

第 2 章 我的瀏覽器

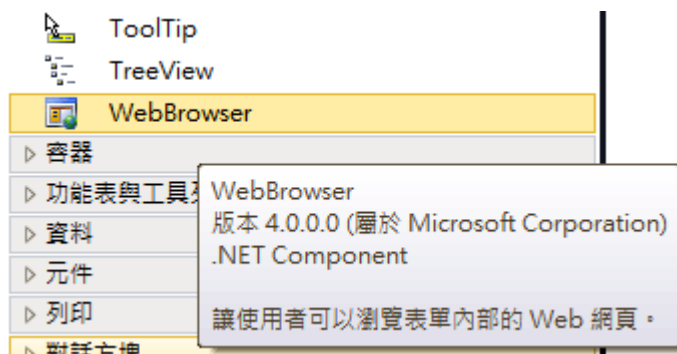
簡介：

網頁瀏覽器是現在多數人使用頻率最高的程式軟體，自製瀏覽器好像很困難？其實在 C#的程式工具箱中已經有相當現成的元件可以使用，並不難做。本章的主要目的是讓同學們知道多數程式其實是「組裝」而成的！找到適合的組件，善用元件的屬性與方法等現成資源加以調整，你就可以用極少的程式碼完成看似相當複雜的程式軟體。

同時，本章也將介紹如何建置與發行程式的基本程序，就是將設計好的程式製作成可以執行與安裝的檔案群組。之後你就可以把自己寫的程式安裝到開始功能表的程式集之中！當然也可以將上課的成果變成可以寄給友人的小禮物，上網供別人下載，甚至出售都行！學完本課相信你一定會覺得與專家們的距離縮小很多了！

2-1 WebBrowser 物件與首頁

[瀏覽器物件在工具箱的位置]



請先開啟新專案，如上圖，到工具箱中的「通用控制項」分類中，找到最後一項的 **WebBrowser**，將它拖放到表單。此物件如其名稱就是一個網頁(Web)瀏覽器(Browser)，它已經具備了瀏覽器的核心功能，外觀預設會直接填滿表單(屬性 Dock = Fill)，看起來就像一個空白的記事本。當然，如果你不希望它佔滿表單，修改 Dock 屬性就可以了！

[設定瀏覽器物件的首頁]

此時執行程式，此物件顯示的「網頁」是空白的！如同一般瀏覽器，必須先替它設定要顯示的網頁。請先點選此物件，至其屬性視窗尋找 Url 屬性(相當於網址)，填入任何一個你熟記的網址，範例中自然是作者的教學網站囉(<http://ycc.tsu.edu.tw>)！在此順便將表單標題(Text 屬性)改為「我的瀏覽器」，也將表單圖示(Icon 屬性)加上，就是去搜尋一個副檔名為「.ico」的圖示檔，在一般電腦內部搜尋一下(*.ico)就有很多圖示可用。此時再執行程式就會看到網頁了！

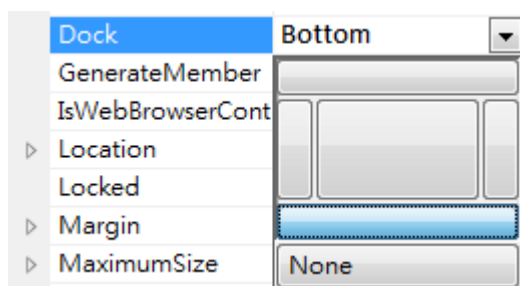


補充說明：Url 的英文全名是：Universal Resource Locator，直接翻譯應該是「通用資源指標」，這是指網際網路上可以定位取得的檔案資源，一般就是指網頁，但是也可以是放在公開網站內的圖檔或其他任何電腦檔案！瀏覽器取得檔案之後如果是它自己可以處理的檔案類型，如網頁、文字檔或圖檔就會直接開啟於瀏覽器內，如果不是瀏覽器可以處理的檔案類型，通常就會問你是不是要下載儲存，或用甚麼軟體開啟了！

2-2 網址列輸入與翻頁

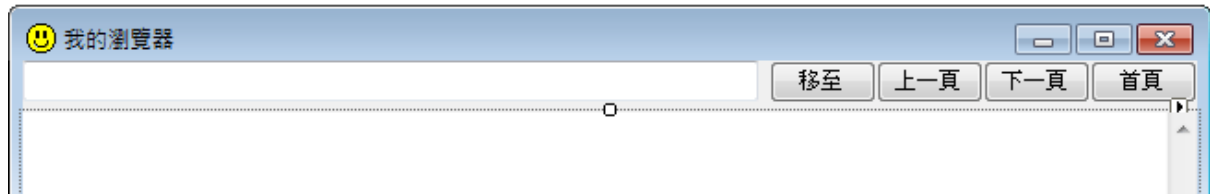
[加入切換網頁的介面]

瀏覽器除了要能顯示網頁之外，也必須要有改變網址與翻頁(上一頁與下一頁)的功能，但是剛剛載入的 webBrowser1 物件好像把視窗的空間都佔滿了！這些操作介面要怎麼放呢？首先要了解 Dock(停駐)這個屬性，請到瀏覽器物件的屬性視窗找到這個屬性，將其值由預設的 Fill 改為 Bottom，如下圖，點選緊鄰最大方塊下方的長條矩形就是 Bottom 了！



上面的圖示意義非常清楚，按中間最大方塊會讓物件填滿頁面就是 Fill 狀態，選擇上(Top)、下(Bottom)、左(Left)或右(Right)方的矩形就會使得物件貼緊視窗相對應的一個

邊緣。選擇最下面的 None 就是物件完全由使用者定義位置，不會自動貼附到任何邊界。Dock 屬性改為 Bottom 之後瀏覽器物件的上緣就不再緊貼表單上邊界，也可以調整，我們就能騰出空間置入需要的物件了。如下圖，我們加入一個較長的文字方塊(textBox1)供輸入網址之用，其後分別是「移至」、「上一頁」、「下一頁」與「首頁」四個 Button 按鍵(button1~button4)。



[加入切換網頁的程式]

「移至」按鍵當然是指移到前面的 textBox1 中輸入的網址，其餘包括上、下一頁與首頁都是一般瀏覽器的基本功能，WebBrowser 物件內部都已經寫好相關的程式，就是該物件的「方法」(Method)！我們只需要呼叫使用它們就可以了！製作這些文字方塊與按鍵只是讓使用者有操作「介面」可以去執行那些功能而已！譬如工廠作好的電視機當然有轉台的功能，但是如果沒有面板上的按鈕或遙控器，那一般人就真的無法轉台了！所以遙控器就是「使用者介面」了！現在分別點擊上面四個按鍵切入程式碼視窗，寫程式如下：

```
//移至
private void button1_Click(object sender, EventArgs e)
{
    webBrowser1.Navigate(textBox1.Text);
}
//上一頁
private void button2_Click(object sender, EventArgs e)
{
    webBrowser1.GoBack();
}
//下一頁
private void button3_Click(object sender, EventArgs e)
{
    webBrowser1.GoForward();
}
//首頁
private void button4_Click(object sender, EventArgs e)
{
    webBrowser1.GoHome();
}
```

上述程式非常簡單，「移至」(button1)按鍵呼叫 webBrowser1 物件的 Naviagate 功能(方法)，其參數(小括號內)加入網址(textBox1.Text)就可以了！「上一頁」(button2)是 GoBack，「下一頁」(button3)是 GoForward，首頁(button4)是 GoHome，如同上英語初級班，非常容易的就可以完成四個功能！

學到這裡你應該已經有點習慣使用「X.Y」這種語法了！通常前面的 X 就是一個「物件的名稱」，「.」的後方則是這個物件的「屬性」、「方法」或者「事件」。譬如 `textBox1.Text` 是物件的一個屬性，`webBrowser1.GoHome()` 是物件的方法(或者說它的功能)，`button1.Click` 則是物件所觸發的事件。簡單說，事件、方法與屬性都與物件有「從屬」關係。如果是方法，就是必須作出一些動作，通常還需要夾帶一些參數，所以必須附加一組小括號，裡面準備置入參數，即使不需要參數，空括號還是必須的，否則會出現錯誤訊息「只有指派、呼叫、遞增、遞減和新增物件運算式可以當做陳述式使用」！怎麼樣？很難懂吧？沒辦法 C 語言講話就是這個調調，很像火星文！

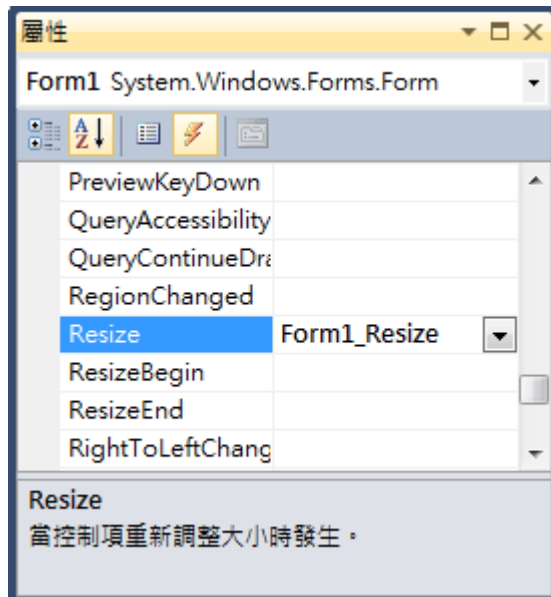
2-3 用程式動態調整版面

上面的程式已經可以上網也可以切換網址與翻頁，但是如果改變一下它的視窗大小，譬如放到最大，就會發現 `Dock=Bottom` 的屬性設定雖然可以讓瀏覽器貼在下邊緣，但是縮放時物件本身的高度不會變！就是會在整個視窗放大時，網址列與瀏覽器中間變出了多餘的空白；縮小時則讓網頁的上面有一部分被擠出畫面。下圖就是縮小視窗時出現的不合理版面：



[自動調整瀏覽器大小的程式]

我們希望的狀況是不論視窗大小如何變動，瀏覽器與網址列等介面應該都是維持貼齊的狀態。這時應該是在視窗(Form1)「改變大小」時寫入調整位置的程式。如何開始呢？請到設計工具檢視視窗，點選表單標題(選擇 Form1 物件的意思)，再到屬性視窗點選閃電標記，出現下拉式選單後選擇「Resize」事件，如下圖：



這樣就會出現程式框架 `Form1_Resize`，亦即表單改變大小時，請寫入程式如下：

```
private void Form1_Resize(object sender, EventArgs e)
{
    webBrowser1.Height = this.ClientSize.Height - textBox1.Height;
}
```

直譯程式碼的意思是：將表單(`this`)可編輯區域(`ClientSize`)的高度(`Height`)減去文字方塊 `textBox1`(網址列)的高度，作為瀏覽器(`webBrowser1`)的高度。這類動態調整畫面的技巧在軟體製作之中常常用到，因為螢幕解析度的可能性變化很多，如果想讓程式在不同螢幕上看起來都很正常，就必須熟悉這些技巧。

[`this` 就是本表單]

這裡也出現了一個有趣的關鍵字「`this`」！我們都知道這是英文中「這裡」的意思，在 `C#`裡面這是「本表單」的代名詞。但是為何不用「`Form1`」呢？本表單的名稱不是 `Form1` 嗎？如果你將此處的 `this` 改成 `Form1` 反而會出現錯誤！原因是在物件導向程式概念中 `Form1` 被視為一個「類別」(英文為 `Class`)，好像工具箱中的顯示的物件，看起來是一些物件，其實它們都只是程式「模型」，而不是實體(`Instance`)。只有在我們將工具箱物件拖到表單時才會產生一個「作好」的，「實際存在」的，可以寫程式去操作的物件實體(原文為 `instance`)。

不僅工具箱物件，連 `Form1` 表單本身也是在執行程式時才產生的一個實體(就是 "`this`"囉)，理論上也可以繼續由程式碼產生多個 `Form1` 的實體(我們不是可以同時開啟好幾個 `IE` 視窗嗎?)，此時 `this` 就是指這個實體(視窗)的本身，程式裡如果寫 `Form1` 則好像是指所有同型的 `Form1` 物件，這是程式設計規則所不允許的！

2-4 看看 HTML 原始碼

[讀寫檔案的 File 物件]

如果你學過一些網頁設計就會知道網頁是由 HTML 語言組成的檔案，常見的 IE 瀏覽器功能表裡面就有個檢視→原始碼的功能，可以看到目前顯示網頁的『程式碼』。在此我們也可以增加一個按鍵建立一下這個功能，請在前面四個按鍵之後加入一個 HTML 按鍵，並寫入程式碼：

```
private void button5_Click(object sender, EventArgs e)
{
    System.IO.File.WriteAllText(@"D:\html.txt", webBrowser1.DocumentText);
    webBrowser1.Navigate(@"D:\html.txt");
}
```

上述程式中 `System.IO.File.WriteAllText` 是呼叫 .NET 函式庫中負責檔案輸出入 (Input/Output) 命名空間的 File 物件來寫出文字 (HTML 原始碼) 的動作，接著在第一個參數中寫 `@"D:\html.txt"`，這是要寫入的檔案名稱。

[逸出字元與路徑名稱的衝突處理]

需要在檔名前面加上一個 @ 的原因是：在 C 語言中 `"\"` 這個字碼稱為逸出 (Escape) 字元，不加上 @ 程式會認為它有特殊的意義，譬如 `"\n"` 代表換行等等，加上 @ 就是告訴 C# 不要想太多，此處的 `"\"` 只是磁碟目錄名稱的前置符號而已！事實上，還有另一個方法可以解決這個問題，讓 C# 知道這裡的 `"\"` 不是逸出字元，就是直接重複倒斜線字元，將檔案路徑寫成：`D:\\html.txt` 就可以了！

`File.WriteAllText` 的第二個參數是要寫到檔案的文字內容 (字串)，就是目前瀏覽器物件 (`webBrowser1`) 裡面網頁的文字 (`DocumentText`)，也就是 HTML 原始碼了！`webBrowser1.Navigate(@"D:\html.txt");` 則是讓瀏覽器重新載入這個文字檔，請注意！雖然其實文字內容與之前的 HTML 檔案是一樣的，但是副檔名已經改成 `"txt"`，所以瀏覽器只會將它當作一般文字檔顯示，瀏覽器其實是看副檔名來決定怎麼處理檔案的！

[文字編碼的差異]

下面是程式執行的畫面，先是正常的網頁，後是原始碼，由於目前 .NET 函式庫對於中文 BIG5 的編碼支援還不完整，所以硬要將此功能用在 BIG5 網頁中文會顯示得不正常，下面例子是 Yahoo 網頁，使用的是 UTF8 編碼就沒問題。順便也介紹大家知道：UTF8 編碼是所謂的 Unicode，就是各國文字都可以統一編碼的方式，如果沒有特別原因，目前網頁都會自動採用，也因此目前網頁因為編碼錯誤出現亂碼的情況越來越少了。



2-5 建置與發行

[建置程式執行檔的位置]

所謂「建置」就是將你寫的程式「編譯」(Compile)成為電腦可以執行的檔案，詳細過程相當複雜，在此僅簡要的說明，你需要知道的是建置完成之後會產生一個副檔名為 exe 的執行檔。C#軟體在每次你按偵錯執行程式時都會先進行編譯，也就是「建置」

的工作，製作出來的執行檔會放在專案目錄的 **bin\Debug** 目錄之中！如果你的專案是預設的 **WindowsApplication1** 那麼執行檔就會是 **WindowsApplication1.exe**！試試看在目前設計的專案中找出這個檔案，直接執行(雙擊圖示)就會與你偵錯時看到的畫面一模一樣！即使你將 **Visual Studio** 關閉，這個檔案也一樣可以直接執行哦！

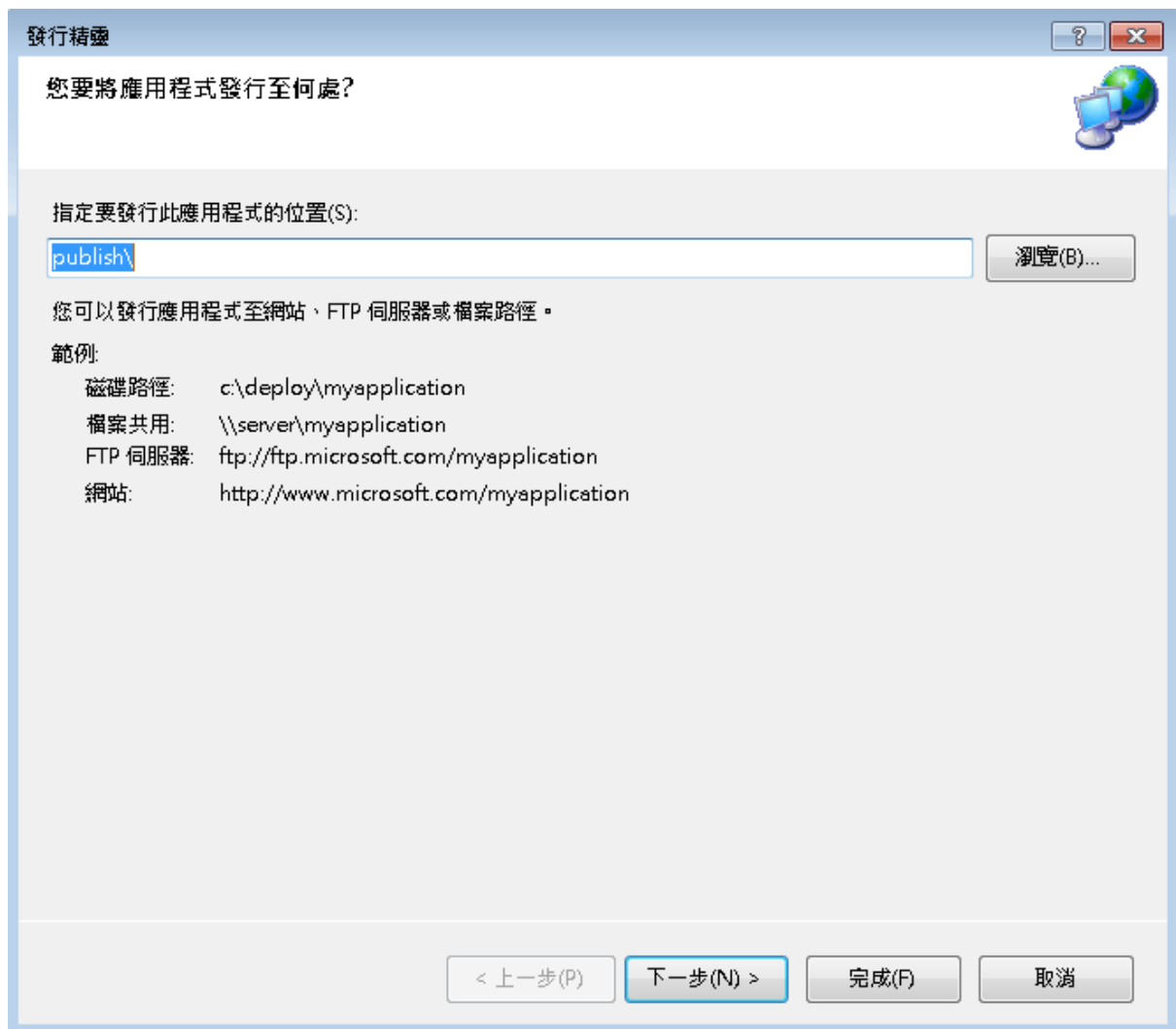
[執行檔也要依賴系統環境]

但是如果將這個 **exe** 檔案移到別部電腦也可以直接執行嗎？這就必須看情況了，此處的 **exe** 檔案其實還是不能完全獨立運作的！如果另一電腦環境有安裝 **Visual Studio** 軟體，或者 **C# Express**，甚至 **Visual Basic Express** 就一定可以執行(事實上應該說是對應版本的 **.NET Framework Library**)，否則就要看該電腦之前安裝過的軟體中有沒有這個 **exe** 檔需要的所有資源而定。通常安裝過與此程式使用同一版本的程式語言(**Visual Studio**)開發的程式應該就沒問題(已安裝對應版本的 **.NET Framework Library**)。

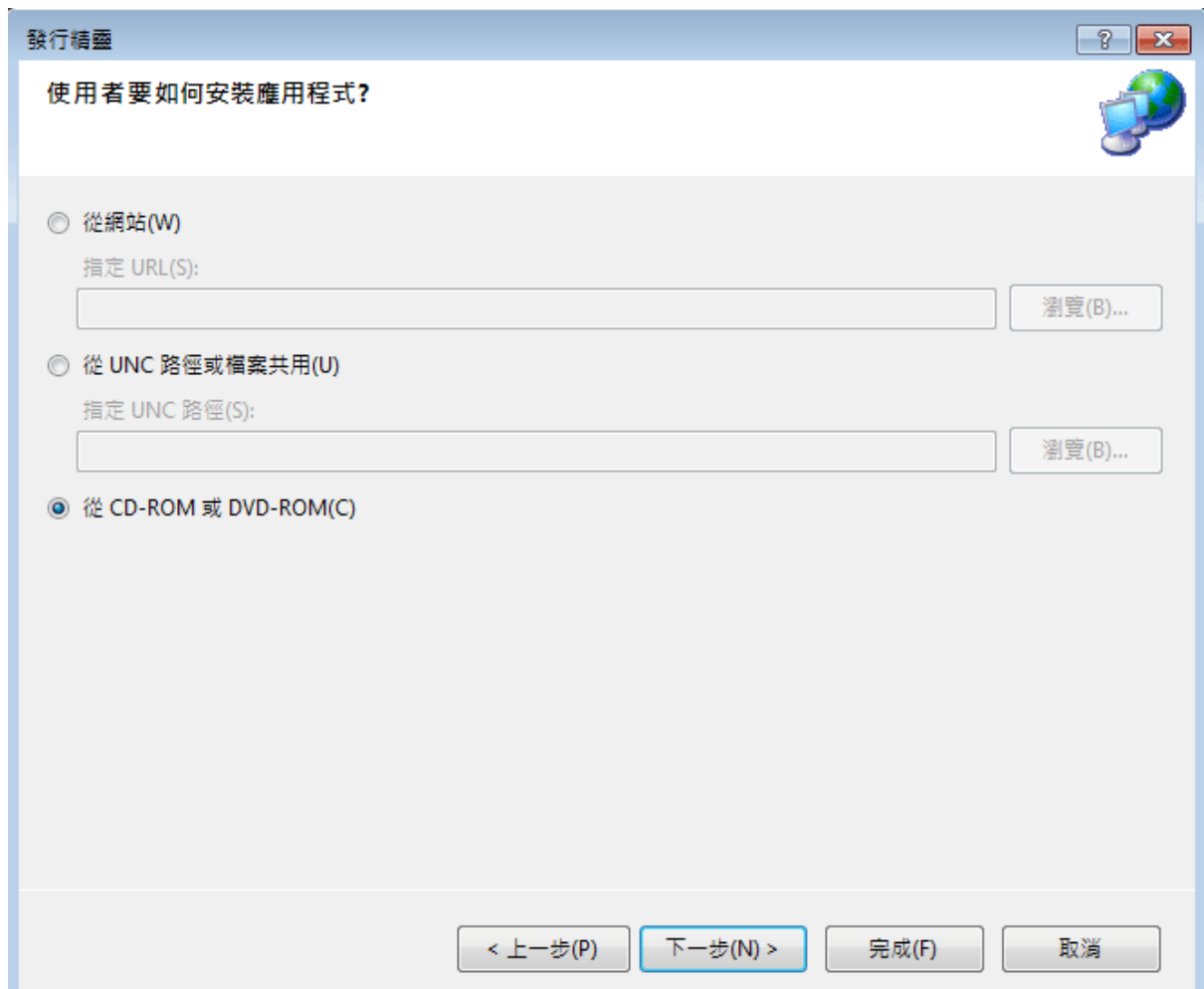
[發行程式的程序]



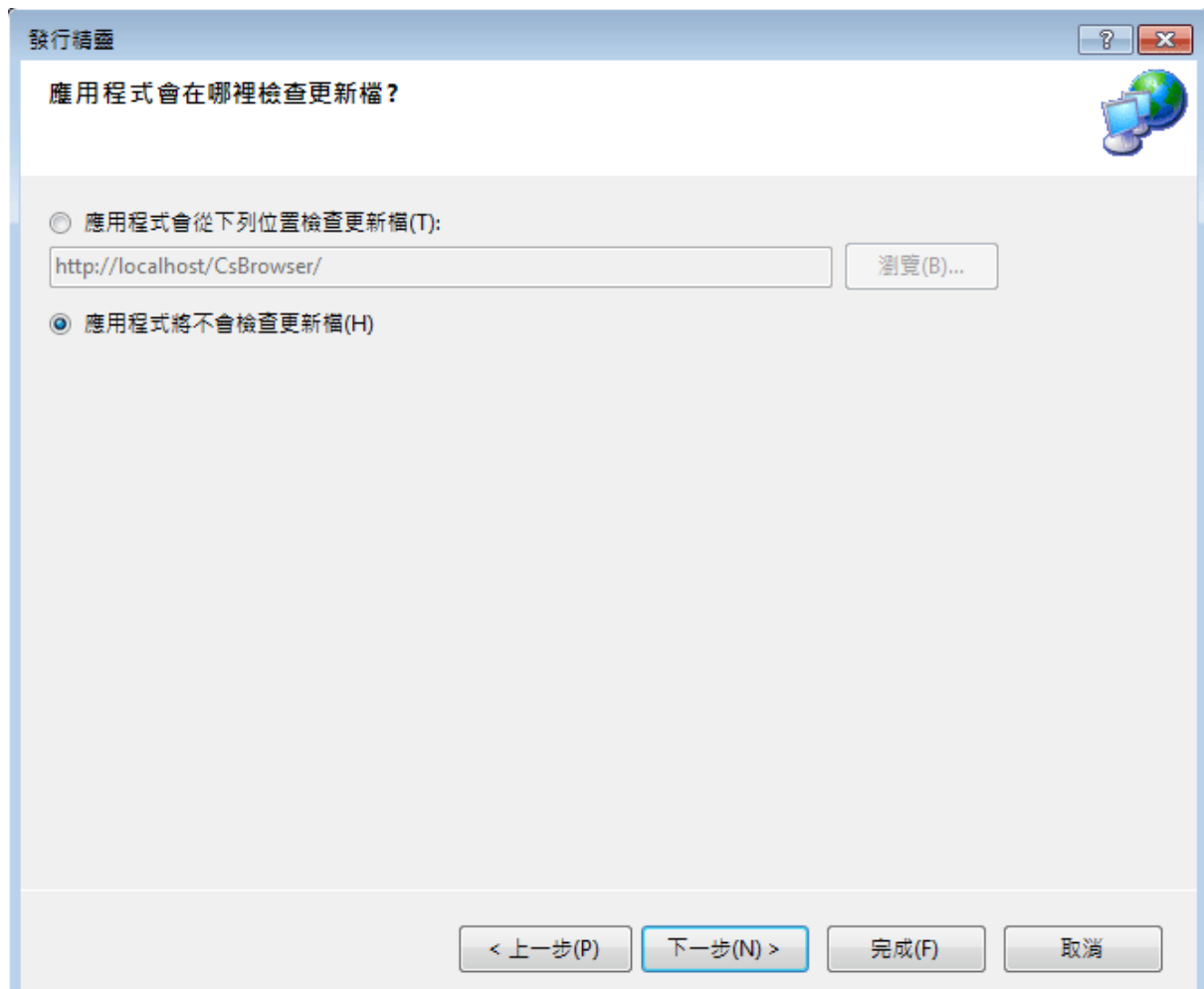
要建立完整的安裝檔在此稱為「發行」，請先到主功能表的「專案」選項拉開畫面如上圖。選擇上面的「發行 **CsBrowser2010**」選項(本專案範例被命名為 **CsBrowse2010**)，就會進入一個程式精靈，引導我們逐步的完成安裝檔的製作，第一個畫面如下：



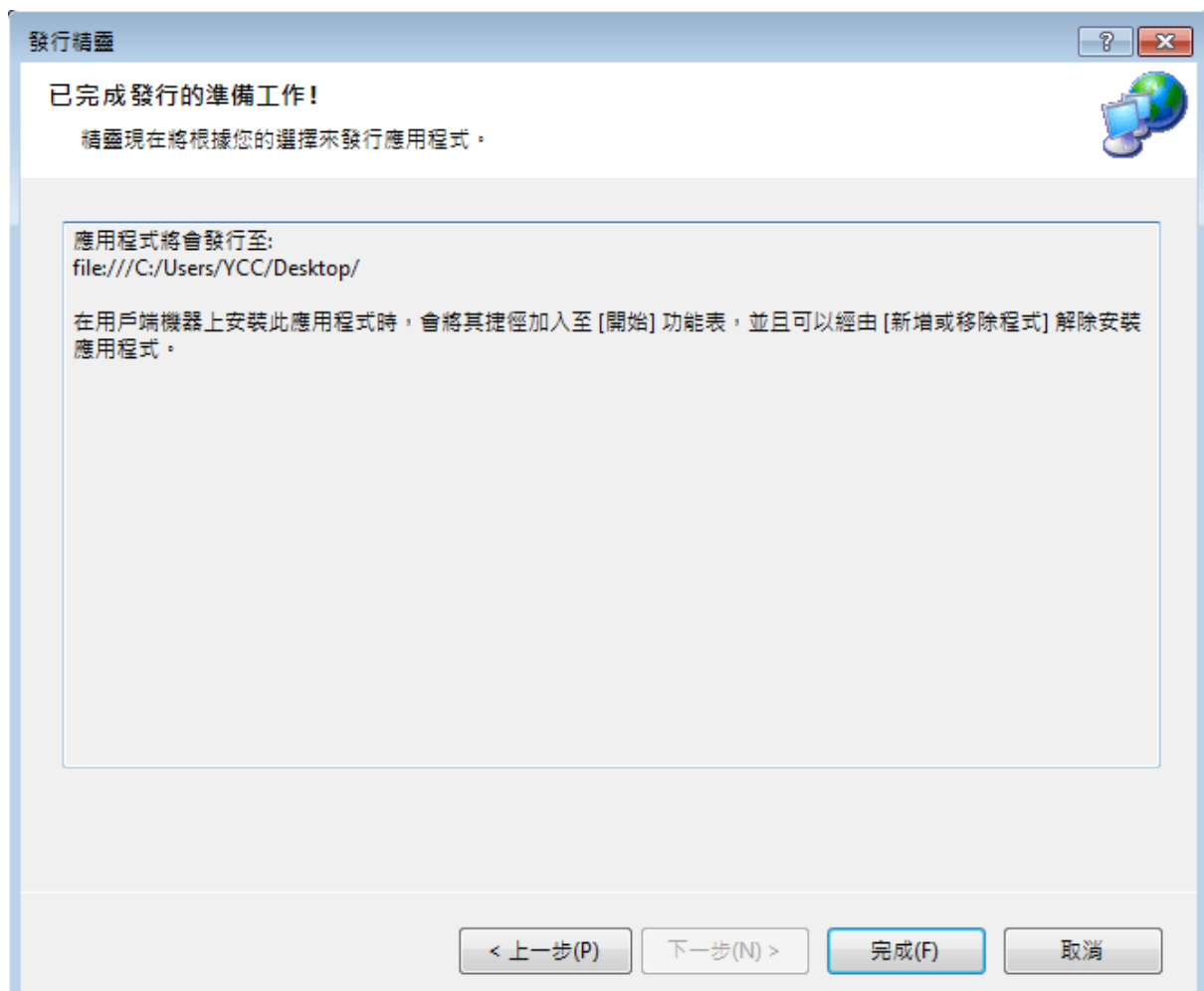
預設的指定發行位置顯示是「publish\」意思是製作出來的安裝檔案群會輸出到專案目錄內一個顯示名為「publish」的目錄，你也可以自己修改目錄到自選的目錄。接著按下一步。



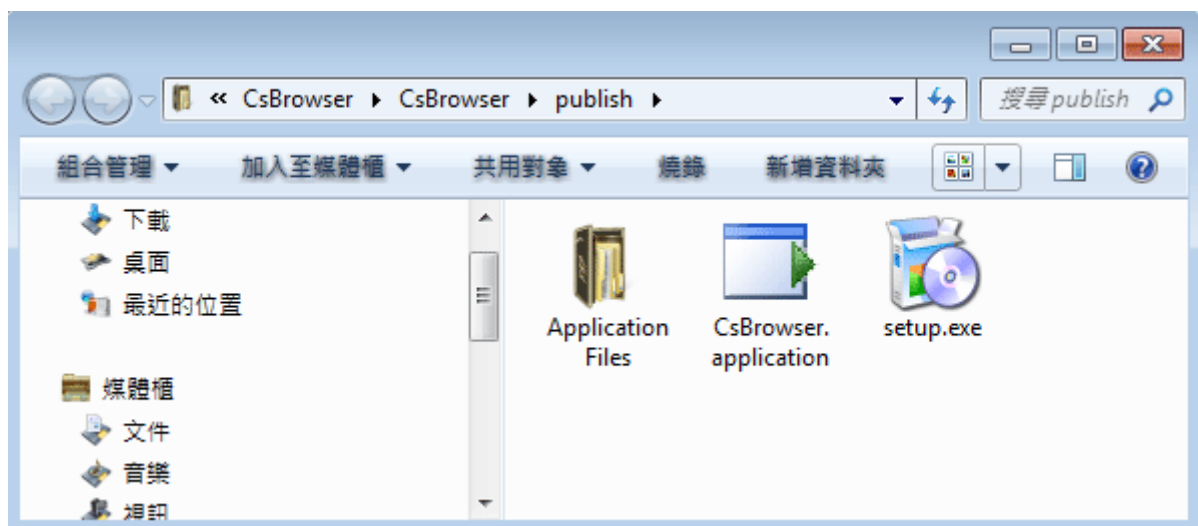
此步驟使用預設選項即可，比較迷惑人的是這個選項好像暗示安裝程式一定要寫到 CD 或者 DVD 上面？沒這回事！用任何硬碟、隨身碟甚至記憶卡都可以的。下一步畫面如下：



在此請使用預設值「不檢查更新」。因為如果要讓程式可以自動更新做法是必須將安裝檔案放在某個網站，程式每次啟動時會自動上網檢查版本，如果網站版本較新就會問使用者是否要安裝新版？相信常用電腦的人都有此經驗，我們在此還不必花功夫去設網站。



出現以上畫面按下「完成」，稍候數秒鐘就會跳出一個檔案總管視窗，概略如下圖：

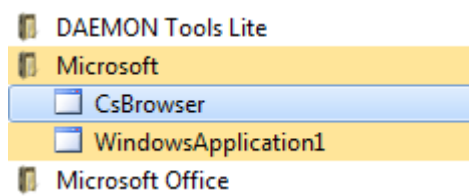


目錄內的檔案與目錄就是完整的安裝檔了！你可以先在自己的電腦執行 **setup.exe** 程式，請記住它不是獨立的！必須與旁邊的檔案與目錄放在一起執行。結果應該是你的電腦程式集之內會出現新的程式，與一般軟體安裝完全相同！需要移除時也與一般程式一樣，可以到「控制台」選擇「新增移除程式」的選項執行移除。

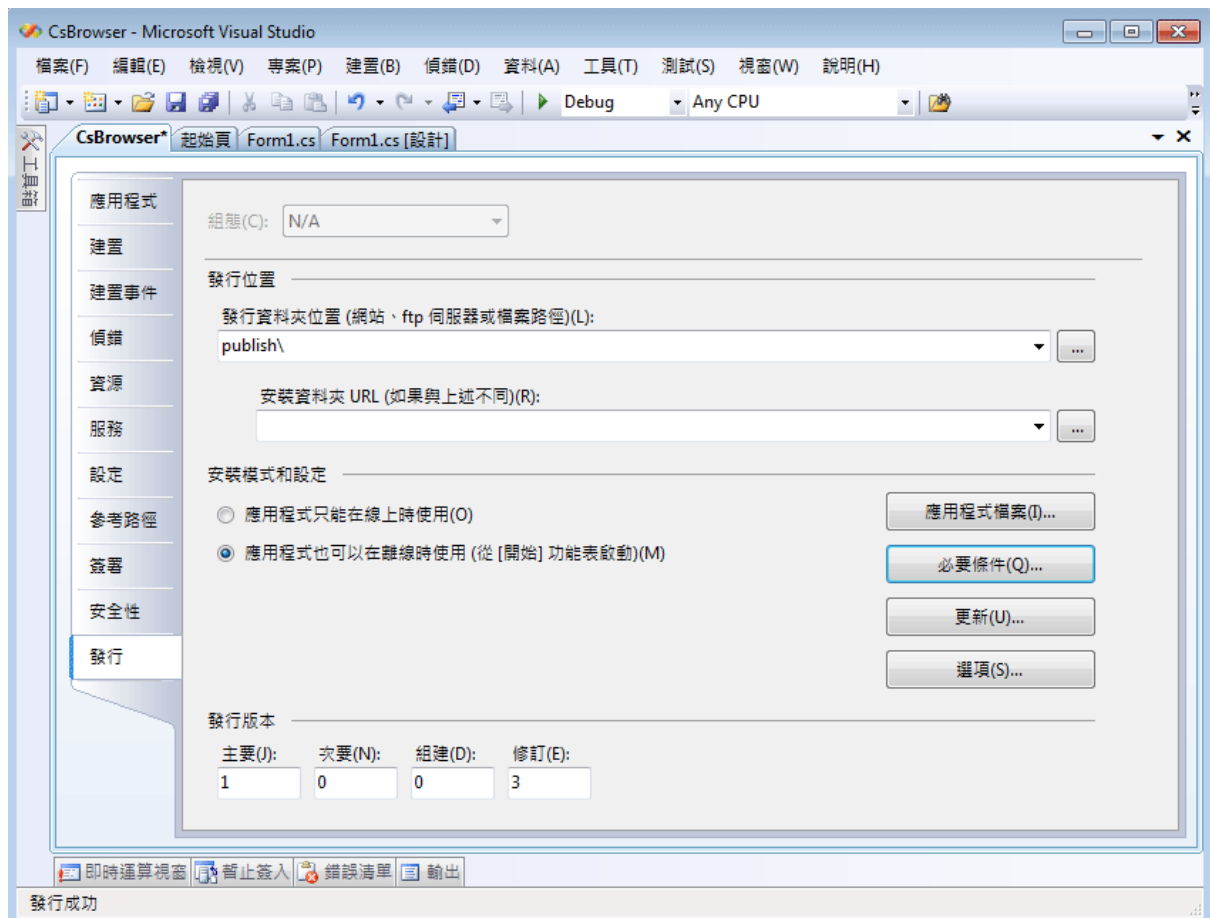
2.6、建立你的品牌

[修改你的程式在程式集中出現的外觀]

將上述的安裝檔案執行安裝之後程式集顯示畫面可能如下(此為作者的 Windows 7 系統畫面)：



你的「公司」名稱居然是 Microsoft？程式可能是預設的「WindowsApplication1」，圖示檔也是預設值，一看就知道是初學者的作品，真的遜斃了！如此這般，這一章的學習就完全破功了！要修改很容易的，首先先到方案總管視窗開啟 Properties 選項，切到發行頁籤，畫面如下：



裡面有個「選項」按鈕，按下去會出現下面的視窗，修改一下「發行者」與「產品名稱」，通常「發行之後開啟部署網頁」也是不需要的可以取消，如下圖：

發行選項

發行語言(P):
(預設)

發行者名稱(L):
台灣首府大學

產品名稱(N):
我的瀏覽器

支援 URL(S):
瀏覽(B)...

部署網頁(W):

每次發行之後自動產生部署網頁(G)

發行之後開啟部署網頁(O)

封鎖應用程式，使其無法透過 URL 啟動(K)

使用 ".deploy" 副檔名(D)

允許傳遞 URL 參數至應用程式(U)

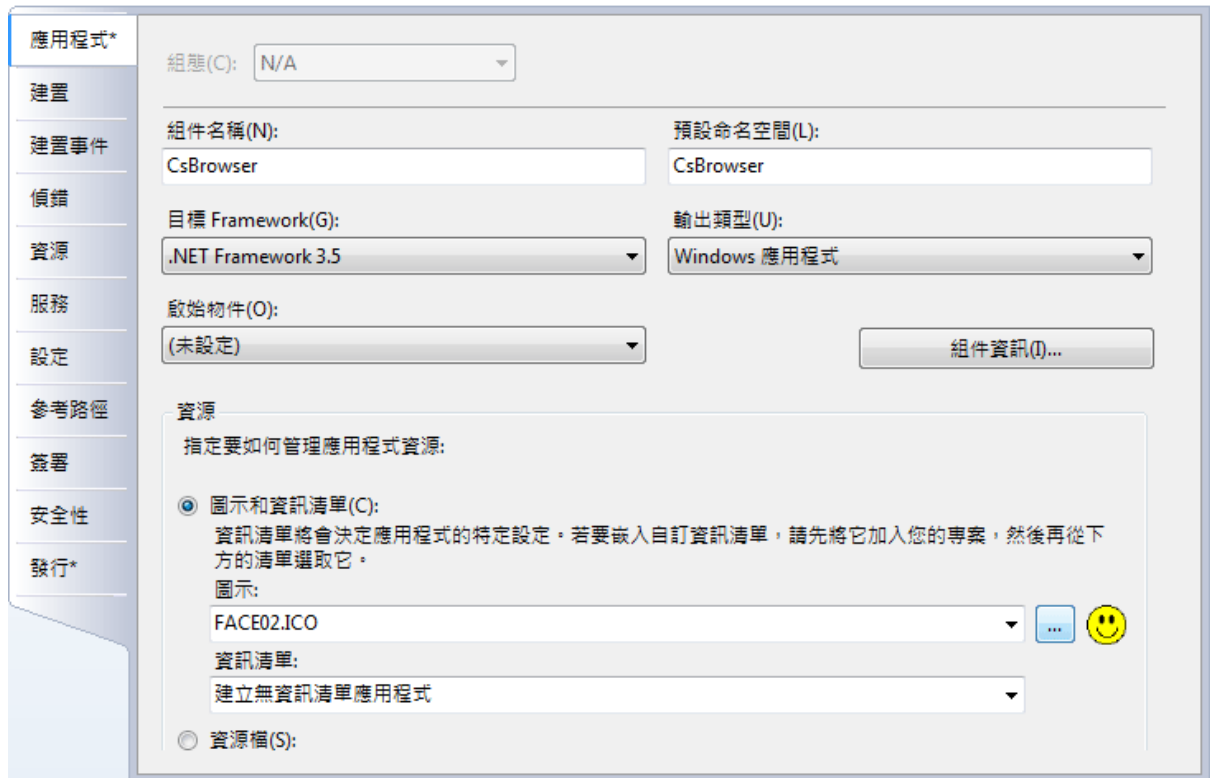
若是使用光碟安裝，在插入光碟後自動啟動安裝程式(C)

驗證上載至 Web 伺服器的檔案(V)

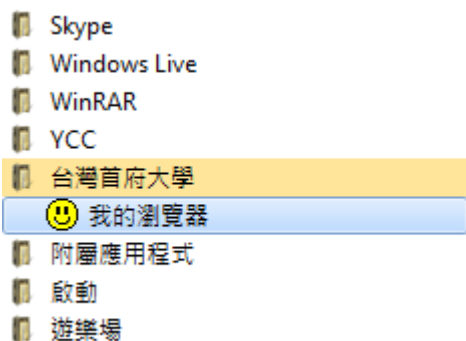
使用應用程式資訊清單供信任資訊使用(A)

確定 取消

接下來回到專案屬性視窗選擇「應用程式」頁籤，在「圖示」那個地方去選一個副檔名為 `ico` 的圖示檔即可。如果沒有現成的圖示檔，可以在電腦內蒐尋`(*ico)`找出一個，通常隨便都可以找到幾百個，或上網去抓一個，只要不將程式出售之前都不必擔心有侵犯版權的問題。畫面如下：



此時再執行發行與安裝動作，你的程式集出現的就是有品牌與圖示的專業樣貌了！



2-7、進階挑戰

一、如何讓前一個單元的程式也建立漂亮的安裝檔，與此單元隸屬於同一品牌？

提示：讓「發行者」名稱一樣即可。

二、如何讓瀏覽器程式的網址列隨著視窗縮放也可以伸長與縮短？

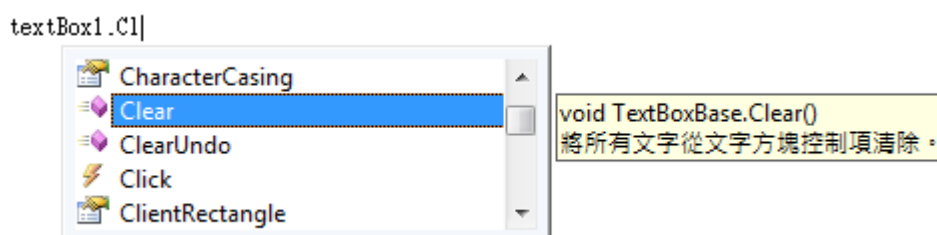
提示：修改 TextBox 的 Width 屬性，同步必須修改 Button 物件的位置(Left 屬性)。

課後閱讀

智慧感知會用了嗎？

[無所不在的貼心提示]

相信同學在寫程式碼過程中已經習慣看到自動提示的選項了，很棒！不是嗎？它有個漂亮的名稱叫作 **IntelliSense**，中文翻譯叫作智慧感知。很多同學都會說英文不好所以學程式有困難，其實寫程式用的英文字彙不多，大概幾百個單字就很夠用了！有了智慧感知功能更是只需要記住每個字的前兩三個字母就夠了，這讓英文能力差距對學程式的影響降到最低的程度。



[英文好不好都應該用]

我常常對於堅持自己打字，卻又常常打錯的同學開玩笑說：『你的英文這麼好哦？都自己打字哦？老師寫程式都是靠提示的欸！』這其實無關英文程度如何，即使你英文很好也應該盡量使用它，這是工作效率的問題！尤其是 **C#** 中字母大小寫是有區別的，也不像 **VB** 語言大小寫打錯了會自動更正，打字費時就算了，錯了還得手動修改，更加耗時，哪有不盡量用智慧感知的道理？而且有些使用細節也必須介紹一下，可以幫你用得更快更順暢！

[常見的小問題]

初學者一開始看到智慧感知出現就會用滑鼠點取，但是因為你正在打字，手離開鍵盤去抓滑鼠是會浪費時間的，其實使用上下鍵移到選項比較快，移到之後只要按下 **Enter**(輸入)鍵就可以讓選項直接寫進程式碼。新手常常會發現智慧感知不見了！怎麼老師打的時候都有提示，我打就常常沒有？

實際情況是智慧感知不是只看字典，也會隨時根據前面程式的需求篩選選項，如果智慧感知已經發現你有明顯的錯誤，就會停止提示。因此，如有無提示的情況，第一個反應就是看看前面的程式碼是不是錯了？不論有無錯誤，其實都可以強迫它繼續提供服務的，就是按下 **Alt+向右鍵(→)**，提示就一定會再度出現的！

網路瀏覽器物件存在的意義

[收看網際網路訊息的電視機]

無庸置疑，網路早已是現代人生活的一部分，打開電腦第一件事通常就是開瀏覽器看看外面的世界，所以現在的電腦與其稱為 **Computer** 還不如稱為 **Communicator**(通訊器) 更為貼切。所謂的網際網路(**Internet**)世界基本上就是使用 **http** 通訊協定為溝通模式的一個平台，**http** 的全名是「**HyperText**(超文字) **Transfer**(傳輸) **Protocol**(協定)」。本來它只是一種文件的格式加上一些通訊的規則，隨後因為使用方便與逐步出現的各種需求，網路世界的技術變得越來越豐富而多元，我們可以經過這個平台做的事情也越來越多。但是好像我們看電視一樣，即使外面有上百個衛星頻道，總需要一台電視機來收視吧？瀏覽器就是這個關鍵的通訊器材了！

[我的程式也可以上網]

我們已經知道，寫程式時可以在工具箱找到許多需要的零件，但是可曾想過我的程式如果需要找高鐵時刻表或要查郵遞區號怎麼辦？可不可以在程式內部「呼叫網路」，取得這些資源呢？本單元用的 **WebBrowser** 物件就是為此而出現在工具箱的！表面上本單元範例只是在程式內開個網頁而已，但實際上代表的重要意義是我的程式確實可以和網際網路直接交換訊息！當你使用 **Navigate** 指令讓 **WebBrowser** 傳出一個網址時，其實也可以附帶查詢字串(提出問題)，網站回覆你的也可以是符合 **http** 通訊協定的任何東西(問題解答)，不只是美美的網頁而已。上面說的查時刻表與郵遞區號等功能，使用 **WebBrowser** 物件加以設計確實都是可以辦到的！

有關程式的編譯

[編譯就是從人的文字語言變成電腦的數位語言]

電腦程式從我們可以看懂的簡單英文到變成可以在電腦執行的軟體，事實上會經過很多複雜的程序，最後翻譯成電腦可以執行的機器碼檔案，也就是副檔名為 **exe** 的執行檔。這個過程我們通常稱之為「編譯」(**Compile**)，如何編譯其實與電腦的硬體環境及作業系統都是相關的！所以不是說編成了執行檔案就保證可以在任何電腦執行。這是一個非常麻煩的問題，所以多數程式語言都很努力讓這種障礙盡量減少，如果作得不錯就自稱為「跨平台」程式語言！但實際上這還是一個理想，因為電腦的硬體乃至作業系統也一直在演進，真正跨平台的程式語言可能永遠都不會真的出現。

[編譯和作業系統息息相關]

作為一個應用程式的設計者，「編譯」這部分就讓研究編譯程式的專家們去傷腦筋吧！我們必須接受的現實是我們寫的程式未必可以在所有的地方執行，寫程式還是必須事先考慮最終執行的環境，譬如要作一個 **Android** 系統的手機程式目前還是不能用我們的 **C#** 直接設計的！甚至編譯好的安裝檔案，如果到了太新(或太舊)版的 **Windows** 作業系統無法安裝或正常運作，那也是常有的事。這不是我們的錯，而是電腦本身一直在進化。適度的修改程式、更新編譯程式並仔細選擇編譯選項還是必要的功課。