

## 第幾章 利用 DataGrid 進行資料「編輯」

網 站：[http://www.taconet.com.tw/mis2000\\_aspnet](http://www.taconet.com.tw/mis2000_aspnet)

站長信箱：[mis2000lab@yahoo.com.tw](mailto:mis2000lab@yahoo.com.tw)

DataGrid 除了用來呈現資料與進行排序（Sort）之外，還提供了資料編輯（Edit）的功能。只要幾個簡單的動作，就可以立即編輯與修改資料。不管是操作的簡便或是資料庫的控管，都是 ASP.NET 的新功能與新突破。

以 ASP 為例，以前要在 Web 畫面上進行資料編輯，至少要撰寫兩隻程式。而且對於初學者來說，要瞭解的基本觀念不少，並不是簡單的事情。而利用 Visual Studio .NET 來撰寫的 ASP.NET 卻能在短時間內，完成這樣的動作，其他的 Web 程式語言（ASP、PHP 或 JSP）都只能自嘆不如。

### X-1 DataGrid 的前置作業

首先，我們先配置好資料庫的連結（SqlConnection1）、設定 SQL 指令（SqlCommand1。在此先不撰寫 SQL 指令，等到需要的時候，再由後置程式碼來撰寫之）以及拖曳一個 DataGrid 到畫面上（詳見下圖）。

我們利用前面撰寫過的「排序」程式，以此做為藍本，進一步的修改與增添新功能。

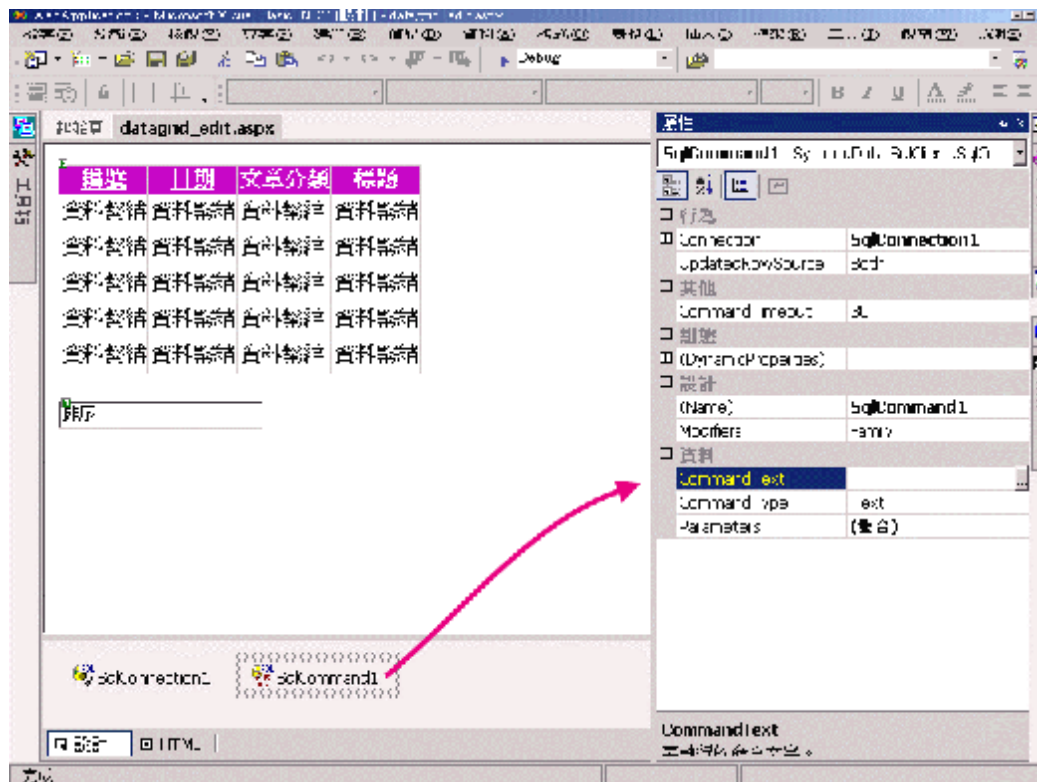


圖 1 前置作業（務必保留 CommandText 的空白。先不撰寫 SQL 指令）

我們先將 DataGrid 的標題欄位，重新自訂標題（AutoGenerateColumns = **False**）。順便加入「編輯」的功能。

只要利用 DataGrid 的「屬性產生器」，便能完成上述的動作。要打開屬性產生器，有下列兩種方法：

- 滑鼠指標對準 DataGrid，然後按下「滑鼠右鍵」，會出現一個繃現視窗。視窗的下方就有「屬性產生器」。
- 或是從畫面右邊，找到 DataGrid 的「屬性」視窗，最下方也有「屬性產生器」。



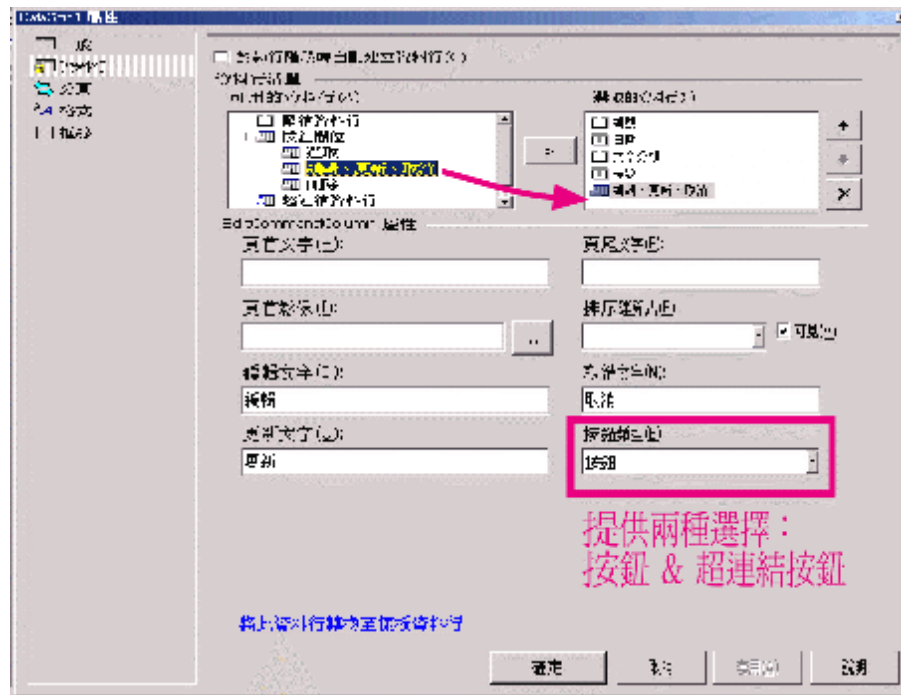


圖 3 DataGrid 的屬性產生器，從「按鈕欄位」裡面製作「編輯」按鈕

「按鈕欄位」有兩種呈現方式，分別是「**按鈕**」與「**超連結按鈕**」。這兩種型態都是「按鈕」型態，只是在畫面的呈現方式不同而已。

「**超連結按鈕**」不是「網頁的超連結」！千萬留意！……「超連結按鈕」是一種按鈕型態，只是在畫面上看起來像是網頁的超連結而已。初學者很容易搞錯，請特別留意。如果要使用傳統的超連結（Hyperlink），請選用圖片左上方的「超連結資料行」。

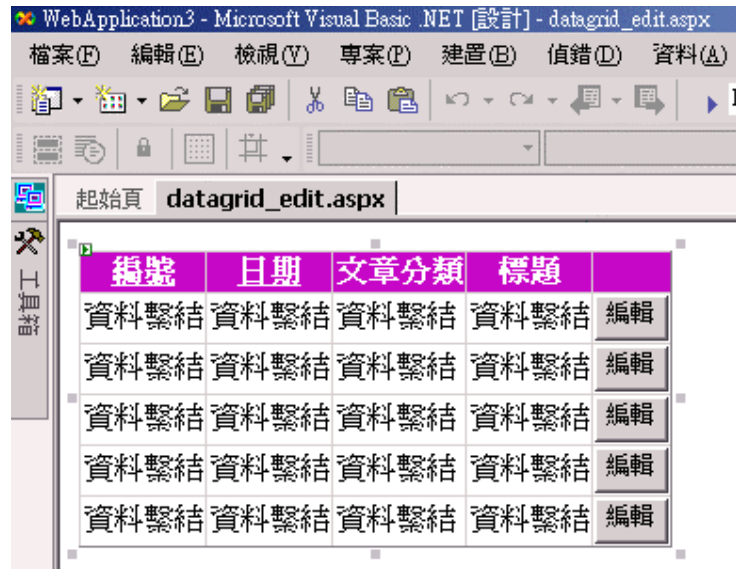


圖 4 完成的畫面（採用「編輯按鈕」）

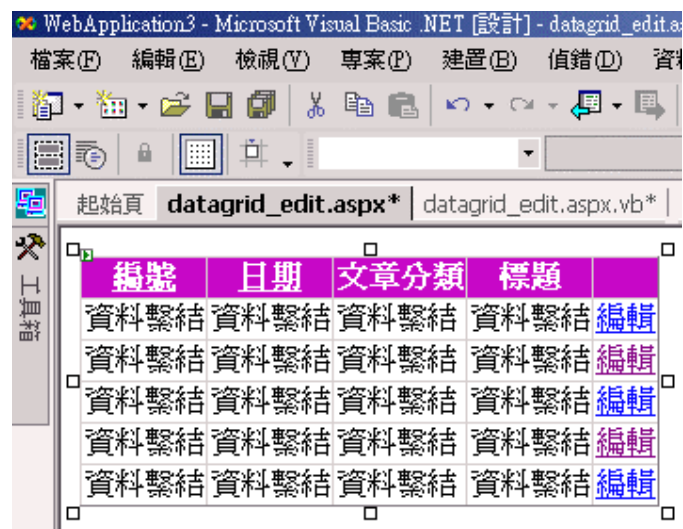


圖 5 採用「超連結按鈕」的成果

詳細的 DataGrid 程式碼如下，<Columns>標籤之間的程式，便是 DataGrid 的自訂標題欄位與編輯按鈕：

```
<asp:datagrid id="DataGrid1" runat="server" DataKeyField="id" AllowSorting="True"
AutoGenerateColumns="False">

<HeaderStyle Font-Bold="True" HorizontalAlign="Center" ForeColor="White"
BackColor="#C000C0"></HeaderStyle>
```

```
<Columns>
    <asp:BoundColumn DataField="id" SortExpression="id" ReadOnly="True"
HeaderText="編號"></asp:BoundColumn>
    <asp:BoundColumn DataField="test_time" SortExpression="test_time" HeaderText="
日期" DataFormatString="{0:yyyy/MM/dd}"></asp:BoundColumn>
    <asp:BoundColumn DataField="class" HeaderText="文章分類"></asp:BoundColumn>
    <asp:BoundColumn DataField="title" HeaderText="標題"></asp:BoundColumn>

    <asp:EditCommandColumn ButtonType="PushButton" UpdateText="更新
" CancelText="取消" EditText="編輯"></asp:EditCommandColumn>
</Columns>
</asp:datagrid>
```

## X-2 撰寫後置程式碼（Code Behind）

完成了畫面的設計，現在開始進程式設計的部份。

我們首先來設計「編輯」按鈕的程式。當使用者按下「編輯」按鈕，畫面就會變成：

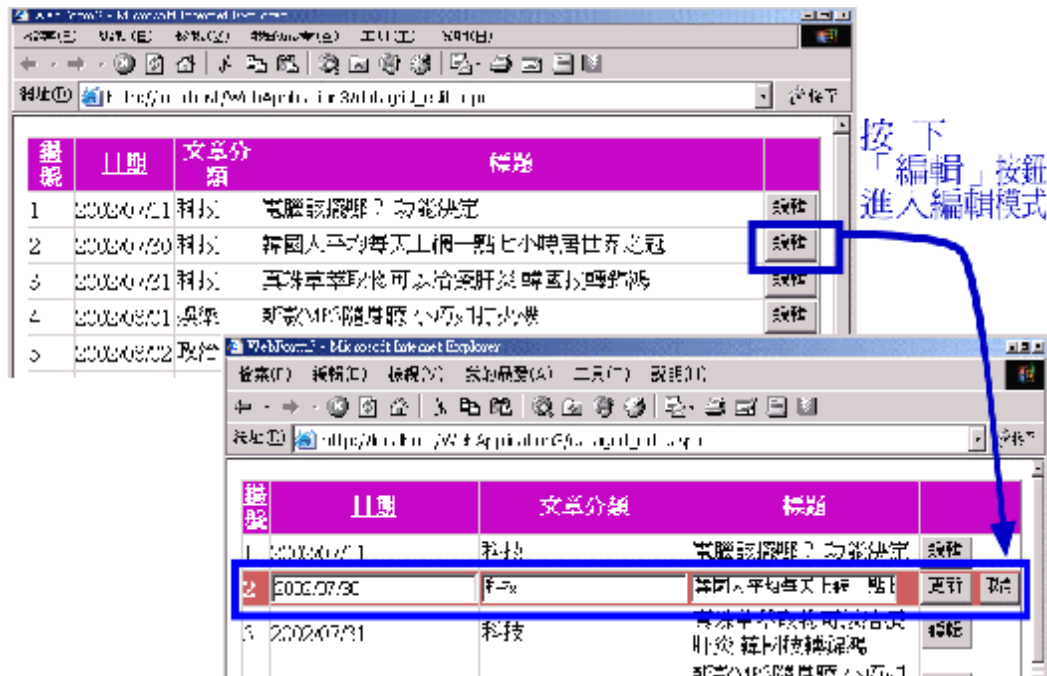


圖 6 按下「編輯」按鈕，便會進入編輯模式，這時候便可以修改這一系列資料。

## x-2-1 編輯

看完上面的圖片解說後，DataGrid 的資料編輯，真是一個很不可思議的功能吧！更讓人驚訝的是：這麼強大的功能，只要兩段程式碼便可以完成呢。

首先，請將「編輯」所需的程式，撰寫在 `DataGrid1_EditCommand()` 副程式裡面，如下圖。

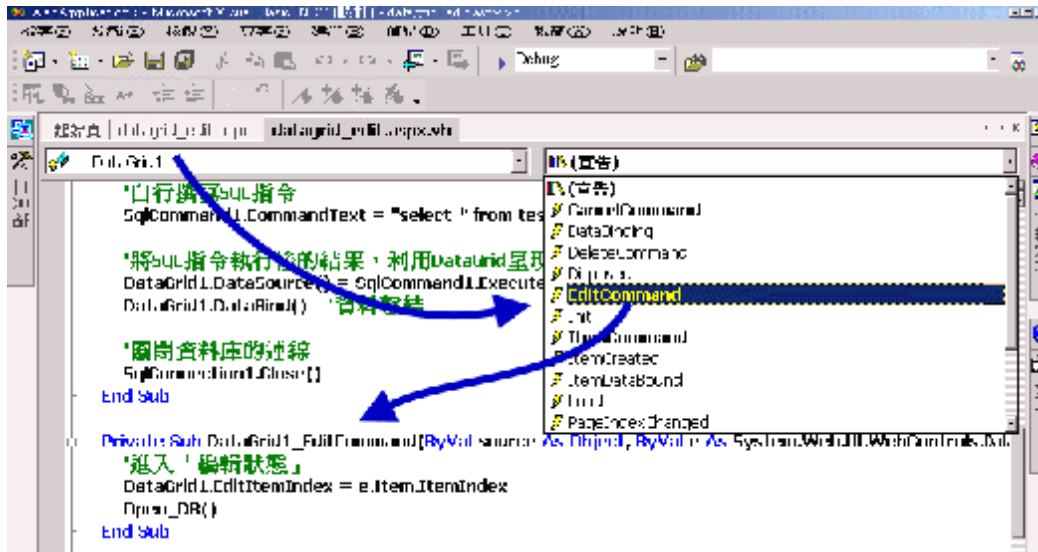


圖 7 進入「編輯」模式的程式碼

簡單的兩行程式碼（如下），卻啟動了這麼棒的資料編輯功能。

當使用者點選某一列的資料「編輯」按鈕（`e.Item.ItemIndex`），就啟動了 `DataGrid1` 這一系列資料的資料編輯模式（`DataGrid1.EditItemItemIndex`）。

```
Private Sub DataGrid1_EditCommand(ByVal source As Object, ByVal e As
System.Web.UI.WebControls.DataGridCommandEventArgs) Handles
DataGrid1.EditCommand
```

‘進入「編輯狀態」。

‘`e.Item.ItemIndex` 表示「使用者點選的那一列」的編號。

```
DataGrid1.EditItemIndex = e.Item.ItemIndex
```

‘進入編輯模式之後，仍然要重新連結資料庫，做資料繫結（`DataBind`）的動作。不然的話，即使進入編輯模式，文字輸入方塊 `TextBox` 裡面會是完全空白的！請特別留意！！

```
Open_DB()
```

```
End Sub
```

程式執行結果，如下圖。雖然進入了編輯模式，但是這裡有一個不合理的地方，



就是「主索引鍵」怎麼可以允許修改呢？

我們都知道：資料表的主索引鍵，應該是「獨一無二的」！如果允許使用者任意修改，造成錯誤或是重複，將是一場大災難。

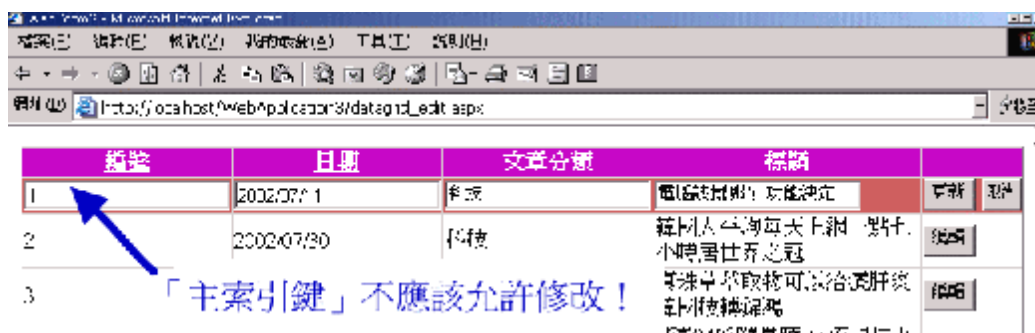


圖 7-1 DataGrid 的「資料編輯模式」  
(DataGrid1.EditItemIndex)

爲了避免這種狀況，我們必須在 DataGrid 的「屬性產生器」裡面，將「主索引鍵」這個欄位（即下圖的「編號」欄位，對應 test 資料表的主索引鍵--id）設定成「唯讀（Read-Only）」。如此一來，即使進入編輯模式，這個欄位也不允許修改了。

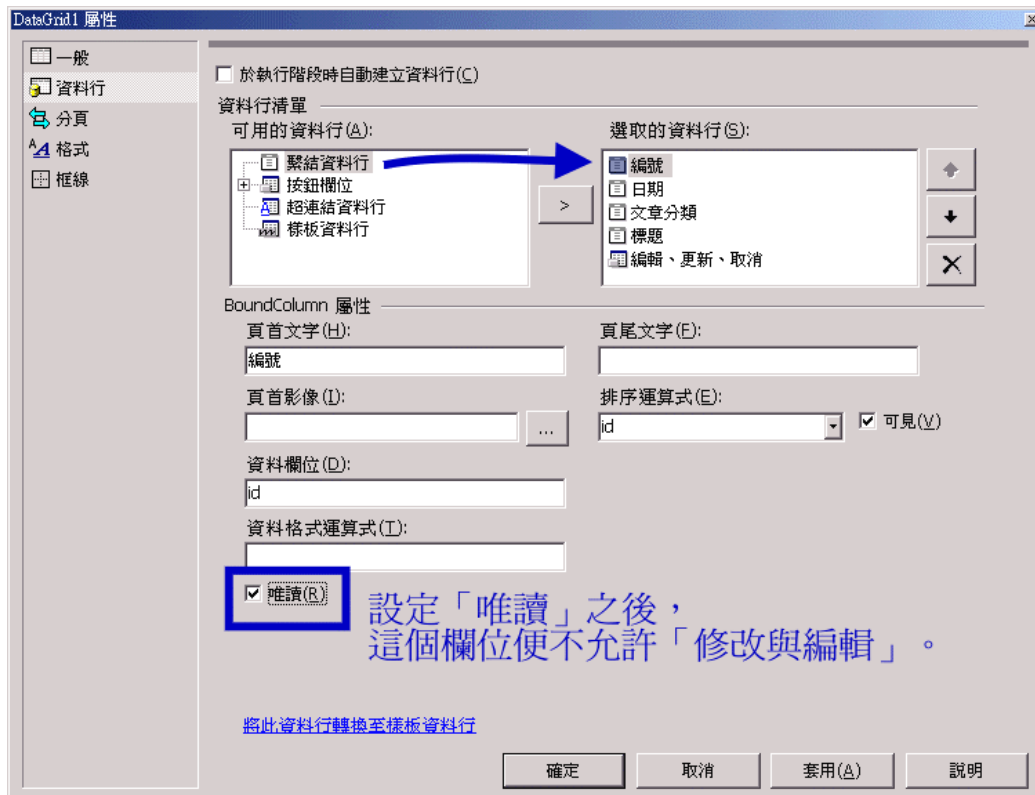


圖 7-2 將「主索引鍵」設定成「唯讀」，  
即使進入編輯模式，也不可修改之

修改之後的結果，如下圖，我們現在已經可以確保「主索引鍵」是唯讀( Read-Only )的狀態，即是在編輯模式下也不會被修改了。

編號	日期	文章分類	標題	
1	2002/07/11	科技	電腦該擺哪？功能決定	編輯
2	2002/07/30	科技	韓國人平均每天上網一點七	更新 取消
3	2002/07/31	科技	真珠草萃取物可以治療 肝炎 韓國技轉錦鴻	編輯

圖 8 DataGrid 的「資料編輯模式」  
(DataGrid1.EditItemIndex)  
主索引鍵，不允許修改（唯讀狀態）

關於資料庫的連結、SQL 指令的執行與 DataGrid 的資料繫結，我們已經將這段常用的程式，寫成一個 Open\_DB( )副程式了：

Sub Open\_DB( ) 這個副程式，是用來連結資料庫與展現資料！

'第一，開啟資料庫的連線

```
SqlConnection1.Open()
```

'第二，自行撰寫 SQL 指令

```
SqlCommand1.CommandText = "select * from test"
```

'第三，將 SQL 指令執行後的結果，利用 DataGrid 呈現在畫面上

```
DataGrid1.DataSource() = SqlCommand1.ExecuteReader
```

```
DataGrid1.DataBind() '資料繫結
```

'第四，關閉資料庫的連線

```
SqlConnection1.Close()
```

```
End Sub
```

## x-2-2 取消

在資料編輯模式中，如果想要「取消」編輯狀態，恢復到 DataGrid 資料呈現的畫面。相關程式也十分簡單。我們先看一下程式執行的結果：

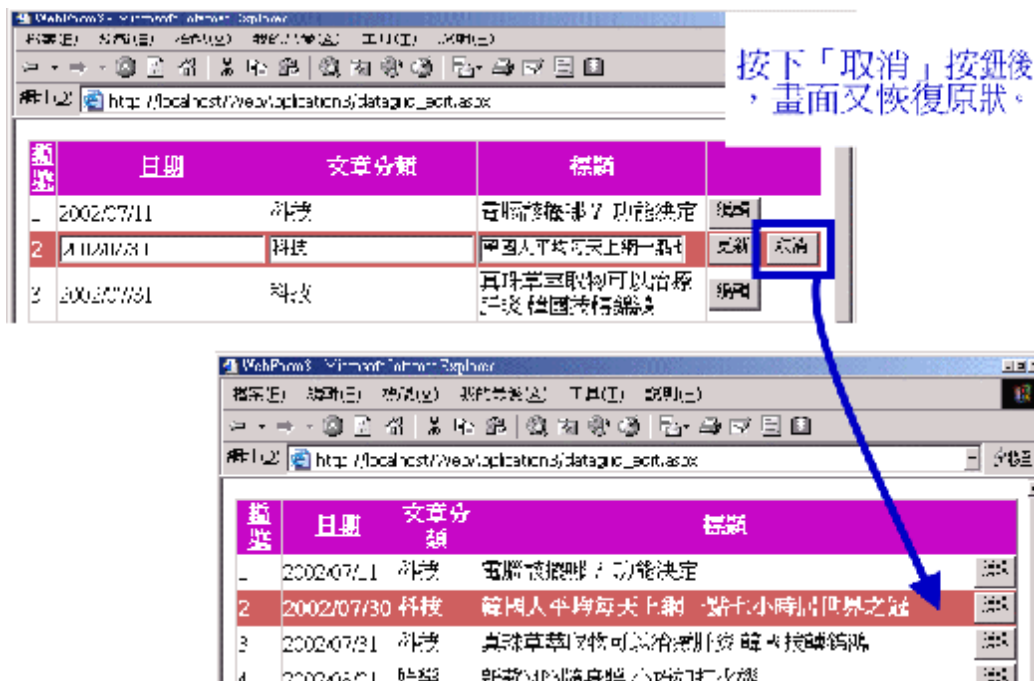


圖 9 按下「取消」按鈕之後，便可以離開編輯狀態。

將編輯狀態「取消」的副程式為----DataGrid1\_CancelCommand()，我們可以利用

用 Visual Studio .NET 來自動產生這個副程式的名稱與相關參數。

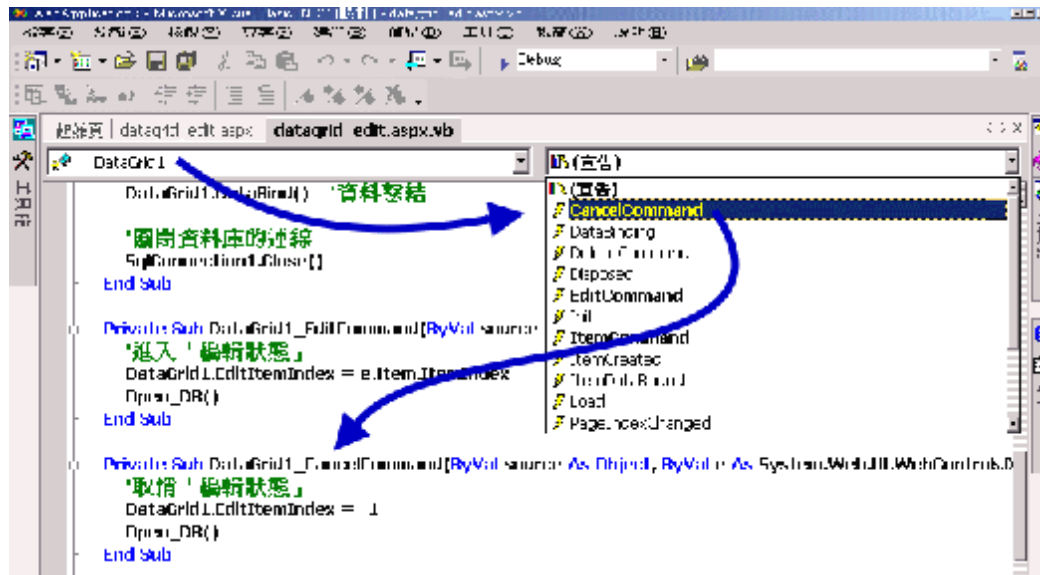


圖 10 「取消」按鈕的程式碼

```
Private Sub DataGrid1_CancelCommand(ByVal source As Object, ByVal e As  
System.Web.UI.WebControls.DataGridCommandEventArgs) Handles  
DataGrid1.CancelCommand
```

取消「編輯狀態」，恢復成 DataGrid 資料呈現的狀態

**DataGrid1.EditItemIndex = -1**

‘注意！！「取消」編輯模式之後，仍然要重新連結資料庫，做資料繫結 (DataBind) 的動作。不然的話，畫面還是會繼續停留在「編輯模式」。

**Open\_DB()**

```
End Sub
```

想要把 DataGrid 的編輯狀態 (EditItemIndex) 給取消掉，只要使用「-1」就可以恢復到原本的畫面狀態。「-1」是個重要的關鍵字，我們將會時常見到它！

### x-2-3 更新

學會了進入「編輯」模式與「取消」編輯模式的基本觀念之後，就可以進入最後的階段了。「更新 (Update)」的程式部份，難度較高！請特別注意囉！

我們可以利用 Visual Studio .NET 來自動產生這個副程式的名稱與相關參數，資料更新的副程式----DataGrid1\_UpdateCommand()。

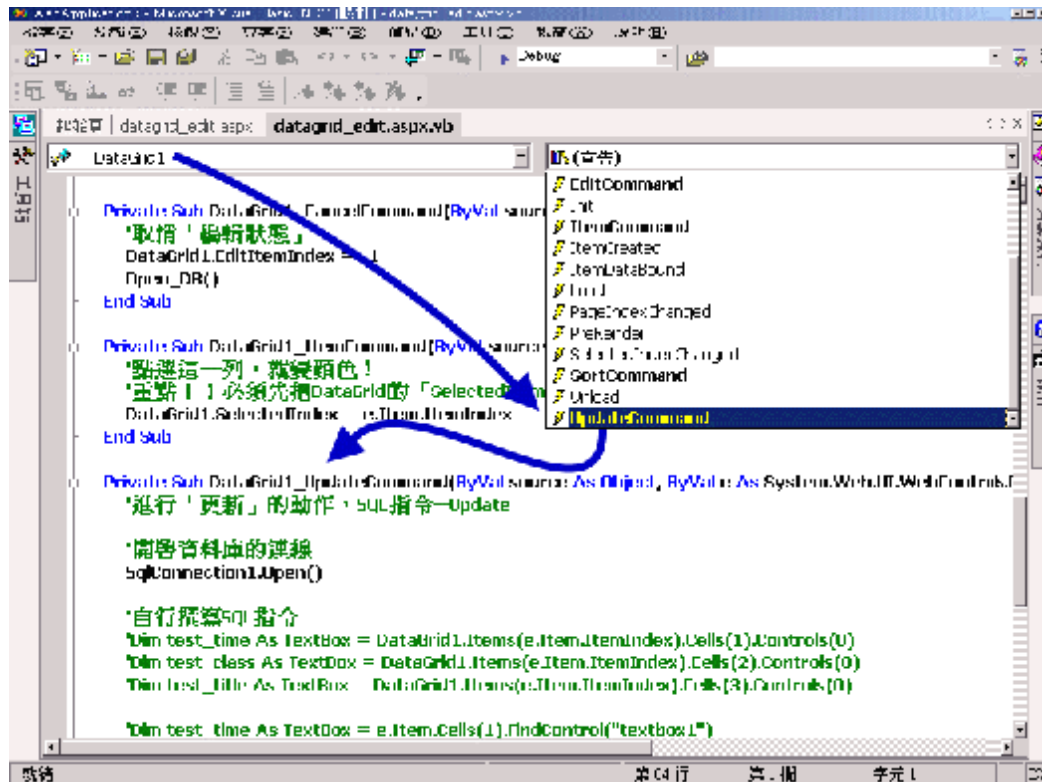


圖 11 資料更新的副程式----DataGrid1\_UpdateCommand()

我們進入 DataGrid 的編輯模式之後，DataGrid 會把這一系列資料，依照欄位的不同，放進不同的文字方塊（TextBox）裡面。我們只要能將這些文字方塊裡面的修改「文字（TextBox.Text）」，重新寫回資料表，就可以完成更新的動作了。概念很簡單，但是程式的撰寫稍微繁複了些。

編號	日期	文章分類	標題	操作
1	2002/07/11	科技	電腦該操哪？ 尚靠決定	刪除
2	2002/07/06	生活	韓國人平均每天工作一週	更新 刪除
3	2002/07/31	科技	真球草藥藥物可以治療 肝炎 韓國換轉轉	刪除

文字輸入方塊（TextBox）

文字輸入方塊（TextBox）裡面的「字」，  
就是**TextBox.Text**

圖 12 如何取得文字方塊 (TextBox) 裡面的字？

要擷取 DataGrid 編輯狀態下的各個文字輸入方塊 (TextBox) 並不容易。我們必須先定義一個 TextBox 的變數，例如：test\_time，表示我要擷取的是上圖裡面，日期欄位的那一個「文字方塊 (TextBox)」。

```
'取得「編輯狀態」下的“第二個”文字方塊 (TextBox)  
Dim test_time As TextBox = e.Item.Cells(1).Controls(0)
```

e.Item 大家已經使用過很多次了，它表示「使用者點選的那一行」。而 Cells(1) 表示這一行資料的「第二個格子」，因為電腦是從「零」算起，所以第二個格子被稱為 Cells(1)。第三個格子被稱為 Cells(2)，以此類推。



圖 13

已經可以利用 e.Item.Cells(數字).Controls(0) 來擷取 DataGrid 的資料列裡面的特定格子，那麼，我們也可以取出「格子裡面的字」，TextBox.Text 代表著「文字輸入方塊」裡面的「字」。

```
'取得「編輯狀態」下的“第二個”文字方塊 (TextBox)  
Dim test_time As TextBox = e.Item.Cells(1).Controls(0)
```

‘不相信的話，可以把 test\_time 這個文字輸入方塊 (TextBox) 裡面的這行字給印出來啊！（結果請看下圖）

```
Response.Write(“您剛剛修改的字，就是----“ & test_time.Text )
```

編號	日期	文章分類	標題	
1	2002/07/11	科技	電腦該擺哪？功能決定	編輯
2	2002/07/30	科技	韓國人平均每天上網一點七小時居世界之冠	更新 取消

進入編輯狀態，呈現出原本的資料

修改之後，按下「更新」按鈕，呈現出下列的畫面：

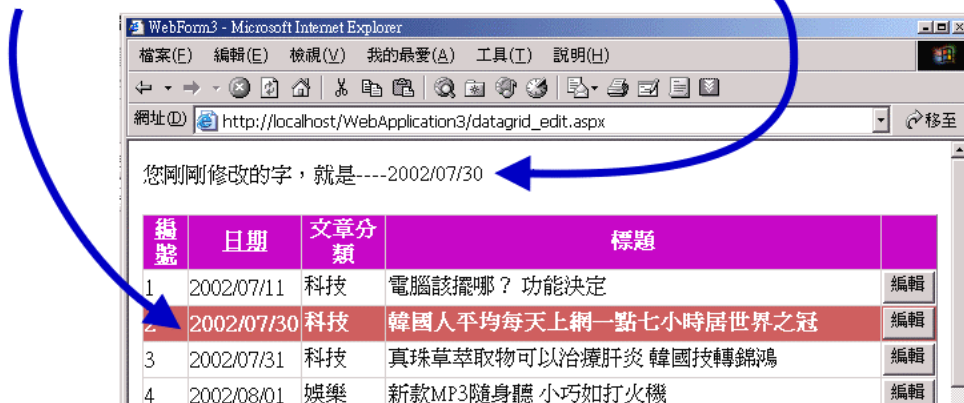


圖 15

除了上述的寫法之外，本書也整理出其他的方式，都可以擷取到 DataGrid 裡面「某一行」資料的「某一個格子」。

關於取得「編輯狀態」的文字方塊（TextBox）還有其他兩種寫法供您參考，任選一種使用即可：

‘第一種：

```
Dim test_time As TextBox =
DataGrid1.Items(e.Item.ItemIndex).Cells(1).Controls(0)
Dim test_class As TextBox =
DataGrid1.Items(e.Item.ItemIndex).Cells(2).Controls(0)
Dim test_title As TextBox =
DataGrid1.Items(e.Item.ItemIndex).Cells(3).Controls(0)
```

‘第二種：

```
Dim test_time As TextBox = e.Item.Cells(1).FindControl("textbox1")
Dim test_class As TextBox = e.Item.Cells(2).FindControl("textbox2")
Dim test_title As TextBox = e.Item.Cells(3).FindControl("textbox3")
```

一旦能正確擷取到文字輸入方塊（TextBox）裡面的修改文字，我們就可以把這

些修改後的「字 (TextBox.Text)」，重新寫回資料表，做資料更新 (update) 的動作。

'進行「更新」的動作，使用 SQL 指令--Update

'開啟資料庫的連線

SqlConnection1.Open()

'取得「編輯狀態」的各個文字方塊 (TextBox)

Dim test\_time As TextBox = e.Item.Cells(1).Controls(0)

Dim test\_class As TextBox = e.Item.Cells(2).Controls(0)

Dim test\_title As TextBox = e.Item.Cells(3).Controls(0)

'撰寫更新 (Update) 的 SQL 指令

SqlCommand1.CommandText = "update test set test\_time = " & test\_time.Text & ",class = " & test\_class.Text & ",title = " & test\_title.Text & " where id = " & DataGrid1.DataKeys(e.Item.ItemIndex) & ""

'執行 SQL 指令，進行更新 (Update)

SqlCommand1.ExecuteNonQuery()

'關閉資料庫的連線

SqlConnection1.Close()

上面這段更新資料的程式碼，大家應該不難理解。只有兩個重點：

- 要鎖定使用者點選的那一列資料的「主要索引鍵 (Primary Key)」，就是 DataGrid1.DataKeys( e.Item.ItemIndex )。使用之前，必須先在 DataGrid 的設計畫面，先設定好「DataKeyField」屬性。



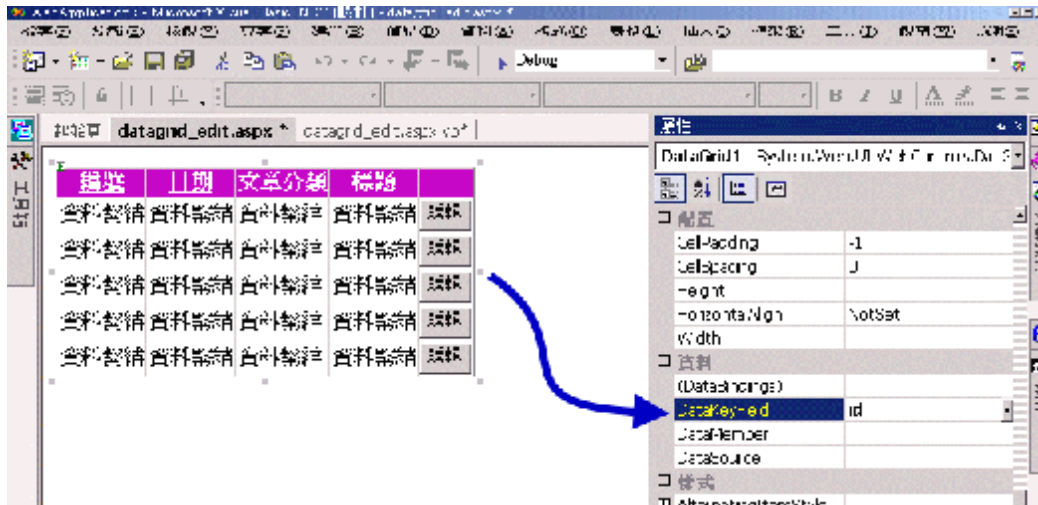


圖 15 設定「主索引鍵」----DataKeyField

- 因為我們現在處理的動作是「資料更新」，所以我們必須使用 `ExecuteNonQuery` 方法來執行 SQL 指令（update）。重新幫大家複習一遍：
  - 使用 `select` 的 SQL 指令，請採用 `ExecuteReader` 方法來執行。
  - 使用 `update`、`delete`、`insert` 的 SQL 指令，請採用 `ExecuteNonQuery` 方法來執行。

完成資料更新之後，別忘了要離開 DataGrid 的「編輯模式」喔！

```
Private Sub DataGrid1_UpdateCommand(ByVal source As Object, ByVal e As
System.Web.UI.WebControls.DataGridCommandEventArgs) Handles
DataGrid1.UpdateCommand
```

資料更新（Update）的相關程式碼！在此先省略.....

'重點！！更新成功之後，取消「編輯」模式，重新利用 DataGrid 展現資料！

```
DataGrid1.EditItemIndex = -1
```

```
Open_DB()
```

```
End Sub
```

經過漫長的歷程，「資料更新」這部份完整的程式碼如下：

```
Private Sub DataGrid1_UpdateCommand(ByVal source As Object, ByVal e As  
System.Web.UI.WebControls.DataGridCommandEventArgs) Handles  
DataGrid1.UpdateCommand
```

‘進行「更新」的動作，SQL 指令--Update

‘開啓資料庫的連線

SqlConnection1.Open()

‘取得「編輯狀態」的各個文字方塊（TextBox）

Dim test\_time As TextBox = **e.Item.Cells(1).Controls(0)**

Dim test\_class As TextBox = **e.Item.Cells(2).Controls(0)**

Dim test\_title As TextBox = **e.Item.Cells(3).Controls(0)**

‘撰寫更新（Update）的 SQL 指令

```
SqlCommand1.CommandText = "update test set  test_time = '" &  
test_time.Text & "',class = '" & test_class.Text & "',title = '" & test_title.Text &  
" where id = '" & DataGrid1.DataKeys(e.Item.ItemIndex) & "'"
```

‘執行 SQL 指令，進行更新（Update）

SqlCommand1.**ExecuteNonQuery()**

‘關閉資料庫的連線

SqlConnection1.Close()

‘重點！！更新成功之後，取消「編輯」模式，重新利用 DataGrid 展現資料！

DataGrid1.EditItemIndex = **-1**

Open\_DB()

```
End Sub
```