可。

直接在網頁載入時下指令, 作法類似, 但觀念是相同的, 只差在下指令的時機而已, 如程式 2 所列。

程式 2：以在網頁載入時執行指令的方法, 執行用戶端指令碼並回傳結果

```
<!-- HTML -->

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://
www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title>執行用戶端指令碼：直接執行</title>
  <script language="javascript" type="text/javascript">
    function executeScript()
    {
      document.getElementById("hiddenField").value =
        window.navigator.userAgent;
      document.form1.submit();
    }
  </script></head>
<body>
  <form id="form1" runat="server">
    <div>
      瀏覽器的類型：<asp:Label ID="labelBrowserString" runat="server" />
      <asp:HiddenField ID="hiddenField" runat="server" />
      <asp:Literal ID="ScriptSpace" runat="server"
        EnableViewState="false" />
    </div>
  </form>
</body>
</html>
```

```
// in code
protected void Page_Load(object sender, EventArgs e)
{
    if (Page.IsPostBack)
        this.labelBrowserString.Text = this.hiddenField.Value;
    else
    {
        string script = @"
        <script>
            executeScript();
        </script>";
        this.ScriptSpace.Text = script;
    }
}
```

這裡作法不同的地方是, 透過 **Literal** 控制項輸出指令, 呼叫用戶端中的指令碼, 然後透過和前一種方法相同的方式回傳結果。

但在 ASP.NET 2.0 中, 要在 `<%@ Page %>` 中, 加入一個關閉事件驗證的指令 `EnableEventValidation="false"`, 讓 ASP.NET 2.0 不會檢查用戶端傳回的資料與事件對應的關係, 否則會出現驗證失敗的錯誤訊息。

相關問題

- Q16 : 如何在 ASP.NET 程式中輸出用戶端指令碼 ?

Q18

如何由遠端抓取需要登入帳戶才能存取的圖片，並顯示在 Image 控制項？

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

? 問題

我設計了一個小型的網路相簿瀏覽器，用來瀏覽網頁相簿的，但我要顯示的照片都在遠端，而且有些照片存在的網站是有用帳戶與密碼保護的，我要如何抓取這些圖片？

! 問題說明

有些分享的網站，或者是相簿網站，都會給檔案的所有人一個權限，讓他來決定檔案是否公開，或者要經過帳戶驗證才可以瀏覽。在資訊安全逐漸被重視的情況下，有很多的公司也開始在檔案的出入口架設了帳戶管理機制，讓公司內部的檔案管理能夠更安全。

如果想要瀏覽具有帳戶密碼保護的資源，則必須在瀏覽前輸入帳戶密碼，或者在瀏覽前先登入，取得權限後才可以瀏覽，這在瀏覽器上是做得到的，然而若是要在程式中實作，就有些困難，必須要在程式中撰寫登入指令，才能存取到被保護的資源。

首先，必須要先取得具有存取權限的帳戶密碼，如果沒有這組帳戶密碼，怎麼寫都不可能存取到（除非破解），然後，撰寫 HTTP 通訊的程式碼，將帳戶密碼帶入即可。

System.Net 命名空間中有二個專為 HTTP 簡單通訊而設計的類別，分別為 `HttpWebRequest` 與 `HttpWebResponse` 二個類別，各負責 `Request` 和 `Response` 的工作。`HttpWebRequest` 有一個 `Credentials` 屬性，這個屬性會將帳戶和密碼載送到遠端做驗證，例如：

```
HttpWebRequest request =  
    WebRequest.Create(context.Request.QueryString["url"])  
    as HttpWebRequest;
```

```
if (context.Request.QueryString["uid"] != null)
    request.Credentials = new NetworkCredential(
        context.Request.QueryString["uid"],
        context.Request.QueryString["pwd"]);
```

爲了要能夠將讀回的資料流直接設定給 Image 控制項, 可以利用 HTTP Handler (HTTP 處理常式) 來實作, 如此 Image 控制項只要設定 URL 指向 HTTP Handler 並傳入必要的參數, 再由 HTTP Handler 直接輸出圖片資料流, 就能將圖片顯示在 Image 控制項中。例如：

```
Image1.ImageUrl = "MyHandler.ashx?url=xxx&uid=xxx&pwd=xxx";
```

HTTP Handler 是一個程式的處理單元, 它可以直接處理 HTTP 資料流, 並且輸出特定的資料到用戶端, 它最大的特點是, 沒有 Page 類別需要處理的 Rendering, 所以速度上會比較快, 若是想要輸出自己所指定的資料格式或類型時, 使用 HTTP Handler 會比使用網頁輸出要快。

每一個 HTTP Handler 副檔名均爲.ashx, 並且要實作 IHttpHandler 介面中的 ProcessRequest() 方法以及 IsReusable() 方法。ProcessRequest() 方法是最重要的核心處理常式, 所有的 Request 和 Response 都要在這裡處理。一個 HTTP Handler 的原型如下列程式碼所示。

```
<%@ WebHandler Language="C#" Class="Handler" %>

using System;
using System.Web;

public class Handler : IHttpHandler {
```

```
public void ProcessRequest (HttpContext context) {

    // 填入要處理 HTTP Request 並輸出 Response 的程式。
    // 由 context.Request 取得 Request 物件,
    // 由 context.Response 取得 Response 物件。

}

public bool IsReusable {
    get {
        return false;
    }
}
}
```

解決方案

1. 建立 HTTP Handler。
2. 撰寫 HttpRequest 程式, 由遠端取得資源, 並且利用 HttpRequest.Credentials 屬性將必要的登入資訊填入。

```
HttpRequest request =
    WebRequest.Create(context.Request.QueryString["url"])
    as HttpRequest;

if (context.Request.QueryString["uid"] != null)
    request.Credentials = new NetworkCredential(
        context.Request.QueryString["uid"],
        context.Request.QueryString["pwd"]);
```

3. 由 `HttpWebRequest.GetResponse()` 取得 `HttpWebResponse` 物件。
4. 由 `HttpWebResponse.GetResponseStream()` 取得回傳的影像資料流物件。

```
HttpWebResponse response = request.GetResponse() as HttpWebResponse;

// 設定資料容器大小
byte[] data = new byte[(int)response.ContentLength];
int buffer = 8192; // 設定讀取時的緩衝區大小
int readBytes = 0;
int offset = 0;
string contentType = response.ContentType;
Stream s = response.GetResponseStream();
```

5. 讀取影像資料流, 取得影像資料的位元組陣列。

```
do
{
    if (offset == response.ContentLength) // 比對位移是否已到底端
        break;

    if (buffer >= (int)response.ContentLength) // 比對緩衝區大小
        readBytes = s.Read(data, offset, (int)response.ContentLength);
    else
        readBytes = s.Read(data, offset,
            Math.Max(buffer, (((int)response.ContentLength) - offset)));

    offset += readBytes; // 增加位移量。
}
while (readBytes > 0);
```


6. 關閉 `HttpWebResponse`, 釋放連線。

7. 設定回傳的 `ContentType`, 並且將影像資料以 `Response.BinaryWrite()` 輸出。

```
response.Close();

context.Response.Buffer = false;
context.Response.ContentType = contentType;
context.Response.BinaryWrite(data);
```

8. 將 Image 控制項的 `ImageUrl` 設定為 HTTP Handler。

以下為 HTTP Handler 的完整程式碼。

```
<%@ WebHandler Language="C#" Class="GetImageWithCredentials" %>

using System;
using System.IO;
using System.Web;
using System.Drawing;
using System.Net;
using System.Net.Sockets;

public class GetImageWithCredentials : IHttpHandler {

    public void ProcessRequest (HttpContext context)
    {
        HttpWebRequest request =
            WebRequest.Create(context.Request.QueryString["url"])
            as HttpWebRequest;
```

```
if (context.Request.QueryString["uid"] != null)
    request.Credentials = new NetworkCredential(
        context.Request.QueryString["uid"],
        context.Request.QueryString["pwd"]);

HttpWebResponse response =
    request.GetResponse() as HttpWebResponse;

byte[] data = new byte[(int)response.ContentLength];
int buffer = 8192;
int readBytes = 0;
int offset = 0;
string contentType = response.ContentType;
Stream s = response.GetResponseStream();
do
{
    if (offset == response.ContentLength)
        break;

    if (buffer >= (int)response.ContentLength)
        readBytes = s.Read(data, offset, (int)response.ContentLength);
    else
        readBytes = s.Read(data, offset,
            Math.Max(buffer, (((int)response.ContentLength) - offset)));

    offset += readBytes;
}
while (readBytes > 0);

response.Close();
```

```
context.Response.Buffer = false;

context.Response.ContentType = contentType;

context.Response.BinaryWrite(data);
}

public bool IsReusable {
    get {
        return false;
    }
}
}
```

小常識 HttpWebRequest 和 WebClient 用哪一個比較好?

WebClient 可以說是更簡化版本的 HttpWebRequest, 它將平常用到的動作都做完了, 例如 DownloadFile() 可以下載遠端資料到檔案, 但它把資料接收的工作都做掉了, DownloadData() 則是下載位元組陣列的資料, 同時它也提供了非同步的資料傳輸機制。但也因為它暗中做掉了一些事情, 因而無法做到更多客制化的能力。

因此, 若想要省工而且沒有太多想要自己做的事情的話, 可以考慮使用 WebClient 來實作通訊, 反之, 則應利用 HttpWebRequest, 方法和 HttpWebRequest 的作法差不多, 只差在 WebClient 不必處理 Stream, 直接用 DownloadData() 或 DownloadFile() (若已知檔案名稱時) 即可下載資料。

相關問題

- Q25 : 如何動態產生圖片?

Q19

我要如何要求使用者依據指定的格式輸入電話之類的資料？

適用範圍： ☒ ASP.NET 1.0 ☒ ASP.NET 1.1 ☒ ASP.NET 2.0 ☒ ASP.NET 3.5 ☒ ASP.NET AJAX

問題

最近公司的 DBA (資料庫管理員) 在反應, 公司部份資料庫的資料內容格式不一 (出貨單號碼格式, 客戶電話與地址格式, Email 格式等), 希望能夠將輸入時的格式就確定好, 請問我要如何做？

問題說明

資料格式的問題, 相信有很多應用程式都有這個困擾, 尤其是「使用者為大」的前提下, 開發人員想要設定強制定格式, 都可能會被使用者罵, 如果還可以在後端修正格式的話就還可以處理, 若一定得要由使用者輸入, 而沒有做一些管制, 那麼資料的格式一定會很亂。

例如光是電話格式, 可能就有這麼多種：

- 02-20392839
- (02)20392839
- 0220392839
- 02-2039-2839
- (02)2039-2839
- ...

地址格式也是不遑多讓, 例如：

- 高雄市 xx 路 xxx 號
- 高雄市 xx 區 xx 路 xxx 號
- 高雄市 xx 區 xx 里 xx 路 xxx 號
- 高雄市 xx 區 xx 里 xx 鄰 xx 路 xxx 號

很多類似這種資料的格式, 幾乎都是以使用者「一份格式, 各自表述」的思維去輸入, 很少人會去想資料格式統一這個問題, 因為「這樣輸入也通, 那樣輸入也通」, 但這可苦了想用這些資料來做分析的人。

一般爲了要實作特定資料格式的輸入, 開發人員都會把表單按資料的格式去排列, 例如地址就會分縣市, 鄉鎮市區, 路街, 巷弄, 號碼以及樓室等, 而電話就會切分區號和號碼, 不過 Email 可能就沒有這麼好做, 因為 Email 的格式都是 `username@mailserverurl` 的格式, 如果用分開的控制項來輸入, 對於使用者來說真的不太方便。

爲了要驗證 Email 這樣的資料格式, 通常熟練的開發人員都會利用 **Regular Expression** (通用運算式) 來設計檢驗的資料格式樣板 (pattern), 然後套用到 **RegularExpressionValidator** 來做驗證, 例如一個驗證 Email 的 **RegularExpression** 可能會長的像這樣：

```
^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}$
```

不過由於 **Regular Expression** 的複雜度真的不低, 光是那幾個符號的用法就不太容易懂了, 因此進入的門檻算是很高, 一般開發人員不會刻意去學它 (除非真有必要), 而網路上也有不少的範例網站可以看, 例如 <http://www.regular-expressions.info> 這個網站就有提供一些範例的指令, 也可以在搜尋引擎中輸入 **Regular Expression** 來搜尋, 亦可找到不少資料。

ASP.NET 2.0 在 **ASP.NET AJAX** 的加持下, 可以利用 **ASP.NET AJAX Control Toolkit** 中的 **MaskedEdit** 控制項, 來設定使用者的輸入格式, 而這也是本問題的主要解法, 但若是 **ASP.NET 1.x**, 就沒有這樣的福利了。

MaskedEdit 可以在使用者輸入時, 顯示輸入資料的格式, 並且可以強制使用者輸入字元或數字等, 同時亦可以即時的驗證資料的格式正確與否。它是屬於 **ASP.NET AJAX Control Toolkit** 所附的控制項之一, 也需要搭配 **ASP.NET** 的控制項來實作 (通常是搭配文字方塊), 其參數說明如程式 1。

程式 1：MaskedEdit 控制項參數說明

```

<ajaxToolkit:MaskedEditExtender ID="maskedEdit" runat="server"
    TargetControlID="TextBox2" // 要加入 MaskedEdit 設定的目標控制項 ID
    Mask="9, 999, 999.99" // 輸入格式
    // 是否顯示驗證訊息的 Tooltip (true/false)
    MessageValidatorTip="true"
    // 設定當控制項取得焦點時要套用的 CSS 類別
    OnFocusCssClass="MaskedEditFocus"
    // 設定當控制項格式錯誤時要套用的 CSS 類別
    OnInvalidCssClass="MaskedEditError"
    MaskType="Number" // 遮罩類型
    InputDirection="RightToLeft" // 輸入方向
    AcceptNegative="Left" // 接受顯示負數符號的方向
    DisplayMoney="Left" // 顯示貨幣符號的方向
    // 是否顯示驗證錯誤的 Tooltip (true/false)
    ErrorTooltipEnabled="True"/>

```

其中

- 至少要填入 TargetControlID 以及 Mask 兩個屬性。
- MaskType 有下列的值可以使用：{ None, Date, Time, Number, DateTime }。
- Mask 可用的設定有：

9	表示一個數字。
L	表示一個字元。
\$	表示一個字元或一個空白。
C	表示一個自訂符號（分大小寫）。
A	表示一個字元或一個自訂符號。
N	表示一個數字或一個自訂符號。
?	表示任何的字元或數字。

/	日期的分隔字元, 例如 9999/99/99
:	時間的分隔字元, 例如 99:99
.	十進位分隔字元, 例如 999, 999.999
,	千位數分隔字元, 例如 999, 999, 999
\	逸出字元, 用在上述符號字元但要出現在 Mask 中當成一般文字時, 例如 \NT\\$999, 999, 999。

設定範例：

- 輸入日期時間：9999/99/99 99:99
- 輸入金額：\\$999, 999, 999
- 輸入電話：(99)9999-9999
- 輸入 FM0091-9992-AG01 格式：LL9999-9999-LL99

每一個 MaskedEdit 控制項, 可搭配一個 MaskedEditValidator 來執行驗證工作, 而驗證的結果可以透過顯示字串或者是 Tooltip 的方式來通知使用者, 其參數說明如程式 2。

程式 2：MaskedEditValidator 控制項的參數說明

```
<ajaxToolkit:MaskedEditValidator
    ControlExtender="MaskedEditExtender2" // 要搭配的 MaskedEdit 控制項 ID
    ControlToValidate="TextBox2" // 要驗證的控制項 ID
    IsValidEmpty="False" // 是否允許空白值 (true/false)
    MaximumValue="12000" // 允許的最大值
    // 空值時顯示在 Tooltip 的錯誤訊息
    EmptyValueMessage="Number is required"
    // 值無效時顯示在 Tooltip 的錯誤訊息
    InvalidValueMessage="Number is invalid"
    // 超過 MaximumValue 時顯示在 Tooltip 的錯誤訊息
    MaximumValueMessage="Number > 12000"
```

```

// 少於 MinimumValue 時顯示在 Tooltip 的錯誤訊息
MinimumValueMessage="Number < 100"
MinimumValue="100" // 允許的最小值
// 空值，且要驗證的控制項失去焦點時顯示的錯誤訊息
EmptyValueBlurredText="*"
// 無效值，且要驗證的控制項失去焦點時顯示的錯誤訊息
InvalidValueBlurredMessage="*"
// 超過 MaximumValue，且要驗證的控制項失去焦點時顯示的錯誤訊息
MaximumValueBlurredMessage="*"
// 少於 MinimumValue，且要驗證的控制項失去焦點時顯示的錯誤訊息
MinimumValueBlurredText="*"
Display="Dynamic" // 顯示的模式
// 當要驗證的控制項是空值且取得焦點時顯示的訊息。
TooltipMessage="Input a number: 100 - 12.000"
/>

```

例如，若要實作一個可輸入金額 0-25000 的 MaskedEdit 控制項，並且不允許空白，且要在輸入錯誤時提示，其宣告的方法如程式 3 所示。

程式 3：範例應用：必須輸入 0-25000 之間數字的 MaskedEdit 與 MaskedEditValidator 控制項。

```

輸入金額：<asp:TextBox ID="T_Amount" runat="server" Columns="15" />
<ajaxToolkit:MaskedEditExtender ID="ME_T_Amount"
    TargetControlID="T_Amount" InputDirection="rightToLeft"
    runat="server" MaskType="Number" AcceptNegative="None"
    Mask="\NT\$99, 999" MessageValidatorTip="true"
    ErrorTooltipEnabled="true" />
<ajaxToolkit:MaskedEditValidator ID="MEV_T_Amount" Display="none"
    MinimumValue="0" MaximumValue="25000" runat="server"
    ControlExtender="ME_T_Amount" MaximumValueMessage="超過 25000"
    MaximumValueBlurredMessage="超過 25000"
    ControlToValidate="T_Amount"
    TooltipMessage="必須在 0-25000 之間" />

```


解決方案

若是 ASP.NET 1.x, 則只能利用控制項排列或是 Regular Expression 來檢驗格式, 若是 ASP.NET 2.0, 則可以善加利用 ASP.NET AJAX Control Toolkit 中提供的 MaskedEdit 以及 MaskedEditValidator 控制項來控制輸入的格式。

小常識 資料格式的規劃與控制

一般在實務設計上, 資料格式通常在設計時期就要訂定好, 然後寫到商業邏輯層去, 前端的使用者介面在輸入資料時, 就要限定好格式, 若無法限定, 也要做控制項排列, 讓使用者輸錯格式的機率降低, 倘若資料是來自於外部 (例如資料轉移), 就要盡可能維持正確格式 (但通常很困難)。

若格式太過特殊, 無法轉換成正確格式時, 可能就要考慮由人工的方式來修正, 最常見的格式錯誤應該算是地址和電話吧, 但如果格式有錯誤時, 會對後續的分析造成影響, 甚至影響分析結果, 因此資料格式的淨化 (Clean) 在資料品質 (Data Quality) 中, 是非常重要的一環, 也因為資料要夠乾淨, 分析的結果才不會偏誤太大。

相關問題

若有 ASP.NET AJAX 組態設定問題, 可參考：

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

Q20

我之前有寫好的用戶端驗證程式碼，要如何整合到 ASP.NET 的驗證功能中？

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

❓ 問題

ASP.NET 內建的各式驗證控制項真的很好用，在發展 Web 應用程式時的幫助很大，但是現在有個問題，如果我的規則是二個欄位以上（例如填寫的欄位二個可以只填其中一個），或是一些特殊的驗證規則就無法支援，之前我是有寫了一些用戶端驗證的指令碼，那現在我要怎麼樣將這些指令碼整合在 ASP.NET？

💡 問題說明

ASP.NET 提供了四種不同的驗證控制項，讓開發人員依不同的需要選用，在大多數的日常工作中，這四種控制項都能夠勝任愉快，其中的 `RangeValidator`、`RequiredFieldValidator` 以及 `CompareValidator` 三個控制項都是以特定任務為主的驗證控制項，而具有彈性的則是 `RegularExpressionValidator` 控制項，它可以視開發人員所訂定的 `Regular Expression` 來做驗證工作。

但這些驗證控制項都有一個共通的缺點，就是無法跨欄位做驗證，也就是說，它們都只能針對一個欄位做驗證，沒辦法跨二個以上的欄位。這就會有一些驗證上的問題，例如電話是必填，但不限於手機或住家電話時，是要指示使用者，或者是保留手機和住家電話的欄位？這個就值得玩味了。

微軟也考量到不同的應用程式會有不同的表單驗證需求，因此在 ASP.NET 中提供了 `CustomValidator` 控制項，讓開發人員可以自己開發驗證的邏輯，用戶端和伺服器驗證都可以。

若要運用 `CustomValidator` 來做驗證的話，驗證函式必須要符合下列的函式格式，有了用戶端的驗證函式，就可以將它指定給 `CustomValidator`。

```
function [funcation_name](oSrc, args)
// oSrc: 引發驗證的控制項, args: 參數
{
    var validated = false;
    // 自己的驗證碼。

    if (validated = true)
        args.IsValid = true; // 驗證成功
    else
        args.IsValid = false; // 驗證失敗
}
```

一定要設定 `args.IsValid` 的值, 否則 `CustomValidator` 無法得知驗證結果。

若想在伺服器端驗證資料, 則可以處理 `CustomValidator` 的 `ServerValidate` 事件, 來驗證控制項的資料是否正確。

```
<asp:CustomValidator
    ClientValidationFunction="client_validation_function"
    OnServerValidate="event_handler" />
```

ClientValidationFunction	指定用戶端要用來驗證的函數名稱。
OnServerValidate	處理 <code>ServerValidate</code> 事件的常式名稱。

🔧 解決方案

將現有的用戶端指令碼, 引入網頁中, 再於網頁中加入 `CustomValidator` 需要的用戶端指令碼函式, 把驗證指令加入到函式中, 即可讓 `CustomValidator` 處理用戶端的驗證工作。

相關問題

- Q16: 如何在 ASP.NET 程式中輸出用戶端指令碼?

Q21

如何使用 ASP.NET Ajax 函式庫執行表單驗證？

適用範圍： ☒ ASP.NET 2.0 ☒ ASP.NET 3.5 ☒ ASP.NET AJAX

❓ 問題

我的網站上有使用表單驗證模式來做的安全功能,最近老闆要我將 ASP.NET AJAX 功能納入網站的設計中,基本上問題不大,但目前的麻煩是要怎麼讓 ASP.NET AJAX 可以使用表單驗證來驗證使用者的登入資訊,請問有何方法？

❗ 問題說明

ASP.NET AJAX 除了部份網頁更新能力以及網頁載入能力的 UpdatePanel 與 UpdateProgress 兩個主要控制項外,還加入了一些核心的程式支援功能,讓 ASP.NET AJAX 可以支援一些 ASP.NET 2.0 上所發展的新服務,例如 Membership 以及 Profile,因此在 ASP.NET AJAX 中有一個稱為 Application Service 的功能,就是針對這二個服務所設計,同時也提供了可客制化的延展性。

表單驗證 (Forms Authentication) 就是 Application Service 的其中一項,它預設的功能是直接支援 Membership Service,但它也考量到了現有網站所需要的驗證方式,所以也支援了自訂的驗證方法,讓沒有使用 Membership Service 的 ASP.NET 2.0 網站也可以利用。ASP.NET AJAX 的 Application Service 不像 UpdatePanel 和 UpdateProgress 一樣,直接拖拉元件就可以用,而是要自己寫一點程式碼的,若要自訂驗證,則要寫的程式碼就更多了。

ASP.NET AJAX Application Service 是經由 Sys.Service 命名空間中的 AuthenticationService 以及 ProfileService 來實作,各掌管 Forms Authentication 以及 Profile Service 二個服務,AuthenticationService 將 XMLHTTP 的通訊方式都隱藏起來,開發人員只需要呼叫下列的函式,以及處理相關的事件。

函式／事件	用途
login()	執行登入作業（原型如程式 1）。
logout()	執行登出作業（原型如程式 2）。
defaultLoginCompletedCallback	預設情況下，登入成功時的 Callback 處理常式。
defaultLogoutCompletedCallback	預設情況下，登出成功時的 Callback 處理常式。
defaultFailedCallback	預設情況下，登入或登出失敗時的 Callback 處理常式。
isLoggedIn	確認是否已登入。
timeout	登入或登出的逾時設定。

程式 1：Sys.Services.AuthenticationService.login 的參數說明

```

Sys.Services.AuthenticationService.login(
    userName, // 使用者名稱
    password, // 密碼
    isPersistent, // 是否允許跨瀏覽器 (true/false)
    redirectUrl, // 登入成功時要轉向的 URL，若設 null 則不會轉向。
    customInfo, // 保留給未來使用，一定要填 null
    loginCompletedCallback, // 登入成功時的 Callback 處理常式
    failedCallback, // 登入失敗時的 Callback 處理常式
    userContext // 其他的使用者資訊
);

```

程式 2：Sys.Services.AuthenticationService.logout 的參數說明

```

Sys.Services.AuthenticationService.logout(
    redirectUrl, // 登出成功時要轉向的 URL，若設 null 則不會轉向。
    logoutCompletedCallback, // 登出成功時的 Callback 處理常式
    failedCallback, // 登出失敗時的 Callback 處理常式
    userContext // 其他的使用者資訊
);

```

若想要自訂驗證的模式, 開發人員要另外新增一個 Web Service (若已有的話則要修改一下參數), 用來接收來自於 `Sys.Services.AuthenticationService.login()` 傳送出來的登入資訊, 並執行驗證後回傳結果。這個函式的原型必須如程式碼 3 所示。

程式 3：支援自訂驗證方法的 Web Service 原型。

```
<%@ WebService Language="C#" Class="MyAuthenticationService" %>

using System.Web.Services;
using System.Web.Script.Services;

[ScriptService] // 標註 Web Service 可支援 ASP.NET AJAX 的呼叫
public class MyAuthenticationService : System.Web.Services.WebService
{
    [WebMethod]
    public bool Login(string userName, string password,
        bool createPersistentCookie)
    {
        // 撰寫登入驗證程式碼, 並且若驗證成功, 回傳 true, 否則回傳 false
        return true;
    }

    [WebMethod]
    public void Logout()
    {
        // 撰寫登出程式碼。
    }
}
```

有了這些知識, 就可以來解這個問題了。

解決方案

1. 在網頁上放置 Script Manager, 這個控制項是總管 ASP.NET AJAX 的指令碼工作, 所以不能沒有。
2. 在網頁上建立登入的畫面。



3. 撰寫執行登入的 JavaScript 指令碼, 然後設定登入與登出按鈕呼叫用戶端指令碼, 如程式 4 與程式 5。

程式 4：用戶端登入指令碼

```
// 初始化 ASP.NET Ajax library 物件
if (typeof(Sys) !== "undefined")
    Sys.Application.notifyScriptLoaded();

var panelLogin = null;
var panelLogout = null;

function pageLoad()
{
    panelLogin = $get("panelLogin");
    panelLogout = $get("panelLogout");
}

function login()
{

```

```
var username = $get("<% = this.T_UserName.ClientID %>").value;
var password = $get("<% = this.T_Password.ClientID %>").value;
// 登入處理
Sys.Services.AuthenticationService.login(
    username, password, false, null, null,
    OnLoginCompleted, OnFailed, "User Context");
}

function logout()
{
    // 登出處理
    Sys.Services.AuthenticationService.logout(
        null, OnLogoutCompleted, OnFailed, null);
}

function OnFailed(error, userContext, methodName)
{
    // 顯示通知訊息
    alert("登入時發生錯誤，訊息：" + error.get_message());
}

function OnLoginCompleted(validCredentials, userContext, methodName)
{
    // 根據結果變更畫面
    if (validCredentials == true)
    {
        panelLogin.className = "displayHide";
        panelLogout.className = "displayVisible";
    }
    else
    {
        alert("使用者帳戶或密碼錯誤。");
    }
}
```



```
function OnLogoutCompleted(result)
{
    panelLogin.className = "displayVisible";
    panelLogout.className = "displayHide";
}
```

程式 5：設定登入與登出的指令按鈕呼叫用戶端指令碼

```
...
<asp:Button ID="cmdLogin" OnClientClick="login(); return false;"
    runat="server" Text=" 登入 " />
...
<asp:Button ID="cmdLogout" runat="server" Text=" 登出 "
    OnClientClick="return confirm('確定要登出？若登出，目前在進行且未儲存的工作記錄就會消失。');" />
...
```

4. 撰寫負責處理登入驗證的 Web Service (如程式 6)，其登入驗證的函式原型必須要符合 ASP.NET AJAX 的呼叫格式，否則在測試時一定會發生錯誤，不是格式不正確，就是方法錯誤。

程式 6：處理 ASP.NET AJAX 登入功能的 Web Service 程式

```
using System;
using System.Web;
using System.Collections;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.Web.Script.Services;
[ScriptService]
public class LoginService : System.Web.Services.WebService
{
```

```

[WebMethod]
public bool Login(string userName, string password,
    bool createPersistentCookie)
{
    // 此僅為範例，要用真正處理登入的程式代替。
    if (userName == "test" && password == "password")
        return true;
    else
        return false;
}
}

```

5. 設定 **ScriptManager**, 加入 **AuthenticationService** 以及指向負責驗證的 **Web Service**, 若沒有, 則預設會是呼叫內建的 **Membership Service** 驗證服務, 如程式 7。

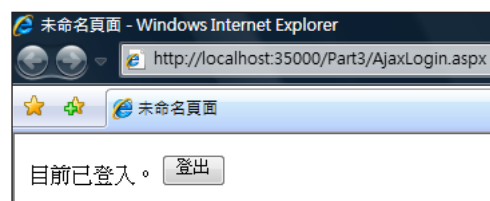
程式 7：設定 **AuthenticationService**, 指向負責驗證的 **Web Service**

```

<asp:ScriptManager ID="ScriptManager1" runat="server">
    <AuthenticationService Path="LoginService.asmx" />
</asp:ScriptManager>

```

6. 啟動瀏覽器, 測試這個網頁, 使用者名稱輸入 "test", 密碼輸入 "password", 並按下登入, 若看到下列畫面, 則表示功能已正常運作。



相關問題

若有 ASP.NET AJAX 組態設定問題, 可參考：

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

Q22

我有寫好的用戶端指令碼, 要如何跟 ASP.NET AJAX 函式庫整合在一起使用?

適用範圍: ☒ ASP.NET 2.0 ☒ ASP.NET 3.5 ☒ ASP.NET AJAX

? 問題

我之前使用 PHP 和 ASP 開發網站應用程式, 因為系統的需要, 寫了很多的用戶端指令碼, 現在公司要導入 ASP.NET 2.0 以及 ASP.NET AJAX, 那麼我先前寫的這些指令碼, 要怎麼樣才能和 ASP.NET AJAX 合併在一起使用?

! 問題說明

很多企業在早期 ASP 的時代, 都發展了不少 Web 應用程式, 當時只有五大物件 (Request, Response, Application, Server 和 Session) 的時代, 連 AJAX 都沒個邊的時候, 一些設計品質較佳的應用程式, 撰寫了大量的指令碼是很稀鬆平常的事, 早期的資料驗證與檢查, 以及互動式的介面設計, 因為沒有一些工具或元件可以支援, 就需要寫一堆的指令碼, 來達成想要的互動式效果。

不過這些能力在 ASP.NET 的歷次改版中逐漸的被 ASP.NET 的內建元件所取代, 讓開發的工作可以稍稍輕鬆些, 但有些工作或效果, 是 ASP.NET 的內建元件無法取代的, 例如 JavaScript 的下拉式日曆, 以及 HTML 的編輯器等, 都無法被 ASP.NET 的內建元件取代, 而且還被廣泛的用在現有的 ASP.NET 程式之中。

ASP.NET AJAX 除了自身的 JavaScript 指令碼外, 也支援由外部參考進來的 JavaScript 指令碼, 甚至於是內嵌在組件資源 (Assembly Resource) 的指令碼, 也可以由 AJAX 取出使用。而這些功能可以利用 ScriptManager 的參數來處理, 若是直接放在 HTML 中的指令碼, AJAX 一樣可以處理使用。

若想要引用一個外部的指令碼檔案 (.js), 則可以設定 ScriptManager 的 ScriptReference 參數:

```
<asp:ScriptManager ID="ScriptManager1" runat="server">
    <ScriptReference Path="~/scripts/myscripts.js" />
</asp:ScriptManager>
```

若想要引用內含在組件中的指令碼資源,則 `ScriptManager` 的 `ScriptReference` 要如此設定：

```
<asp:ScriptManager ID="ScriptManager1" runat="server">
  <ScriptReference Assembly="MyScriptRes"
    Name="MyScriptRes.MyScript.js" />
</asp:ScriptManager>
```

🔧 解決方案

若要將現成的用戶端指令碼加入 ASP.NET AJAX 網頁中,有三種方法可用：

1. 透過 `ScriptManager` 來引用,若是指令碼在組件中時,要指定 `Assembly` 以及 `Name` 屬性,若是一個指令碼檔案,則要設定 `Path`。
2. 將指令碼加到 HTML 網頁中。
3. 將指令碼以 `<script type="text/javascript" src="..." />` 的方式,加入參考到 HTML 頁面中。

相關問題

若有 ASP.NET AJAX 組態設定問題,可參考：

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

Q23

如何使用用戶端指令碼動態產生用戶端的控制項？

適用範圍： ☒ ASP.NET 1.0 ☒ ASP.NET 1.1 ☒ ASP.NET 2.0 ☒ ASP.NET 3.5 ☒ ASP.NET AJAX

? 問題

我有一張問卷表單, 裡面有很多的問題項目, 並且有單選題和複選題, 因為題目太多了, 我想要用動態產生控制項的方式來做, 但是如果每次都用動態產生的話, 又可能有 PostBack 的問題, 請問我有什麼方法, 可以避開這個 PostBack 的狀況？

! 問題說明

這個問題其實很常見, 在很多公司裡面的表單, 有時候會有很多重覆形態的格式 (例如出貨單與銷售清單這一類型的表單), 如果要一個一個拉, 可能拉到專案結束都拉不完 😊, 這個時候用動態控制項生成 (Dynamic Control Creation) 的方法, 可以省下非常多的工夫。

但如此一來就會有個問題, 如果每次格式都要由伺服器來產生的話, 勢必會有 Post-Back 的 Round-trip 問題, 對於使用者的使用經驗來說也不太好, 如果能直接在用戶端做掉的話, 不但可以省下 Round-trip 的時間, 而且也可以保持比較良好的使用者互動經驗, 至少不必再一直看著網頁 Post-Back。

如果要品質好的程式, 不 coding 是騙人的, 如果要做到沒有 Round-trip 的話, 勢必就要自理控制項的生成工作, 而且還要把控制項填入的資料回傳給伺服器處理, 前端的指令都要使用 JavaScript 來做, 然後利用 HTTP POST 方法傳回後端。

若要在前端撰寫操作控制項的 JavaScript 指令碼, 則必須要對幾件事情很熟悉：

- JavaScript 語言本身, 包括語法及函式。
- HTML 各類物件。
- 瀏覽器的 DOM 物件模型。
- HTTP 的通訊方法。
- XML (如果要實作 AJAX 的話)。

如果要用用戶端指令碼建立一個控制項, 可以使用 `document.createElement()` 來實作。

```
var newobject = document.createElement(tagName);
```

參數	說明
tagName	要建立的 HTML 控制項名稱。

範例如下：

```
// 建立一個 Button
var obj = document.createElement("INPUT TYPE=button");

var obj = document.createElement("SELECT"); // 建立一個清單控制項
var obj = document.createElement("INPUT TYPE=text"); // 建立一個文字方塊
var obj = document.createElement("TABLE"); // 建立表格
```

或者, 也可以利用容器物件 (像 DIV ; 表格或是 SPAN 等), 直接在物件的 `innerHTML` 屬性中設定控制項的資訊, 如此就可以像下很多行的 `document.createElement()` 一樣產生控制項。

```
<div id="ctlContainer"></div>
<script language="JavaScript" type="text/javascript">

// 取得 ID 為 ctlContainer 的 DIV 物件
var obj = document.getElementById("ctlContainer");
```

```
// 設定要動態產生的控制項。
obj.innerHTML =
    "<input type='text' id='T_Value' style='width: 300px'>" +
    "<input type='button' id='cmdSubmit' value='送出'>";

</script>
```

動態產生的控制項是沒有事件處理常式的, 如果要設定事件常式的話, 可以用這樣的格式：

```
object.eventName = eventHandler;
```

參數	說明
object	要設定事件常式的物件變數。
eventName	要設定事件常式的事件。
eventHandler	要設定給事件的事件名稱。

範例如下：

```
cmdSubmit.onclick = cmdSubmit_Click;
```

如果不要特別為單一個控制項建立事件常式, 那也可以用這樣的方法來寫：

```
object.eventName = new function() {
    // 填入事件處理常式的內容。
};
```

接著, 就以問題所提之問卷調查為例, 實際撰寫一個簡單的應用。

🔧 解決方案

首先, 先將問卷內容以及題型答案等, 建立一個 XML 格式的資料, 並將使用者的答案也建立一個 XML 清單的資料, 讓資料可以儲存並傳給伺服器, 範例資料如程式 1, 範例回應資料如程式 2 所示。

程式 1: 問卷的 XML 資料格式

```
<Questions>

  <Question ID="1" Caption="請問您對於這次的活動整體滿意度為何 (9 表很滿意, 1 表很不滿意, 5 為普通)" ResponseType="SingleChoice">

    <AnswerList>

      <Answer Caption="9" Value="9" />
      <Answer Caption="8" Value="8" />
      <Answer Caption="7" Value="7" />
      <Answer Caption="6" Value="6" />
      <Answer Caption="5" Value="5" />
      <Answer Caption="4" Value="4" />
      <Answer Caption="3" Value="3" />
      <Answer Caption="2" Value="2" />
      <Answer Caption="1" Value="1" />

    </AnswerList>

  </Question>

  <Question ID="2" Caption="請問您喜歡哪些活動? 可複選" ResponseType="MultipleChoice">

    <AnswerList>

      <Answer Caption="肝膽相照" Value="A" />
      <Answer Caption="舞蹈排練" Value="B" />
      <Answer Caption="營火晚會" Value="C" />
      <Answer Caption="大地遊戲" Value="D" />
      <Answer Caption="烤肉活動" Value="E" />

    </AnswerList>

  </Question>

</Questions>
```



```
</Question>

<Question ID="3" Caption="請問您對於這次的活動有何意見要反映？"
  ResponseType="MultipleText" />

</Questions>
```

程式 2：回應資料 XML 格式。

```
<Responses>

  <Response ID="1"></Response>

  <Response ID="2"></Response>

  <Response ID="3"></Response>

</Responses>
```

準備好 XML 之後，就可以開始寫動態產生控制項的指令碼，依照題型來區分，有單選，多選及文字方塊的題型，所以每一種使用的控制項都不一樣，單選的題目要用 **Radio Button**，而多選要用 **CheckBox** 來做，除了使用的控制項不同外，其實大同小異，如程式 3 所示。

程式 3：動態新增 Radio Button 與 CheckBox 的程式碼 (片段，位在 QuestionLoad() 函式)

```
case "SingleChoice":

  answerList = nodes[i].selectNodes("AnswerList/Answer");

  for (var j=0; j<answerList.length; j++)
    html += "<INPUT TYPE='radio' name='o" +
      nodes[i].attributes.getNamedItem("ID").value +
      "' onclick='setResponse(" +
      nodes[i].attributes.getNamedItem("ID").value +
      ", \"radio\", \"" +
      answerList[j].attributes.getNamedItem("Value").value +
      "\" )' />" +
      answerList[j].attributes.getNamedItem("Caption").value;
```

```

html += "<br><br>";

break;

case "MultipleChoice":

    answerList = nodes[i].selectNodes("AnswerList/Answer");

    for (var j=0; j<answerList.length; j++)
        html += "<INPUT TYPE='checkbox' name='c' +
            nodes[i].attributes.getNamedItem("ID").value + "' id='c" +
            nodes[i].attributes.getNamedItem("ID").value + "_" + j +
            "' onclick='setResponse(" +
            nodes[i].attributes.getNamedItem("ID").value +
            ", \"checkbox\", null)' value='" +
            answerList[j].attributes.getNamedItem("Value").value + "' />" +
            answerList[j].attributes.getNamedItem("Caption").value;

html += "<br><br>";

break;

```

文字方塊的新增就簡單多了, 只要新增 TEXTAREA 就行了, 參數也比較簡單, 如程式 4 所示。

程式 4：文字方塊的動態產生程式碼 (片段, 位在 QuestionLoad() 函式)

```

case "MultipleText":

    html += "<TEXTAREA style='width: 400px; height: 100px'
onchange='setMultipleTextResponse("

```

```
+ nodes[i].attributes.getNamedItem("ID").value + ") '></TEXTAREA>";

break;
```

有了控制項之後，就要有處理資料的程式，在這個範例中，由兩個函式來處理，一個是 `setResponse()`，另一個則是專門給文字方塊用的 `setMultipleTextResponse()`，它們負責的是將使用者的填答寫入回應 XML 中，如程式 5 所示。

程式 5：setResponse()與 setMultipleTextResponse()程式

```
function setResponse(id, type, value)
{
    var xmlDoc = document.getElementById("QAnswer").XMLDocument;

    if (type == "radio")
        xmlDoc.documentElement.selectSingleNode(
            "//Responses/Response[@ID='" + id + "']").text = value;
    else
    {
        var objs = document.getElementsByName("c" + id);
        var list = "";
        // 將有填答的選項結合成 "A, B, C" 格式（即以逗號分隔的清單）。
        for (var i=0; i<objs.length; i++)
        {
            if (objs[i].checked)
            {
                alert(objs[i].value);
                list += (list == "") ? objs[i].value : ", " + objs[i].value;
            }
        }
    }
}
```

```

        xmlDoc.documentElement.selectSingleNode(
            "//Responses/Response[@ID='" + id + "']").text = list;
    }
}

function setMultipleTextResponse(id)
{
    var xmlDoc = document.getElementById("QAnswer").XMLDocument;
    xmlDoc.documentElement.selectSingleNode(
        "//Responses/Response[@ID='" + id + "']").text =
        event.srcElement.value;
}

```

在使用者填完資料後, 要將填好的資料傳給伺服器, 需要有一個容器, 這個容器用 HiddenField 來當最適合了, 再寫一段程式, 在發送前先將回應的 XML 填到 HiddenField 中, 再執行送出的工作, 如程式 6 所示。

程式 6：傳送前處理工作的 finish 函式及套用指令

```

function finish()
{
    document.getElementById("response").value =
        document.getElementById("QAnswer").XMLDocument.xml;
    return true;
}

...

<asp:HiddenField ID="response" runat="server" />
<asp:Button ID="cmdSubmit" runat="server"
    OnClientClick="return finish()" Text="填完了"
    OnClick="cmdSubmit_Click" />

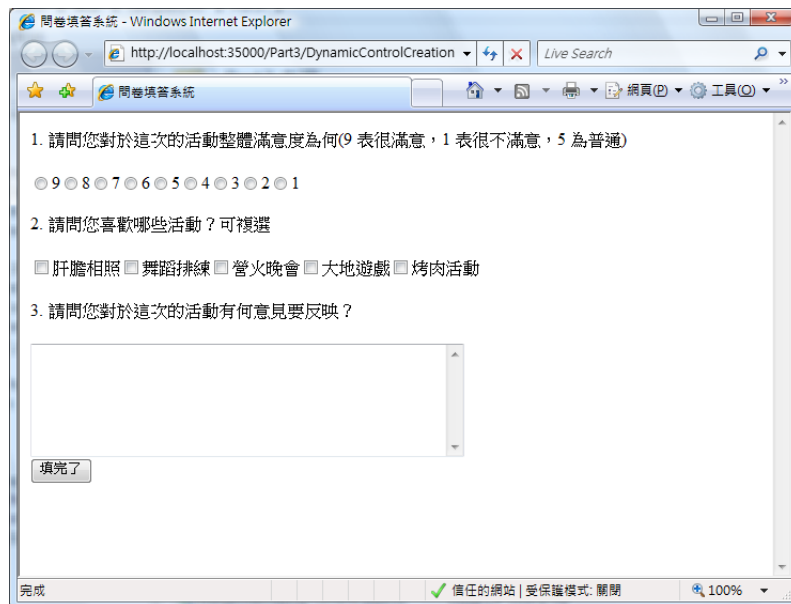
```

最後，下達觸發動態生成的指令，就大功告成了，如程式 7 所示。

程式 7：啟動動態產生問卷畫面的指令

```
<script type="text/javascript" language="javascript">
    QuestionLoad();
</script>
```

本程式執行畫面如下：



在問卷填寫完畢後，按下按鈕，伺服器端會收到一份已經記錄好使用者填答資料的 XML 檔案，伺服器端再利用 System.Xml 的 XmlDocument 來處理即可。

這個範例只是其中的一種應用，完整的程式碼可由筆者的範例網站下載。若想要深入了解的話，可能要看許多的書，才会有能力寫出複雜的用戶端控制項動態生成的應用程式，重點是要熟悉 HTML 和 DOM 物件模型，因為光這個主題就可以寫至少一本書了，所以筆者就敘述到此，有興趣的讀者可到書店找有關的書籍來學習。

Q24

如何在伺服器端動態產生控制項？

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

❓ 問題

我有一張問卷表單, 裡面有很多的問題項目, 並且有單選題和複選題, 因為題目太多了, 我想要在伺服器端用動態產生控制項的方式來做, 請問我要如何做？

💡 問題說明

像這種類型的問題, 其實網路上有非常多的人在問, 幾乎已經是個 FAQ 級的問題了, 解法也有很多人提出來, 以 OOP 的角度來看, 動態產生控制項和產生物件其實沒什麼兩樣, 也是透過 `new` 來產生控制項物件, 然後再把物件插入到容器中而已。

不過在 ASP.NET, 或者是其他的 Web-Based 應用程式發展平台來說, 這可能就不是這麼簡單了, 因為 Web-Based 的應用程式都是無狀態 (State-less) 的, 也就是它不會記憶來自用戶端的任何事情, 除非使用 Web-Based 提供的某些機制來記錄 (Session, ViewState), 而且之前所產生的資訊, 在經過了網頁來回後, 也不會再存在, 先前所做的不論是資料或是控制項生成, 全都會消失。

這也就是說, 利用動態生成的控制項, 必須要在頁面生成時都被建立起來, 才可以繼續保存在頁面上, 包括處理動態生成控制項的事件時, 若控制項不存在於頁面時, 則先前被引發的事件就會被取消而無法發生任何作用。

因此, 在 ASP.NET 網頁中若要使用動態生成控制項機制, 則有幾件事情需要考量：

1. **控制項建立：**在每次提交 Request 到 Web Server 時, 都需要在頁面載入時期, 將控制項重新生成, 讓 ASP.NET 知道有這個控制項存在, 否則無法在資料載入時期將資料套用到控制項中。

2. **控制項命名**：控制項在生成之時，都需要設定控制項的 ID，通常在大量的控制項生成時，ID 數量會變得許多，在設計上通常以具規則性的方式來做，才可以在後續處理上方便許多，否則只會讓處理工作更複雜。
3. **控制項資料存取**：ASP.NET 在執行完成頁面載入之後，就會開始設定資料，將資料設定給控制項，但若在控制項生成階段未產生控制項時，這個動作就會被忽略掉。
4. **控制項資料與狀態保存**：在 ASP.NET 輸出資料之後，所有控制項的資料都會釋放掉，開發人員需要利用一些方法或機制來保存狀態與資料，狀態的控制項在複雜的應用程式是個較不好處理的工作。

如果排除一些事件與無狀態考量的話，其實動態建立控制項是很簡單的。

```
Button cmdSubmit = new Button(); // 建立 Button 物件
cmdSubmit.ID = "cmdSubmit";

// 建立事件處理常式
cmdSubmit.OnClick += new EventHandler(cmdSubmit_Click);

cmdSubmit.Text = "送出";
Page.Controls.Add(cmdSubmit); // 將動態建立的控制項加入 Page 中。
```

不過因為前面有說到，動態生成的控制項在經過了網頁來回後，不會再存在，所以需要在每次頁面載入時就要建立出來，亦即不需要判斷 **PostBack**，如此 **ViewState** 以及 **PostBack** 訊息才會找的到控制項，也才會處理指定的事件（即呼叫事件處理常式）。

但是頁面載入有 **Page_Load** 和 **Page_Init** 兩個地方，要放在哪裡呢？微軟的 **MSDN Library** 中有一篇文章「**Creating Dynamic Data Entry User Interfaces**」，建議動態生成控制項的程式碼要放在 **Page_Init**，然後在 **Page_Load** 設定初值，並且要判斷 **Page.IsPostBack**（否則填入的資料會被預設值蓋掉）。

```
public class _default : System.Web.UI.Page
{
    protected void Page_Init(object sender, EventArgs e)
    {
        // 動態生成控制項的程式碼放這裡
    }

    protected void Page_Load(object sender, EventArgs e)
    {
        if (!Page.IsPostBack)
        {
            // 動態生成控制項的預設值設定放這裡
        }
    }
}
```

例如：

```
public class _default : System.Web.UI.Page
{
    protected void Page_Init(object sender, EventArgs e)
    {
        TextBox textbox = new TextBox();
        textbox.ID = "myTextBox";
        this.PlaceHolder1.Controls.Add(textbox);
    }

    protected void Page_Load(object sender, EventArgs e)
    {
        if (!Page.IsPostBack)
        {

```



```
        TextBox textbox =  
            this.PlaceHolder1.FindControl("myTextBox") as TextBox;  
        textbox.Text = "default value";  
    }  
}  
}
```

筆者以 **Q23：如何使用用戶端指令碼動態產生控制項** 問題中所使用的問卷為例，展示利用伺服端的解決方案。

解決方案

首先，先將問卷內容以及題型答案等，建立一個 XML 格式的資料，並將使用者的答案也建立一個 XML 清單的資料，讓資料可以儲存並傳給伺服器，範例資料與範例回應資料請參閱「如何使用用戶端指令碼動態產生控制項」中的程式 1 與程式 2，在此不再列舉。

接著，在 Page_Init 事件常式中加入動態建立控制項的程式碼，如程式 1 所示。

程式 1：Page_Init 中，動態生成控制項的程式碼

```
protected void Page_Init(object sender, EventArgs e)  
{  
    // 建立問題清單  
    XmlDocument doc = new XmlDocument();  
    doc.LoadXml(this.QContent.DocumentContent);  
    XmlNodeList nodes =  
        doc.DocumentElement.SelectNodes("//Questions/Question");  
    XmlNodeList answers = null;  
    Panel panel = null;  
  
    foreach (XmlNode node in nodes)  
    {
```

```
panel = new Panel();
panel.ID = "Q" + node.Attributes.GetNamedItem("ID").Value;
panel.Width = new Unit(100, UnitType.Percentage);

Label labelCaption = new Label();
labelCaption.Text = node.Attributes.GetNamedItem("ID").Value +
    ". " + node.Attributes.GetNamedItem("Caption").Value +
    "<br><br>";
panel.Controls.Add(labelCaption);

switch (node.Attributes.GetNamedItem("ResponseType").Value)
{
    case "SingleChoice":

        answers = node.SelectNodes("AnswerList/Answer");

        for (int i = 0; i < answers.Count; i++)
        {
            RadioButton opt = new RadioButton();
            opt.ID = "O" + node.Attributes.GetNamedItem("ID").Value +
                i.ToString();
            opt.GroupName =
                "O" + node.Attributes.GetNamedItem("ID").Value;
            opt.Text =
                answers[i].Attributes.GetNamedItem("Caption").Value;
            panel.Controls.Add(opt);
        }

        break;
```

```
case "MultipleChoice":

    answers = node.SelectNodes("AnswerList/Answer");

    for (int i = 0; i < answers.Count; i++)
    {
        CheckBox chk = new CheckBox();
        chk.ID = "C" + node.Attributes.GetNamedItem("ID").Value +
            i.ToString();
        chk.Text =
            answers[i].Attributes.GetNamedItem("Caption").Value;
        panel.Controls.Add(chk);
    }

    break;

case "MultipleText":

    TextBox text = new TextBox();
    text.ID = "M" + node.Attributes.GetNamedItem("ID").Value;
    text.TextMode = TextBoxMode.MultiLine;
    text.Width = new Unit(400, UnitType.Pixel);
    text.Height = new Unit(200, UnitType.Pixel);
    panel.Controls.Add(text);

    break;

default:
    break;
}
```

```
        Literal moreLine = new Literal();
        moreLine.Text = "<br><br>";
        panel.Controls.Add(moreLine);

        this.placeQuestionnaireCtrls.Controls.Add(panel);
    }

    doc = null;
}
```

接著, 撰寫處理接收結果的程式碼, 如程式 2 所示。

程式 2：處理使用者填入資料的程式碼

```
protected void cmdSubmit_Click(object sender, EventArgs e)
{
    // 由回答的 XML 檔中取得使用者填寫的回答資料
    XmlDocument doc = new XmlDocument();
    XmlDocument docAnswer = new XmlDocument();
    doc.LoadXml(this.QContent.DocumentContent);
    docAnswer.LoadXml(this.QAnswer.DocumentContent);
    XmlNodeList nodes =
        doc.DocumentElement.SelectNodes("//Questions/Question");

    foreach (XmlNode node in nodes)
    {
        Panel panel = Page.FindControl("Q"
            + node.Attributes.GetNamedItem("ID").Value) as Panel;

        switch (node.Attributes.GetNamedItem("ResponseType").Value)
        {

```

```
case "SingleChoice":

    foreach (Control ctl in panel.Controls)
    {
        if (ctl.GetType() == typeof(RadioButton))
        {
            RadioButton opt = ctl as RadioButton;

            if (opt.Checked)
                docAnswer.SelectSingleNode("//Responses/Response[@ID=' "
                    + node.Attributes.GetNamedItem("ID").Value
                    + "']").InnerText = opt.Text;
        }
    }

    break;

case "MultipleChoice":

    string list = null;

    foreach (Control ctl in panel.Controls)
    {
        if (ctl.GetType() == typeof(CheckBox))
        {
            CheckBox opt = ctl as CheckBox;

            if (opt.Checked)
                list += (list == null) ? opt.Text : ", " + opt.Text;
        }
    }
```

```
docAnswer.SelectSingleNode("//Responses/Response[@ID=' "
    + node.Attributes.GetNamedItem("ID").Value
    + "']").InnerText = list;

break;
case "MultipleText":

    foreach (Control ctl in panel.Controls)
    {
        if (ctl.GetType() == typeof(TextBox))
        {
            TextBox textBox = ctl as TextBox;

            docAnswer.SelectSingleNode("//Responses/Response[@ID=' "
                + node.Attributes.GetNamedItem("ID").Value
                + "']").InnerText = textBox.Text;
        }
    }

    break;

default:
    break;
}

docAnswer = null;
doc = null;
}
```

讀者可以將本篇問題的程式寫法和「如何使用用戶端指令碼動態產生控制項」問題中的寫法比比看,筆者認為在伺服器使用動態生成控制項要寫的程式碼,反而會比在用戶端使用動態生成要來的多一些,因為伺服端的程式碼缺乏用戶端指令碼那樣的彈性,以及動態的呈現 (Rendering) 能力,而且在伺服器讀取動態生成控制項所裝載的資料時,如果控制項結構愈複雜,要讀取的程式碼也會跟著變多,這也是伺服器程式缺乏彈性所造成的。

因此,若真的非必要,或是只有少量的動態生成控制項的考量下,才使用伺服器動態生成控制項,若是大量的動態控制項生成,則可以考慮使用用戶端的動態控制項生成,並搭配 AJAX 的機制,如此在發展這類型的應用程式時,才會比較輕鬆 (用戶端指令碼和伺服器程式碼,筆者覺得用戶端指令碼比較好用 😊)。

Q25

如何動態產生圖片？

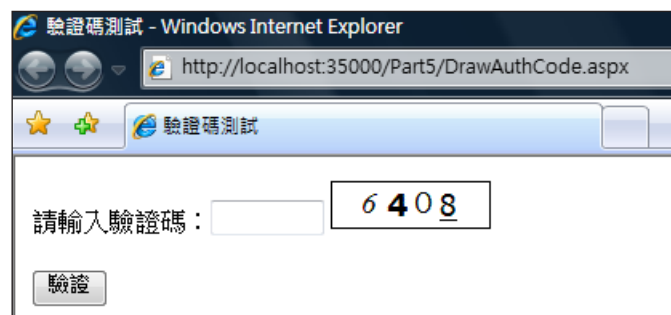
適用範圍： ☒ ASP.NET 1.0 ☒ ASP.NET 1.1 ☒ ASP.NET 2.0 ☒ ASP.NET 3.5

問題

公司最近有一個專案,是要依照給定的資料來畫出圖形,資料格式已經確定了,但老闆說要能夠在 Web 上面畫出圖片,讓瀏覽器可以直接看到繪製完成的圖片,請問如何利用 ASP.NET 繪製圖片？

問題說明

這種問題其實也是一個常見的問題,企業內有時候會因為一些繪圖需求 (例如報表或圖表),需要在網站上自動產生一些圖片,圖表的話有時可以利用像 OWC (Office Web Component) 或其他廠商發展的 Chart 元件來處理,但是如果是一些特殊圖形,就需要自己設計繪圖指令來做,最常見的就是最近為了防止重覆執行指令的機器人 (Agent) 所發展出來的圖片驗證碼：



這種驗證碼其實是很簡單的繪圖應用,只是為了防止機器人有掃描並辨識圖片中文字的能力,因此會做到變化字形或是加入會影響辨識的圖案,讓驗證碼發揮真正應有的功效 (圖片驗證碼的部份請參閱「如何使用驗證碼來防制大量重覆指令」問答)。

.NET Framework 中具有繪圖與影像處理能力的指令，都包裝在 System.Drawing、System.Drawing.Drawing2D 與 System.Drawing.Imaging 命名空間中，其常用的類別有：

類別	說明
Image 類別	影像格式的基底類別。
Bitmap 類別	適用於點陣圖以及 GIF 等圖檔。
Graphics 類別	執行繪圖指令。
Font 類別	定義字型。
Pen, Brush 類別	分別代表畫筆及筆刷。
Point, PointF 類別	代表點的位置（有 F 表示單位為浮點數）。
Size, SizeF 類別	代表尺寸大小（有 F 表示單位為浮點數）。
Rectangle, RectangleF 類別	定義四邊形（有 F 表示單位為浮點數）。
LinearGradientBrush 類別	提供線性漸層筆刷。
PathGradientBrush 類別	提供路徑漸層筆刷。

通常要繪製一張圖，必須要先建立好繪圖空間，在以往 Windows 程式設計中，都要先建立 DC (Device Context) 或點陣圖 (Bitmap)，再於 DC 中處理繪圖指令，在 .NET Framework 中，簡單的繪圖空間可由 Bitmap 類別來處理，因此，若要建立一個指定大小的繪圖空間，可以用下列指令來處理。

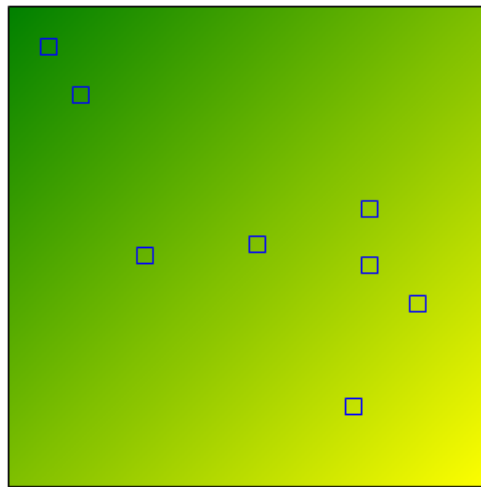
```
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Drawing.Imaging;
using System.IO;

...
```

```
// 建立一個寬 300px, 高 300px 的繪圖空間
Bitmap bmp = new Bitmap(300, 300);

// 在繪圖空間中建立繪圖指令物件
Graphics g = Graphics.FromImage(bmp);
```

然後利用 **Graphics** 提供的繪圖指令, 來處理繪圖的動作, 例如繪出具有漸層效果的底圖, 並且在指定的點上, 繪出藍色的小方框 (10px * 10px)。其繪出的結果如下圖所示。



```
// 定義繪圖空間範圍的 Rectangle 物件。
Rectangle rect = new Rectangle(0, 0, bmp.Width, bmp.Height);
// 產生漸層筆刷並填入漸層效果到繪圖空間。
g.FillRectangle(
    new LinearGradientBrush(rect, Color.Green, Color.Yellow, 45),
    rect);
// 繪製黑色邊框。
g.DrawRectangle(Pens.Black,
    new Rectangle(0, 0, bmp.Width - 1, bmp.Height - 1));
// 使用藍色畫筆繪製小方框。
```

```
g.DrawRectangle(Pens.Blue, 20, 20, 10, 10);  
g.DrawRectangle(Pens.Blue, 40, 50, 10, 10);  
g.DrawRectangle(Pens.Blue, 80, 150, 10, 10);  
g.DrawRectangle(Pens.Blue, 220, 121, 10, 10);  
g.DrawRectangle(Pens.Blue, 150, 143, 10, 10);  
g.DrawRectangle(Pens.Blue, 220, 156, 10, 10);  
g.DrawRectangle(Pens.Blue, 210, 244, 10, 10);  
g.DrawRectangle(Pens.Blue, 250, 180, 10, 10);
```

完成繪圖工作時, 由於要轉成二進位資料流才能夠輸出到用戶端, 因此要先將圖片存入記憶體資料流 (Memory stream) 中, 然後透過 `Response.BinaryWrite()` 輸出到用戶端即可。

```
MemoryStream ms = new MemoryStream();  
bmp.Save(ms, ImageFormat.Png); // 存成 PNG 格式  
ms.Flush(); // 清空緩衝區。  
byte[] data = ms.ToArray(); // 匯出資料到位元組陣列中。  
  
Response.ContentType = "image/png";  
Response.BinaryWrite(data); // 輸出到用戶端。
```

這個方法不但適用於一般繪圖, 也可以適用於縮圖處理, 差異只在繪圖區域大小的變換而已。

🔧 解決方案

可新增一個 HTTP Handler, 然後將必要的繪圖指令碼與資料都放到 HTTP Handler 中, 由它來執行繪圖的工作。使用 HTTP Handler 繪圖的程式碼如程式 1 所例示。

程式 1：使用 HTTP Handler 執行繪圖指令的範例程式

```
public class DrawingChart : IHttpHandler {

    public void ProcessRequest (HttpContext context) {

        MemoryStream ms = new MemoryStream();
        Bitmap bmp = new Bitmap(300, 300);

        bmp.SetResolution(600, 600);

        Graphics g = Graphics.FromImage(bmp);

        g.InterpolationMode = InterpolationMode.High;
        g.SmoothingMode = SmoothingMode.HighQuality;

        Rectangle rect = new Rectangle(0, 0, bmp.Width, bmp.Height);
        g.FillRectangle(
            new LinearGradientBrush(rect, Color.Green, Color.Yellow, 45),
            rect);
        g.DrawRectangle(Pens.Black,
            new Rectangle(0, 0, bmp.Width - 1, bmp.Height - 1));

        g.DrawRectangle(Pens.Blue, 20, 20, 10, 10);
        g.DrawRectangle(Pens.Blue, 40, 50, 10, 10);
        g.DrawRectangle(Pens.Blue, 80, 150, 10, 10);
        g.DrawRectangle(Pens.Blue, 220, 121, 10, 10);
        g.DrawRectangle(Pens.Blue, 150, 143, 10, 10);
        g.DrawRectangle(Pens.Blue, 220, 156, 10, 10);
        g.DrawRectangle(Pens.Blue, 210, 244, 10, 10);
        g.DrawRectangle(Pens.Blue, 250, 180, 10, 10);
```

```
        bmp.Save(ms, ImageFormat.Png);

        g.Dispose();
        bmp.Dispose();

        ms.Flush();
        context.Response.ContentType = "image/png";
        context.Response.BinaryWrite(ms.ToArray());
        ms.Close();
    }

    public bool IsReusable {
        get {
            return false;
        }
    }
}
```

考生停看聽

本主題所討論的內容，可用以準備 Exam 70-528: TS: Microsoft .NET Framework 2.0 Web Application 考試的主題：

- 建立與設計 Web 應用程式：
 - 撰寫 ASP.NET Handler 以動態產生需要顯示在網頁上的影像。

相關問題

- Q47：如何將圖檔儲存在資料庫中？

Q26

如何在網頁中讓使用者可以直接編修 HTML 格式的文件或內容？

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

❓ 問題

最近公司在發展 EDM (電子廣告) 的系統, 要用 Email 的方式來發送由行銷部提供的 DM, 行銷部提了一個需求, 想直接在應用程式上面編修 DM 的內容, 但這個內容是 HTML 的格式, 我要如何做, 才可以讓使用者直接在應用程式中編修 HTML 文件？

💡 問題說明

這個需求筆者相信一定有很多應用程式開發人員想要做, 但有時卻苦無資源可以用, 殊不知網路上已經有一票付費或免費的 HTML 編修元件可以使用了, 這個功能也散見於各大討論區, 功能都大同小異, 且均可編修 HTML 的效果, 但開放的功能有小部份差異而已。

直接在網頁上編修 HTML 在早期瀏覽器還是 IE4.0 以及 5.0 時, 其實就有了, 這種能力稱為 DHTML Editing, 現在稱為 Scriptable Editing, 主要是用在瀏覽器直接編輯 HTML 內容, 以及所見即所得 (What You See, What You Get) 功能, 由於這樣可以讓使用者馬上看到資料的編修效果, 頗受網頁開發人員的歡迎, 但因為要實作這種編輯器的難度較高 (圖片以及指令碼撰寫), 因此在推出的當時並沒有馬上風行, 而是隔了幾年後才開始流行。

目前在網路上有大約 94 種 HTML 編輯器 (查自 <http://www.htmlarea.com>), 其中有部份是要付費的, 但是付費版本的功能都比較強, 而且還支援數種的 CSS 效果, 像是 Office-Style、Windows-Style 等等不同的視覺效果, 並且有提供技術支援服務, 以及完整的線上說明等, 但費用大約在 100-200 美元上下 (當然有便宜的也有貴的)。而免費的多是供下載, 並且提供技術文件而已, 沒有實質上的支援, 若需要支援則要付費, 否則就要在網路上尋找看看有沒有其他人有使用經驗, 並且有分享使用心得與方法的說明。

接下來, 筆者以自己常用的 HTML 編輯器 – FCKeditor 來示範簡單的 HTML 編輯器建置。

🔧 解決方案

1. 到 FCKeditor 的網站 (<http://www.fckeditor.net/>) , 下載兩個元件, 一個是 FCKeditor 的主程式, 另一個是 FCKeditor.NET, 這個是給 ASP.NET 應用程式的元件。

FCKeditor 2.5.1

ZIP File (source code included): [FCKeditor_2.5.1.zip](#)
 GZip File (source code included): [FCKeditor_2.5.1.tar.gz](#)

Nightly Build

Generated every night at 0:30 PST from our [SVN](#):

[FCKeditor_N.zip](#) or [FCKeditor_N.tar.gz](#)

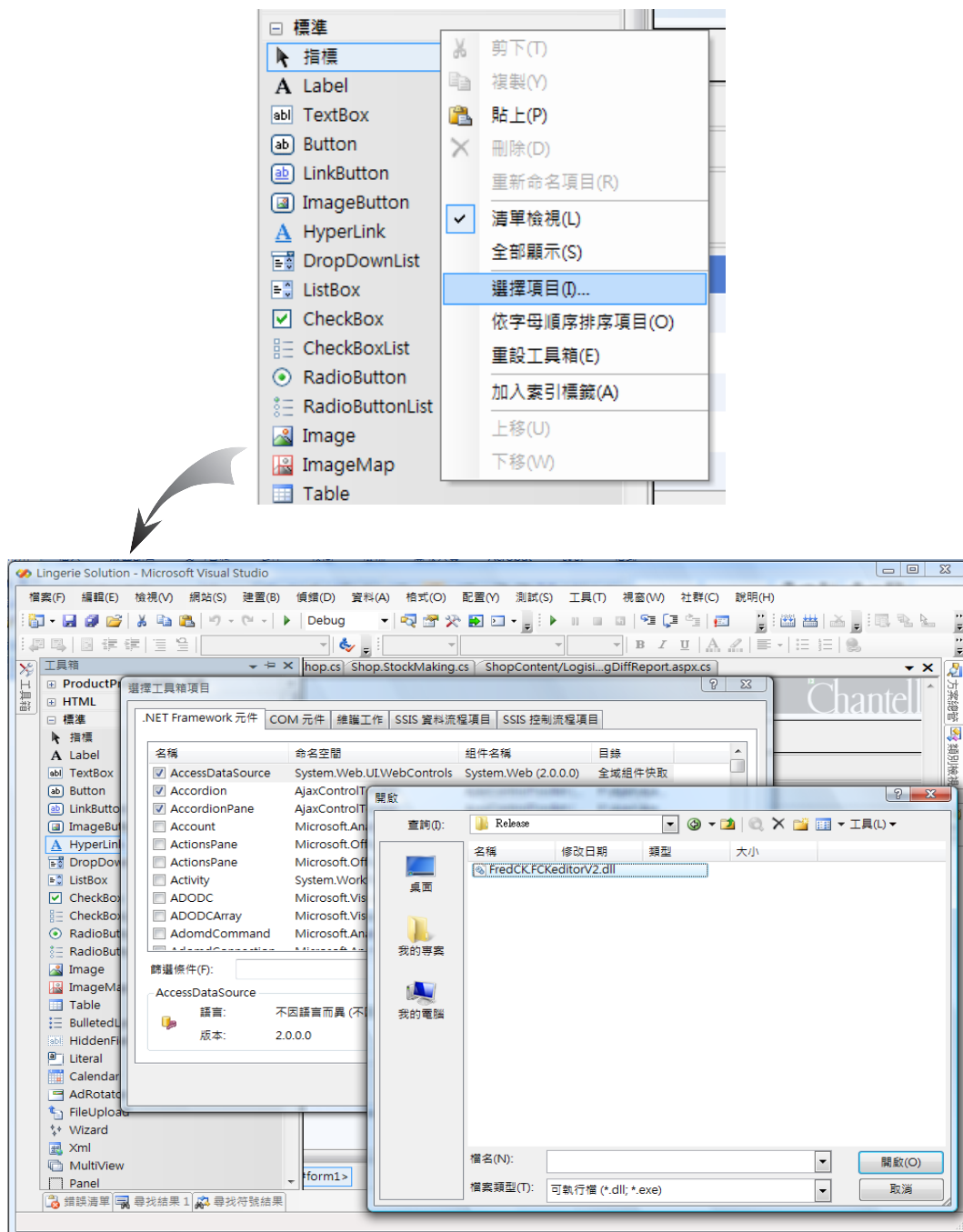
Attention: The above nightly contains the "under development" version of FCKeditor, usually stable, but not deeply tested. You can also [test it online](#).

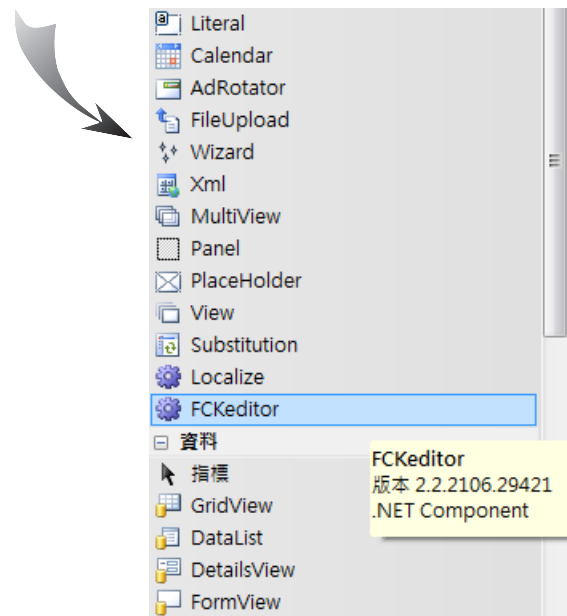
FCKeditor.Net

ASP.Net Control to easily integrate FCKeditor on .Net web pages.
[Click here to download the latest version](#)

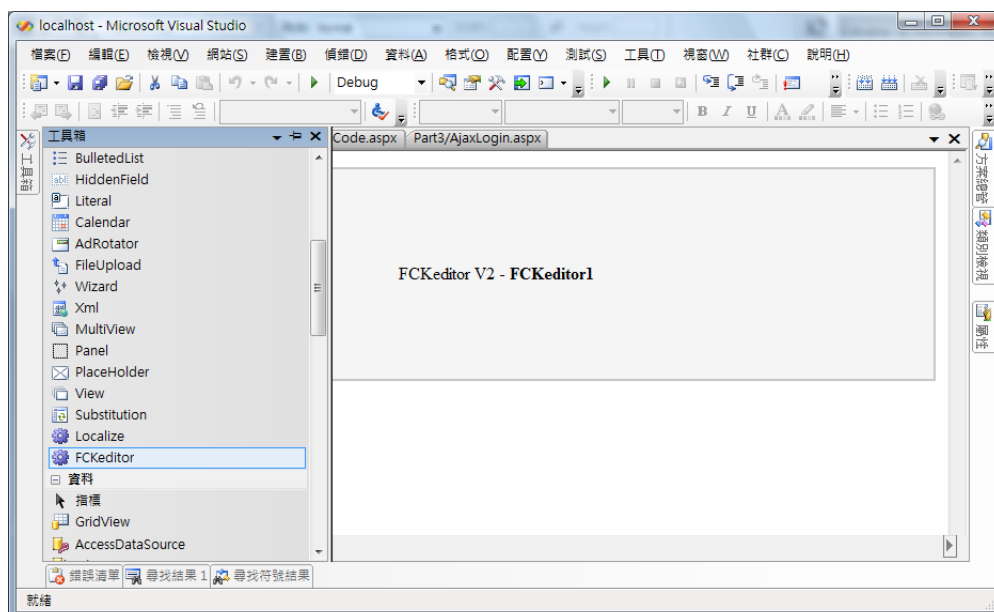
Attention: The main code is not included in this package, just the ASP.Net Control. You still need the FCKeditor scripts to be able to run it, so download it too (the first link in this page).

2. 將下載到的 FCKeditor_2.5.1.zip 解壓縮到網站的根目錄中 (解壓完成後, 根目錄會出現 FCKeditor 的子目錄) , 而 FCKeditor 的 ASP.NET 元件, 只要解壓縮到平常在使用的目錄即可, 它不必要和 FCKeditor 在一起, 但因為它會呼叫 FCKeditor 的主程式, 因此主程式要複製到網站的根目錄。
3. 開啓 Visual Studio, 在工具箱中選擇項目 , 並且瀏覽到先前解壓縮的 FCKeditor.NET 目錄, 將 FredCK.FCKeditorV2.dll 加入, 然後關閉視窗, 如此工具箱會出現 FCKeditor 的元件。

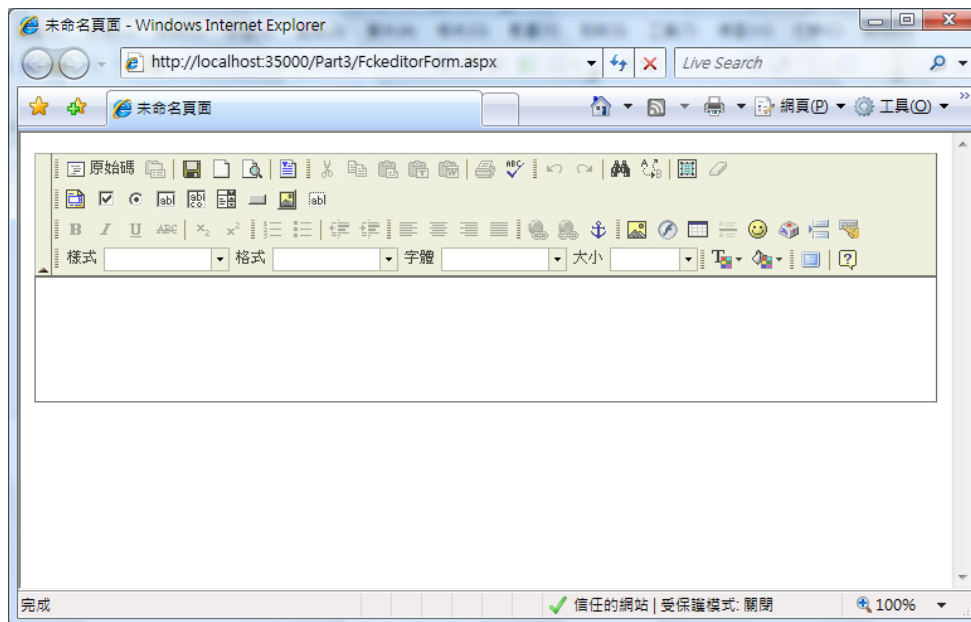




4. 將 FCKeditor 拖曳到網頁中, 並且設定其屬性。



5. 執行這個網頁, 就可以使用了。



若要是使用 HTML 的編修, 在 ASP.NET 2.0 必須要在網頁的檔頭中加入 `ValidateRequest="false"` 來關閉對 HTML 的檢查功能, 否則一傳送就會發生錯誤。另外, 若是在 ASP.NET 程式開發伺服器測試, 則要記得把 `BasePath` 修改為 `"~/FCKeditor"`, 否則在瀏覽時會出現 404 的錯誤訊息。

小常識 自訂 FCKeditor 的工具列

預設的 FCKeditor 工具列是很長的, 並且有一些選項可能開發人員會認為是多餘的, 因此開發人員可以藉由編輯 FCKeditor 的 `fckconfig.js` 來自訂自己的工作列按鈕, 讀者可以打開位在 FCKeditor 目錄下的 `fckconfig.js`, 並且找到工具列的設定：

接下頁

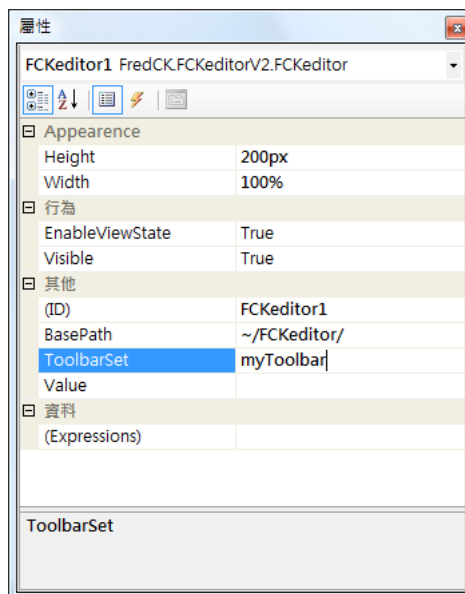
```
FCKConfig.ToolbarSets["Default"] = [
    ['Source', 'DocProps', '-', 'Save', 'NewPage', 'Preview', '-',
     'Templates'],
    ['Cut', 'Copy', 'Paste', 'PasteText', 'PasteWord', '-', 'Print',
     'SpellCheck'],
    ['Undo', 'Redo', '-', 'Find', 'Replace', '-', 'SelectAll',
     'RemoveFormat'],
    ['Form', 'Checkbox', 'Radio', 'TextField', 'Textarea', 'Select',
     'Button', 'ImageButton', 'HiddenField'],
    '/',
    ['Bold', 'Italic', 'Underline', 'StrikeThrough', '-', 'Subscript',
     'Superscript'],
    ['OrderedList', 'UnorderedList', '-', 'Outdent', 'Indent'],
    ['JustifyLeft', 'JustifyCenter', 'JustifyRight', 'JustifyFull'],
    ['Link', 'Unlink', 'Anchor'],
    ['Image', 'Flash', 'Table', 'Rule', 'Smiley', 'SpecialChar',
     'PageBreak',
     'UniversalKey'],
    '/',
    ['Style', 'FontFormat', 'FontName', 'FontSize'],
    ['TextColor', 'BGColor'],
    ['FitWindow', '-', 'About']
];
```

這是預設的工具列宣告，每一個工具列群組由 "[" 和 "]" 指定，而 "/" 代表分行，"-" 代表分開各個小群組的設定，功能項由 "," 來分隔，並且每個功能項都有固定的代字，這個代字表可以在 FCKeditor 的線上文件中找到，這種宣告方式是 JSON 格式 (JavaScript Object Notation)，讀者可以對照畫面中的工具列和程式設定來看，並且設定出自己的工具列。

例如, 要設定出只要有文字編修以及字型控制的功能的工具列, 如下程式所示 :

```
FCKConfig.ToolbarSets["myToolbar"] = [
    ['Cut', 'Copy', 'Paste', 'PasteText', 'PasteWord', '-', 'Print',
     'SpellCheck'],
    ['Undo', 'Redo', '-', 'Find', 'Replace', '-', 'SelectAll',
     'RemoveFormat'],
    '/',
    ['Style', 'FontFormat', 'FontName', 'FontSize'],
    ['TextColor', 'BGColor']
]
```

改完後, 要記得更改 **FCKeditor** 的 **ToolbarSet** 設定, 改為自己的工具列集, 讀者可以在 **Visual Studio** 中設定 :



或者在程式中設定亦可 :

```
// ContentEditor 是 FCKEditor 控制項
this.ContentEditor.ToolbarSet = "myToolbar";
```

Q27

如何將 ASP.NET 2.0 的會員資料存在現有的資料庫中？

適用範圍： ☒ ASP.NET 2.0 ☒ ASP.NET 3.5

? 問題

ASP.NET 2.0 中有一個會員管理的功能, 能夠自動的對會員做管理, 以及和登入控制項做結合等, 可是我所閱讀到的文件都是由空的資料庫開始, 但我已經有一個會員資料庫了, 我要怎麼把會員管理功能和我的資料庫連接？

! 問題說明

Membership Services (會員服務, 位於 System.Web.Security 命名空間中) 是 ASP.NET 2.0 才新增的功能, 它可以讓開發人員在最短的時間內就開發出會員管理的服務, 並且像是 Login 與 CreateUserWizard 等控制項, 都直接支援了 Membership Services, 有內定的資料結構以及方法可使用, 在前端是利用 Membership 類別, 而在資料端則是用 MembershipProvider 抽象類別 (abstract class) 的實作, 來連接到資料來源, 這個服務是利用 Provider Pattern 來實作的, 有點像是 Plug-in 的方法。

目前在 ASP.NET 2.0 中有二個內建的 Membership Provider：

- SqlMembershipProvider: 連接 SQL Server 資料庫中的會員資料表。
- ActiveDirectoryMembershipProvider: 連接 ActiveDirectory 的使用者資料。

其中, SqlMembershipProvider 最常用, 它會以 ADO.NET 的方式連到 SQL Server 的特定資料庫與資料表, 來取出與寫入會員的資料, 若使用的是 SQL Server Express, 則會更方便, 因為資料檔就直接建立好供開發人員使用。

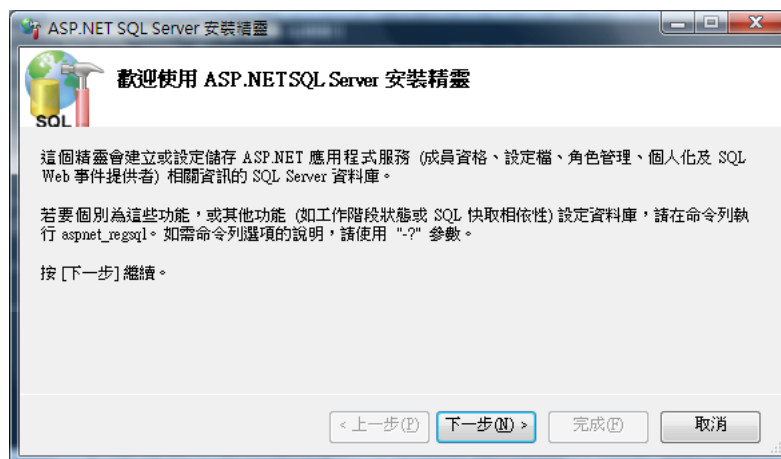
小常識 設定 SQL Server 以支援 Membership Services

Membership Service 在開發人員的本機有安裝 SQL Server Express 時, 會自動啟動, 並且在 App_Data 目錄中加入一個 aspnetdb.mdf 的檔案, 在裡面就有支援 Membership Services 所要的各式資料表。但若開發人員安裝的是 SQL Server, 而非 SQL Server Express 時, 就要做一點設定才行。

.NET Framework 有提供一個處理 SQL Server 支援會員資料的設定的工具 – aspnet_regsql.exe, 開發人員可以給定參數, 讓這個工具建立支援 ASP.NET 特殊服務 (包含 Membership Service, Role Service, Web Part, Web Profile, Web Events 等) 的資料表結構。它的用法很簡單：

```
aspnet_regsql -S "localhost" -E -A m -d "MyDB"
```

這個指令表示要在伺服器 (-S) localhost 中, 以目前使用者帳戶做驗證 (-E), 並且在資料庫 MyDB (-d) 中, 建立會員服務的資料表 (-A m), 若不下任何參數的話, 會出現圖形介面視窗, 方便開發人員設定：



透過使用者介面方式, 可一次安裝 ASP.NET 的應用程式服務所需要的資料表, 但無法設定個別功能, 以及 Session State 與 Cache Dependency 的 SQL Server 支援, 那要由指令參數來設定。也就是說, 透過使用者介面的設定, 如同下了 aspnet_regsql.exe -A all 參數的指令。

更詳細的參數列表, 可以到 .NET Framework 文件中, 查詢 aspnet_regsql, 或在 Google 等搜尋引擎中, 以 aspnet_regsql 作關鍵字搜尋。

但問題來了, 多數的應用程式其實都有自己的資料庫以及資料結構, 若是又要多一個 aspnetdb 資料庫, 會讓資料整合的問題浮現, 例如 :

- 新會員如何和原有的資料整合 ?
- 舊會員如何和新的資料結構整合 ?
- 舊會員如何享受新的服務的功能 ?
- 若有額外的資料結構, 應該要如何儲存 ?
- ...

因此, 若想要讓 Membership Services 能夠和自己的資料庫做結合, 就需要自己建立一個 Membership Provider, 也就是要實作 MembershipProvider 抽象類別中的所有方法, 然後存取自己的資料庫, 以取出和自己的資料庫相同的資訊。

若要實作 Membership Provider, 則要先建立一個類別 (建議使用類別庫), 然後宣告這個類別繼承 MembershipProvider :

```
public class MyProvider : Membership Provider
{
}
```

接著, 利用 override 的方法, 來覆寫在 Membership Provider 中所開放的屬性與方法 :

```
if (machineKey.ValidationKey.Contains("AutoGenerate"))
    if (PasswordFormat != MembershipPasswordFormat.Clear)
        throw new ProviderException("Hashed or Encrypted passwords " +
            "are not supported with auto-generated keys.");
}

override |
// A
// A
private {
    Name { get; }
    OnValidatingPassword (System.Web.Security.ValidatePasswordEventArgs e)
    ToString ()
    return configValue;
}

//
// System.Web.Security.MembershipProvider properties.
//
```

若要完整的實作出一個 **Membership Provider**, 則必須要至少實作出 8 個屬性與 17 個方法, 才能夠滿足基本的功能, 這些屬性和方法為：

屬性	方法
EnablePasswordReset (true/false)	Initialize()
EnablePasswordRetrieval (true/false)	CreateUser()
RequiresQuestionAndAnswer (true/false)	UpdateUser()
RequiresUniqueEmail (true/false)	DeleteUser()
PasswordFormat (true/false)	ValidateUser()
MaxInvalidPasswordAttempts (int)	GetUser() * 2, 不同的參數實作。
PasswordAttemptWindow (int)	GetAllUsers()
ApplicationName (string)	GetNumberOfUsersOnline()
	ResetPassword()
	GetUserNameByEmail()
	ChangePassword()
	ChangePasswordQuestionAndAnswer()
	FindUsersByName()
	FindUsersByEmail()
	UnlockUser()

每個方法都有提供參數, 開發人員需要利用這些參數來將資料儲存到資料庫中, 或者由資料庫提取, 然後放到 **MembershipUser** 物件中。

例如, **Membership** 利用 **Membership.GetUser()** 方法來取得 **MembershipUser** 物件以供後續執行使用, 所以開發人員必須要實作 **Membership.GetUser()** 方法, 如下列 **GetUser** 的實作範例：

程式碼 1：MembershipProvider.GetUser() 的範例實作。

```
public override MembershipUser GetUser(string username,
    bool userIsOnline)
{
```



```
MembershipUser user = null;    // 取得自己的會員資料內容。

DataTable table = DataProvider.GetCustomerInfo(username);

// 將會員資料填入 MembershipUser 物件中。
if (table != null)
{
    user = new MembershipUser(
        "CustomProvider", table.Rows[0]["Name"].ToString(),
        table.Rows[0]["CustomerID"].ToString(),
        table.Rows[0]["Email"].ToString(),
        table.Rows[0]["PwdQuestion"].ToString(),
        table.Rows[0]["Comments"].ToString(),
        true, false, new DateTime(2007, 1, 1, 0, 0, 0),
        DateTime.Now, DateTime.Now,
        Convert.ToDateTime(table.Rows[0]["LastPwdChangeDate"].ToString()),
        Convert.ToDateTime(table.Rows[0]["LastLockedDate"].ToString()));

    table = null;
    return user;
}
```

除了 **MembershipProvider** 的固定方法實作外, 開發人員也可以加入一些額外的方法, 以支援其他類型的資料存取工作, 例如 :

程式碼2：自訂方法。

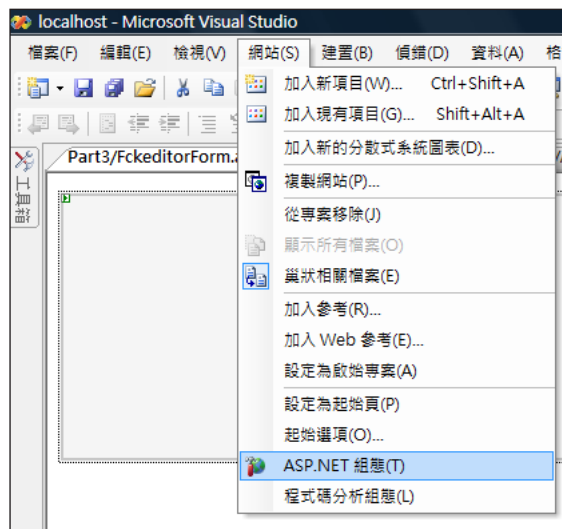
```
// 在 Membership Provider 自訂實作的程式中
public void UpdateProperties(
    MembershipUser user, string Company, string Professions,
    string SocialIDNum, string Cellphone)
{
    DataProvider.UpdateProperties(
        user.ProviderUserKey, Company, Professions,
        SocialIDNum, Cellphone);
}
```

```
// 在呼叫端
CustomProvider cp = Membership.Provider as CustomProvider;
cp.UpdateProperties(
    user, this.Company.Text, this.Professions.Text,
    this.SocialIDNum.Text, this.Cellphone.Text);
```

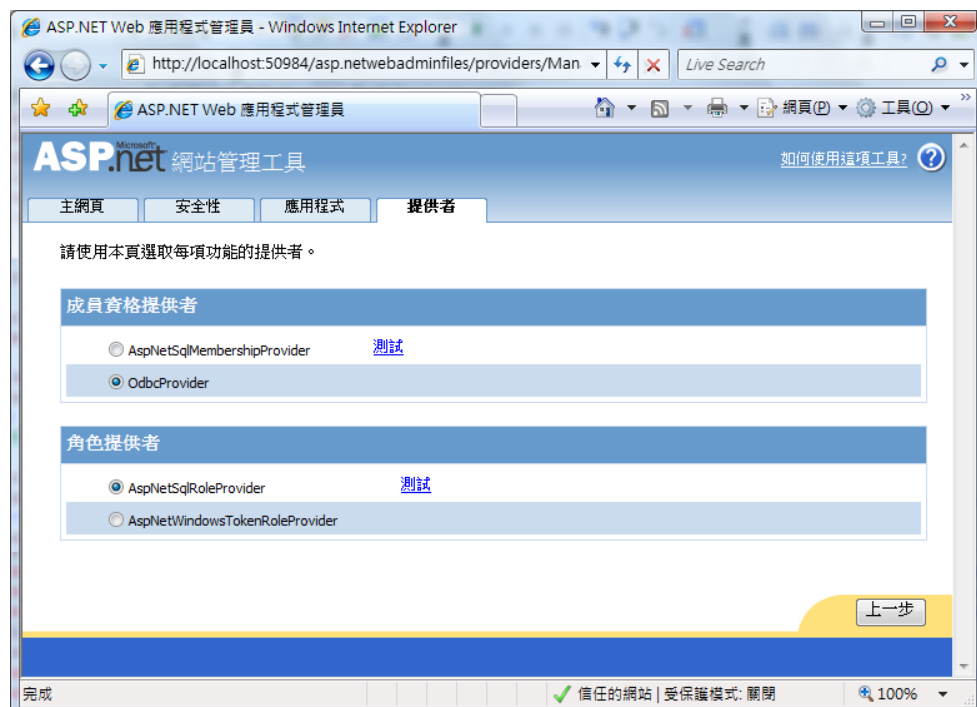
當完成實作自訂的 Membership Provider 後,就要在 Web.config 中設定,讓它能夠被 ASP.NET 執行引擎取用,以微軟.NET Framework SDK 提供的 OdbcMembershipProvider 為例:

```
<membership defaultProvider="OdbcProvider"
    userIsOnlineTimeWindow="15">
    <providers>
        <add
            name="OdbcProvider" // 自訂的 Membership Provider 名稱。
            type="OdbcMembershipProvider" // 自訂 Membership Provider 的型別
            // Membership Provider 使用的連線字串
            connectionStringName="OdbcServices"
            enablePasswordRetrieval="true"
            enablePasswordReset="true"
            requiresQuestionAndAnswer="true"
            writeExceptionsToEventLog="true" />
        </providers>
    </membership>
```

若想要測試看看自訂的 Membership Provider 有沒有成功連結資料庫,可以利用 Visual Studio 來呼叫『**ASP.NET 網站管理工具**』(Web Site Management Tool)來試著連接看看,首先,先啟動『**ASP.NET 組態**』(即網站管理工具)。



然後點選**提供者**頁籤, 若可以看到自訂的 Membership Provider 的名稱, 表示 Web.config 中的設定是可以找到自訂 Membership Provider 的 :



若無法看到自訂 Membership Provider 的名稱, 表示設定上可能有錯誤, 需要檢查 Web.config。

然後點選**安全性**頁籤, 若可以看到下列畫面, 表示自訂的 Membership Provider 已經可以連到資料庫：



若看到的不是這個畫面, 那表示自訂提供者無法連接資料庫, 或是在處理時發生了錯誤, 這時就要檢查程式中的問題。

🔧 解決方案

透過實作 MembershipProvider 抽象類別的方式, 將 Membership Services 和自己的資料庫做連接, 並且將撰寫好的 Membership Provider 設定加入 Web.config 中, 就可以在 ASP.NET 2.0 網站中使用了。

若需要測試, 可以在 ASP.NET 的網站管理工具中進行測試。

備註

本文所使用的範例 **Membership Provider – OdbcMembershipProvider**, 可以在微軟 MSDN 的網站中找到：

```
http://msdn2.microsoft.com/zh-tw/library/44w5aswa(VS.80).aspx
```

OdbcMembershipProvider 的實作範例程式碼：

```
http://msdn2.microsoft.com/zh-tw/library/317sza4k(VS.80).aspx
```

若想要知道更多關於自訂 **Membership Provider** 的訊息, 可參考：

```
http://msdn2.microsoft.com/zh-tw/library/f1kyba5e(VS.80).aspx
```

旗標出版社所出版的「微軟 MVP 的 ASP.NET 2.0 絕技」一書的第三章, 對 **Membership Provider** 的自訂有更深入說明。

小常識 ASP.NET 網站管理工具

為了簡化開發 ASP.NET 2.0 網站所需要的設定, 以及對於 ASP.NET 2.0 新功能的支援, 微軟特別在 Visual Studio 2005 中提供了一個可以設定 ASP.NET 2.0 網站設定的網站管理工具, 這個工具可以設定一些 ASP.NET 的新功能, 包含 **Membership**, **Role**, **Profile** 等等功能, 也可以設定支援這些服務的提供者 (**Provider**), 另外還有一個 ASP.NET 安全性設定精靈, 這個精靈可以幫助開發人員先期做一些像權限 (**URL 授權**) 與角色的設定, 也可以預先建立使用者 (若會員是來自網際網路的才可以建立)。

經驗談

真的有必要爲了使用 Membership 服務而自訂 Membership Provider 嗎？

雖然 Membership Service 內部和 Login, CreateUserWizard, LoginState 等控制項連接, 可以簡化程式設計的過程, 但是不是任何的資料庫都「適合」和 Membership Service 連接的, 例如原本沒有支援密碼重設, 或者是未記錄使用者上線時間的資料表, 可能就因爲要和 Membership Service 連結, 而被迫新增一些欄位或資料, 使得和現有程式的相容性就發生問題, 或是若想要在 Membership Service 中延伸一些屬性或方法, 也可能會因爲 Provider Pattern 先天的限制 (函式原型要固定, 包含參數和名稱) 之下, 會讓延長性受到限制, 此時可能自行設計登入的控制項處理, 反而會比自訂 Membership Provider 來的更容易而且更快。

例如, Login 控制項可以透過處理 Authenticate 事件常式來處理登入的程序, LoginStatus 可以依 Page.IsAuthenticated 來顯示登入的狀況, PasswordRecovery 和 ChangePassword 控制項則可以用少量的程式碼來實作, 唯一比較麻煩的只有 CreateUserWizard, 可利用 Wizard 控制項來模擬實作。

基本上, 在設計與發展應用程式時, 相容性和延展性是兩大主要考量之一, 如果只是爲了想要便宜行事 (用 ASP.NET 登入控制項組設計登入網頁) 的話, 很可能就必須犧牲掉相容性和延展性, 最後也有可能花了很多時間調整資料表結構和實作 Membership Provider, 但效益卻和預期差很多。

因此, 想要利用 Membership Service 帶來的好處前, 請先思考並評估目前的資料庫設計, 是否適合和 Membership Service 連結, 再決定是否要實作自己的 Membership Provider。

考生停看聽

本問題所討論的內容, 可用來準備 Exam 70-528: TS: Microsoft .NET Framework 2.0 Web Client Development 的下列主題：

- 建立與設計 Web 應用程式
 - 設定 Web 應用程式的組態。
 - 使用網站管理工具管理 Web 應用程式的組態設定。
- 實作驗證與授權
 - 使用表單驗證法建立使用者識別功能。
 - 使用會員服務 API 以及 Membership 類別來管理使用者。

Q28

如何動態切換 Master Page ?

適用範圍：  ASP.NET 2.0  ASP.NET 3.5

問題

我的公司的入口網站是我自己設計的, 使用的是 ASP.NET 2.0, 剛開始的時候是用自己設計的醜醜的版面, 用的是 Master Page (主版頁面) 和 Content Page (內容頁面) 來做編排, 後來公司請美工人員設計了一個較美觀的 Master Page, 現在看起來就比較好看一點。

但是最近公司想要依照不同的部門來設計不同的 Master Page, 這樣會變成我在程式上必須要依照不同部門或不同使用者, 來套用他們的 Master Page, 請問我要如何做?

問題說明

Master Page 是 ASP.NET 2.0 爲了要替代原本以 FRAME 爲主的網頁切割設計方式所發展出來的, 有設計過 ASP 或 PHP 的開發人員應該知道, 一般在設計具有一致性風格 (Consistency Style) 的網站時, 可能要東切一塊, 西切一塊, 然後用這樣的方法組合:

```
<!-- #include virtual="header.asp" -->
<!-- #include virtual="menu.asp" -->
```

這主要是因爲早期的 Web 程式設計平台沒有一致性的觀念, 以可以在 Web 上工作爲主, 所以比較沒有考量到一致性風格的功能。隨著 Web 的發展, 愈來愈多的公司或網站已經注意到, 需要有一致性的風格, 讓使用者比較不會眼花瞭亂, 而且對於企業的形象也有正面幫助。

當一開始在設計 Web 應用程式時, 可能就需要先定義好整個網站的 Layout (布局), 然後利用 **Master Page** 來定義好 Layout, 內容的部份就交給 **Content Page** 來顯示, 這樣可以讓開發人員專注在內容發展上, 外觀只要由 **Master Page** 來處理就行了。例如像是登入功能、功能表、共用的功能或是版權說明等, 都可以放在 **Master Page** 來處理。

而更大型的入口網站, 則會需要有多重的版面風格, 一來可讓使用者有不同的選擇, 二來可能視公司政策而定, 會用到二個以上的 **Master Page**, 此時, 開發人員就需要在程式中處理切換工作。

若想要用程式方式切換 **Master Page**, 則必須要在內容頁的程式碼中, 加入 **Page_PreInit** 的事件處理常式, 然後在常式中使用 **Page.MasterPageFile** 來設定內容頁所要使用的 **Master Page**。一定要在 **PreInit** 的事件處理, 在別的事件處理會發生錯誤。

解決方案

在內容頁的程式中, 加上下列的事件處理常式：

```
private void Page_PreInit(object sender, EventArgs e)
{
    // place master page changing code here.
}
```

然後在這個常式中, 設定內容頁的 **Master Page**：

```
private void Page_PreInit(object sender, EventArgs e)
{
    Page.MasterPageFile = "~/masterPages/MasterPage_New.master";
}
```


另外,若想要依使用者的選擇來動態切換 Master Page 的話,可以先將使用者的選擇寫到 Cookie 中,然後再依前述的方法來切換 Master Page,例如：

```
// 儲存使用者選定的 Master Page 路徑。
private void cmdSetMasterPage_Click (object sender, EventArgs e)
{
    Response.Cookies.Add(
        "MasterPage", "~/masterPage/MyMasterPage.master");
}

// 讀取 Master Page 路徑設定, 並設定 Master Page。
private void Page_PreInit(object sender, EventArgs e)
{
    Page.MasterPageFile = Request.Cookies["MasterPage"];
}
```

考生停看聽

本問題中討論的內容,可用來準備 Exam 70-528: TS: Microsoft .NET Framework 2.0 Web Client Development 的主題：

- 使用 Master Page 來實作一致性的使用者介面
 - 動態變更 Master Page。

Q29

我想要在 Content Page 中調整 Master Page 的內容, 請問我要如何做?

適用範圍:  ASP.NET 2.0  ASP.NET 3.5

問題

我的 Web 應用程式有用到 Master Page, 而且我在 Master Page 中放上了像登入控制項, SiteMap 以及 Menu 控制項等的功能, 不過, Menu 是會視內容頁的不同, 而有所改變的, 請問我要如何在內容頁中存取到 Master Page 的 Menu, 讓我可以動態修改 Menu 的內容?

問題說明

在大多數的 Master Page 使用案例中, 都會在 Master Page 放一些所有使用者都會用到的一些控制項, 這些控制項會在 Master Page 套用時顯示共同的功能, 例如像 Sitemap, 或者是一些顯示資料的控制項 (例如圖片) 等等, 這在做 Web Hosting 或是提供會員方式製作個別的網站時會有比較多的支援。

在內容頁是可以存取到 Master Page 中的控制項的, 在 Page 物件中, 有一個 Master 屬性, 這個屬性就連結到 Master Page, 開發人員可以利用 Page.Master 來存取 Master Page 中的控制項。

例如, 若想要修改 Master Page 中的一個圖片控制項 (ID 為 imageHeader), 則可以用這樣的程式碼來存取:

```
Control ctl = Page.Master.FindControl("imageHeader");

if (ctl != null)
{
    Image imageHeader = ctl as Image;
    imageHeader.ImageUrl = "...";
}
```

解決方案

利用 `Page.Master` 來存取 **Master Page** 中的控制項, 若 **Menu** 的 ID 是 `mainMenu` 的話, 則可以用下列的程式來取得 **Menu** 物件:

```
Menu menu = Page.Master.FindControl("mainMenu");
```

取得物件後, 操作的方法就和存取 **Menu** 物件是一樣的。

考生停看聽

本問題中討論的內容, 可用來準備 Exam 70-528: TS: Microsoft .NET Framework 2.0 Web Client Development 的主題:

- 使用 **Master Page** 來實作一致性的使用者介面
 - 由 **Content Page** 參考 **Master Page** 的成員。

Q30

如何讓 Menu 和 SiteMap 共用同一個 XML 資料？

適用範圍： ☒ ASP.NET 2.0 ☒ ASP.NET 3.5

問題

我在網站中使用了 SiteMap 控制項, 也建立好了 Web.sitemap 資料, 同時測試過 SiteMap 功能正常, 但是現在我想要在 Menu 控制項中直接使用 Web.sitemap 來套用選單的結構, 請問我要如何做？

問題說明

在多數的情況下, Site Map 和 Menu 其實是相同, 因為 Menu 是網站所有功能的選單, 而 Site Map 又是網站中所有功能的路徑對應, 因此這兩者如果整合起來, 在管理上會比較方便, 而且只要更改 Web.sitemap, Menu 也可以跟著連動, 可省下一些時間與工夫。

解決方案

Web.sitemap 是給 SiteMap 產生路徑的資料檔, 是 XML 的格式, 因此可以利用 XmlDocument 來讀取它, 接著, 巡覽每個子節點, 然後利用 Menu 的物件模型來將每個項目新增到 Menu 中即可, 範例程式碼如程式 1 所示。

由於 Web.sitemap 中的子節點的結構都相同, 因此可以利用遞迴函式 (Recursive Function) 的方式來實作, 可以節省許多程式撰寫量。

程式 1：將 Web.sitemap 和 Menu 整合。

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!Page.IsPostBack)
    {
        XmlDocument doc = new XmlDocument();
        doc.Load(Server.MapPath("~/Web.sitemap"));
        this.AppendMenuItem(null, doc.DocumentElement.ChildNodes);
        doc = null;
    }
}

private void AppendMenuItem(MenuItem rootItem, XmlNodeList itemNodes)
{
    MenuItem item = null;

    foreach (XmlNode node in itemNodes)
    {
        item = new MenuItem();
        item.Text = node.Attributes.GetNamedItem("title").Value;

        if (node.Attributes.GetNamedItem("url") != null)
            item.NavigateUrl = node.Attributes.GetNamedItem("url").Value;
        else
            item.NavigateUrl = "javascript://";
    }
}
```

```
        if (node.ChildNodes.Count > 0)
            this.AppendMenuItem(item, node.ChildNodes);

        if (rootItem == null)
            this.myMenu.Items.Add(item);
        else
            rootItem.ChildItems.Add(item);
    }
}
```

相關問題

- Q42：我要如何讀取 XML 資料並存入資料庫？
- Q43：如何動態產生 XML 資料？