

# VB.NET 之物件陣列

如果有人同時用過 VB6 跟 VB2008（或更新版的 VB），必然會發現其中有許多的差別，在此不一一列舉，不過我想當有人要寫一個踩地雷的程式時，沒有了物件陣列或多或少會帶來一些的不方便。因此筆者在研究了一段時間之後，找出了 VB.NET 中物件陣列的使用方法：它必須要直接使用 Dim 宣告！以下皆以 VB2008 做為示範。

```
Dim btn(9,9) As Button
```

如此的直觀。

不過在程式碼中打上了這一句之後，點下開始偵錯，卻會發現什麼都看不到，原因為何？因為現在我們只是把它們「宣告」出來，但我們還要把它們「建立」成一個真正的物件，並且「擺上」表單。

```
Private Sub Form1_Load(...) Handles Me.Load
    For i = 0 To 9
        For j = 0 To 9
            btn(i, j) = New Button
        Next
    Next
End Sub
```

選擇能在一打開程式就會發生的 Form1\_Load，利用巢狀迴圈一個一個跑。

有人可能會問，為什麼不在一開始宣告物件陣列時就把「New」加進去呢？因為若如此做，VB2008 便會跑出一條可愛的藍色蚯蚓，告訴你「不可以使用 New 宣告陣列」。

接下來再把按鈕「擺上」表單吧，一樣要擺在巢狀迴圈裡面。

```
Me.Controls.Add(btn(i, j))
```

記得把這行放在「btn(i, j) = New Button」下面，不然因為程式碼是由上往下跑的，當程式要「擺上」按鈕的時候會發現這個按鈕並沒有被「建立」出來，結果就等於——沒有「擺上」。

這時若有人禁不起誘惑點了開始偵錯，便會發現一個殘酷的事實：所有的按鈕都疊在一起，都長的一模一樣，好像只有一個按鈕似的，而且啊，還呈現一種很噁心的長條形狀！聰明如我們，腦中應該立刻浮現一件事：我們要去調整物件屬性，在檢視工具設計那邊找了半天，剛剛宣告的按鈕一個都找不到，要調屬性？別鬧了！既然我們是用程式碼把它們宣告出來的，調整屬性也要從程式碼去調整！不知道諸位是否還記得在 VB 社課中講師一直強調的「物件名稱.物件屬性=屬性值」？

```
btn(i, j).Size = New Size(30, 30)
```

記得要打在「`btn(i, j) = New Button`」這行後面，因為我們要把物件「建立」起來它們才能擁有屬性，其他屬性的調整以此類推.....我相信等到大家打完全部的屬性，手都打到斷掉了。有個陳述式叫「**With**」，是專門用來調整一大串物件屬性的：

```
With btn(i, j)
    .Left = i * 30 + 20
    .Top = j * 30 + 20
    .Size = New Size(30, 30)
End With
```

使用方法不言而喻。

再來進行偵錯，就會發現你的按鈕一個一個立正站好排在那兒等你了。

談完了物件跟屬性，事件呢？我們得幫這些按鈕新增一些事件讓它們有些功用，可是不免又會碰到一些麻煩：如何去新增？上面的類別名稱根本找不到我們剛宣告的「`btn`」啊！程式碼萬能，沒有程式碼萬萬不能，萬能的程式碼又派上用場了！

首先，再宣告一個按鈕：

```
WithEvents btnEvent As Button
```

使用「**WithEvents**」宣告，這樣宣告出來的按鈕就會直接被配置有原本按鈕該有的事件，跟「**New**」一樣，它是不能拿來宣告陣列的。如此做之後，我們就可以在類別名稱那裡找到「`btnEvent`」這個按鈕，也就可以建立一個 `btnEvent.MouseClick` 事件了。

接下來我們就只差一個工作，這是一個筆者認為非常厲害的步驟，就是把這個沒有事件的按鈕陣列在按下按鈕時「導向」`btnEvent.MouseClick` 這個事件

```
AddHandler btn(i, j).MouseClick, AddressOf btnEvent_MouseClick
```

大致上我們可以將這行解釋為「**AddHandler** 被引導之事件, **AddressOf** 目的地」。

這樣，當我們一按下物件陣列 `btn` 中的任一按鈕，就能引發 `btnEvent.MouseClick` 這個事件中的程式碼了。

剩下就回到大家熟悉的部分了，如何把這東西變成一個踩地雷程式就靠你們自己囉，在此祝大家和 VB.NET 相處愉快吧！

David Ye, 2011.6.6

原文刊載於師大附中電算社 32、33 屆社刊