

第十章 常用控制項(二)



10.1 ListBox/CheckedListBox
/ComboBox清單控制項

10.4 HScrollBar/VScrollBar/
TrackBar捲軸控制項

10.2 ImageList影響清單控制項

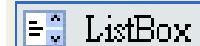
10.5 RichTextBox豐富文字方塊

10.3 MonthCalendar/DateTime
Picker日期時間控制項

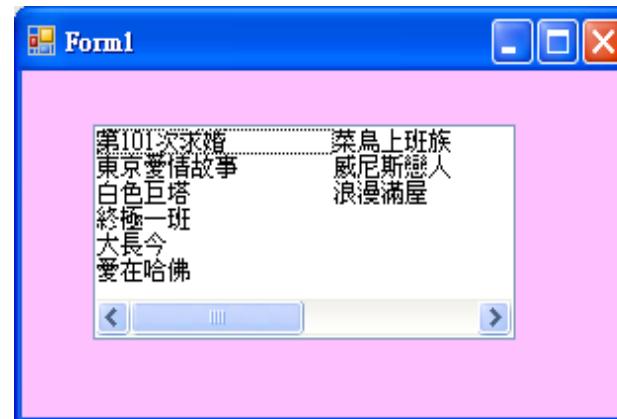
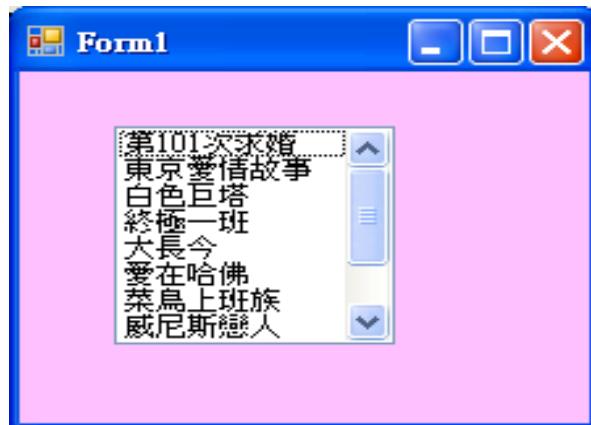
備註：可依進度點選小節



10.1.1 ListBox清單控制項



- 清單控制項提供文字選項供使用者選取
- 清單控制項大小是固定，選項太多透過捲軸移動。
- 清單選項可分多行顯示、可單選也可多選。
- 清單中文字項目輸入方式可在設計階段輸入，
也可在程式執行階段使用 Add 方法來新增項目。





二. ListBox常用屬性

屬性	說明
Items	用來設定控制項的項目集合，設定方法請參閱上面清單控制項的建立方法。
Sort	設定清單控制項內的項目是否依照字母順序排列，預設值為 False 表示不照字母排序。
MultiColumn	設定項目是否可以用多欄的方式顯示，預設值屬性值為 False 表示以單行顯示；若設為 True 可多行顯示。MultiColumn 屬性值為 True 時，結果如下圖所示：
ColumnWidth	當 MultiColumn 屬性值為 True 時，用來設定欄寬（單位為 Pixels）。預設值為 0，使用預設寬度。
ItemHeight	取得清單中項目的高度，其單位為 pixels。我們設定字型大小時，系統會自動調整 ItemHeight 的屬性值來顯示完整的文字，所以不能手動設定。



SelectionMode	設定清單控制項可讓使用者選取多少個項目，其屬性值有： ① One：只能選取一個項目(預設值)。 ② MultiSimple：可不連續選取多個項目。 ③ MultiExtended：可用滑鼠拖曳或加 Shift 鍵連續選取多個項目，或是加 Ctrl 鍵選取不連續項目。 ④ None：無法選取。
Text	Text 屬性只能在執行階段使用，其值等於使用者所選取的第一個項目。若允許使用者選取多個項目，就可以用下列 SelectedItems 屬性來取得被選取的項目。
SelectedItems	程式執行階段中，用來取得或設定清單控制項內的被選取的項目。例如將選取的第一個項目指定給 label1，寫法如下： <code>label1.Text = listBox1.SelectedItems[0].ToString();</code>
SelectedIndex	取得被選取項目的索引值，第 1 個項目的索引值為 0，第 2 個項目的索引值為 1...。
Item.Count	用來取得清單控制項有多少個項目。程式設計階段中使用。



三. ListBox常用方法

1. Add方法

在執行階段新增清單項目，加入的項目自動加到清單的最後面。寫法：

```
listBox1.Items.Add("航海王");
```

2. AddRange方法

一次新增多個項目。可將存放在字串陣列中的項目一次就加入到清單控制項中。

寫法：

```
string[] Drama = new string[]{"變形金剛2", "哈利波特"};
listBox1.Items.AddRange(Drama);
```



3. Insert方法

在執行階段將文字項目插入到清單中所指定索引值的位置，不是最後面。加入一個項目成為第三項目(索引值為2)寫法：

```
listBox1.Items.Insert(2, "冰原歷險記3-恐龍現身");
```

4. Remove方法

在執行階段，將清單控制項的文字項目移除，可指定索引值或項目名稱來做移除。移除一個項目名稱為“來自地獄”的選項，寫法：

```
listBox1.Items.Remove("來自地獄");
```

若要移除第三個項目(索引值為2)，寫法：

```
listBox1.Items.Remove(listBox1.Items[2]);
```



5. Clear方法

在執行階段，將 `listBox1` 清單控制項所有選項全部移除，寫法：

```
listBox1.Items.Clear();
```

6. SetSelected方法

在執行階段，設定清單控制項中指定項目被選取或不被選取。設定第二個項目被選取，第三個項目不被選取：

```
listBox1.SetSelected(1, true); // 第二個項目被選取
```

```
listBox1.SetSelected(2, false); // 第三個項目不被選取
```



7. ClearSelected方法

SetSelected方法將參數設為**false**，可取消選取消單控制項中指定項目。如想一次將選取的項目全部取消選取，就可用 **ClearSelected** 方法。如取消使用者所有選取項目的寫法如下：

```
listBox1.ClearSelected();
```

8. GetSelected方法

在執行階段，取得清單控制項中指定項目被選取或不被選取。如傳回值為 **true**表該項目被選取；傳回值為 **false** 則未被選取。如檢查第二個項目是否被選取，若被選取則將項目名稱顯示在標籤控制項上面，寫法：

```
if (listBox1.GetSelected(1))
    label1.Text = listBox1.Items[1].ToString();
else
    label1.Text = "";
```



9. SelectedIndexChanged 常用事件

最常用的事件，當控制項的 **SelectedIndex** 屬性改變，會觸動本事件。可將使用者選擇項目時要處理的程式碼寫在此事件處理函式中。

10. 多選處理

若要提供多選處理，先將該控制項的 **SelectionMode** 屬性值設為 **MultiSimple** 或 **MultiExtended**，以便讓使用者進行多選，



button1_Click事件處理函式

透過 for 迴圈逐一判斷 listBox1 第 i 個項目是否被選取，
若被選取將項目累加至 show 字串變數，for 迴圈執行完後
透過訊息方塊顯示 show 字串變數多選的結果。

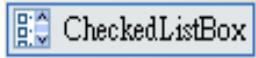
範例：ListBox1.sln

```
private void button1_Click(object sender, EventArgs e)
{
    string show = "";
    for (int i = 0; i < listBox1.Items.Count; i++)
    {
        if (listBox1.GetSelected(i))
        {
            show += listBox1.Items[i] + ", ";
        }
    }
    MessageBox.Show(show, "選擇電影");
}
```





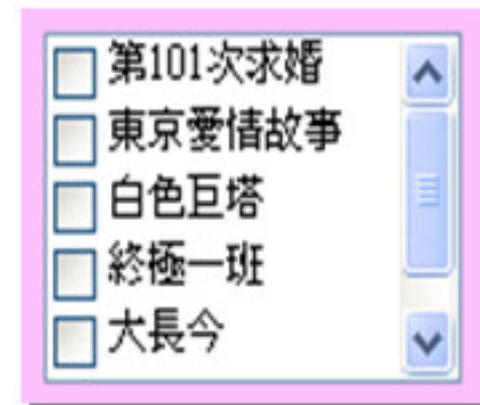
10.1.2 CheckedListBox 核取方塊清單控制項



核取方塊清單控制項比 **ListBox**

前面多一個核取方塊

- 可視為 **ListBox** 和 **CheckBox** 的結合。
- **CheckedListBox** 控制項可同時選取多個項目。
- **ListBox** 清單控制項透過 **SelectionMode** 屬性來設定單選或多選。
- **ComboBox** 僅能單選。
- **ListBox** 和 **CheckedListBox** 也可多行顯示。





一. CheckedListBox 常用屬性

屬性	說明
CheckOnClick	用來設定勾選項目滑鼠點按的次數，其屬性值有 False (必須接兩下才勾選，為預設值) 、 True (只要按一下就勾選) 。
SelectionMode	設定核取方塊清單控制項是否讓使用者選取項目，其屬性值有 One (可以選取項目，為預設值) 、 None (無法選取) 。本屬性和清單控制項不同，無法設定為 MultiSimple 和 MultiExtended 屬性，但執行時仍然可以複選。
MultiColumn	當項目數目超過控制項高度時是否採多欄顯示？若設為 True 表多欄顯示；若設為 False ，當項目數目超過控制項高度時會出現垂直捲軸。(預設值為 False)



二. CheckedListBox 常用方法與事件

1. Add方法

程式執行階段新增核取方塊清單項目的方法
所加入的項目會自動加到清單的最後。

`checkedListBox1.Items.Add("朵拉 ",true);`

`checkedListBox1.Items.Add("Momo歡樂谷 ",false);`



2. SetSelected方法

可在程式執行階段，設定核取方塊清單控制項中指定項目是否被選取。

例：設定第二個項目被選取，第三個項目不被選取：

```
checkedListBox1.SetSelected (1, true);
```

```
checkedListBox1.SetSelected (2, false);
```

3. GetItemChecked方法

在程式執行階段，知道核取方塊清單控制項中指定項目被選取或不被選取。

傳回值 **true** 表該項目被選取；傳回值 **false** 未被選取。



4. ItemCheck 事件

當使用者勾選或取消勾選項目時會觸動本事件。

將使用者勾選或取消勾選項目時，要處理程式碼寫在 ItemCheck 事件處理函式中。

5. SelectedIndexChanged 事件

- 為預設事件，當控制項項目被點選時會觸動本事件。
- 若 CheckOnClick=False (按兩下才勾選) 時，
第1次點選只會觸動 SelectedIndexChanged 事件。
第2次勾選或取消勾選時，先觸動 ItemCheck 事件，
再是 SelectedIndexChanged 事件。
- 若 CheckOnClick = True (只要按一下就勾選) 時，
點選先觸動 ItemCheck 事件，再是 SelectedIndexChanged 事件。
- 將使用者選擇項目時，要處理的程式碼寫在 SelectedIndexChanged 事件處理函式中。



三. CheckedListBox 多選處理

■ CheckedListBox 提供多選

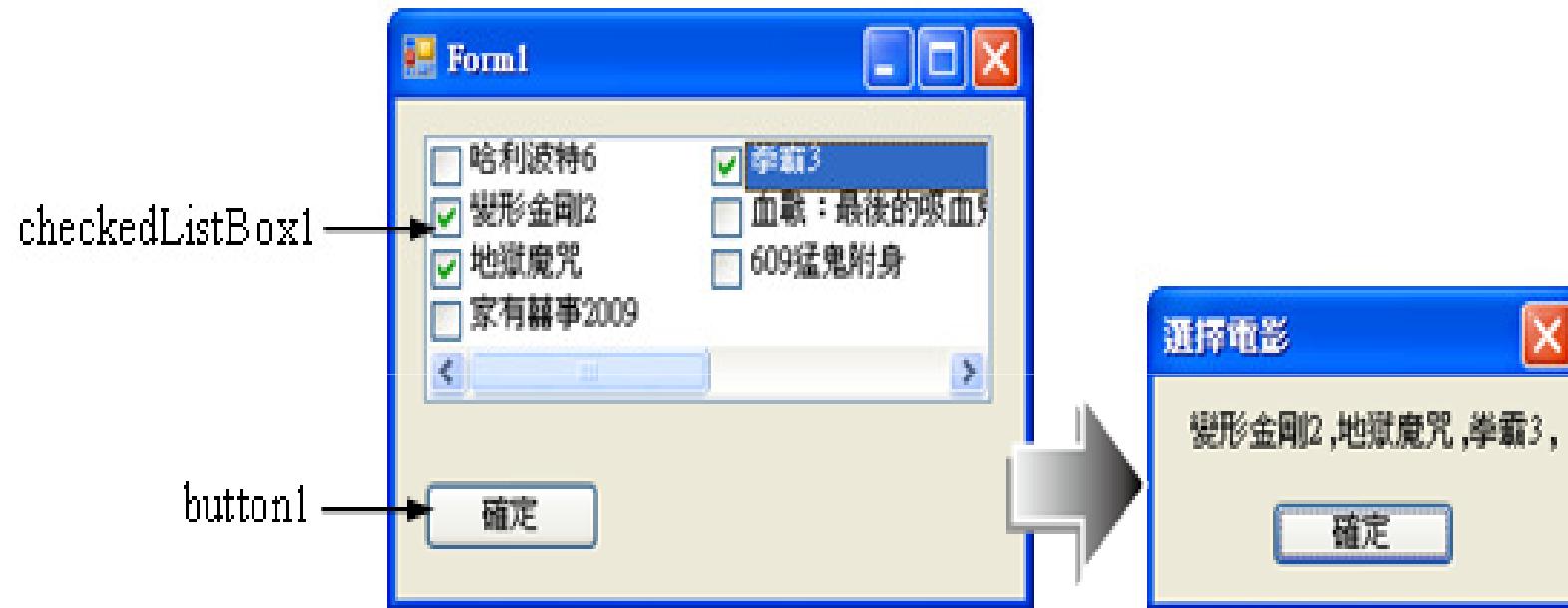
■ 透過下面程式碼：

- 當完成多選 checkedListBox1 的項目後
- 按 **確定** 鈕 執行 button1_Click 事件處理函式
- 在此函式透過 for 迴圈逐一判斷 checkedListBox1 第 i 個項目是否被選取
- 若被選取將項目累加至 show 字串變數
- for 迴圈執行完，接著透過訊息方塊顯示 show 字串變數多選的結果。

■ 範例：CheckedList Box1.sln



```
private void button1_Click(object sender, EventArgs e)
{
    string show = "";
    for (int i = 0; i < checkedListBox1.Items.Count; i++)
    {
        if (checkedListBox1.GetItemChecked (i))
        {
            show += checkedListBox1.Items[i] + " ,";
        }
    }
    MessageBox.Show(show, "選擇電影");
}
```





10.1.3 ComboBox下拉式清單控制項



下拉式清單控制項和ListBox清單控制項

大致相同。



主要差別清單控制項在表單中的大小固定，



在下拉式清單控制項多 下拉鈕，且當按下拉鈕，
下拉式清單控制項才顯示清單項目，節省版面空間。

屬性	說明
Text	取得或設定下拉式清單控制項內被選取的項目。例如設定預設選項為第 1 個項目的寫法為： <code>comboBox1.Text = comboBox1.Items[0].ToString();</code>



DropDownStyle	<p>用來設定下拉式清單顯示的樣式，其屬性值有下列三種：</p> <ul style="list-style-type: none">① DropDownList (預設值)：按下拉鈕才出現清單，文字方塊可以輸入文字資料。② DropDownList：按下拉鈕才出現清單，無法輸入文字資料。③ Simple：清單一直出現可輸入文字及由清單選取 <div style="display: flex; justify-content: space-around; align-items: center;"><div style="text-align: center;"><p>DropDown</p></div><div style="text-align: center;"><p>DropDownList</p></div><div style="text-align: center;"><p>Simple</p></div></div>
MaxDropDownItems	設定清單顯示項目的最大個數，其預設值為 8，若項目個數超過就會出現捲軸。



二. ComboBox常用事件

1. TextChanged事件

- 當在下拉式清單控制項的文字方塊中輸入文字資料時，就會觸動本事件。
- 使用者點選下拉式清單控制項中的項目時，文字方塊中的內容改變就會觸動本事件。
- 將使用者輸入文字資料時，要處理的程式碼寫在 **TextChanged** 事件處理函式中。



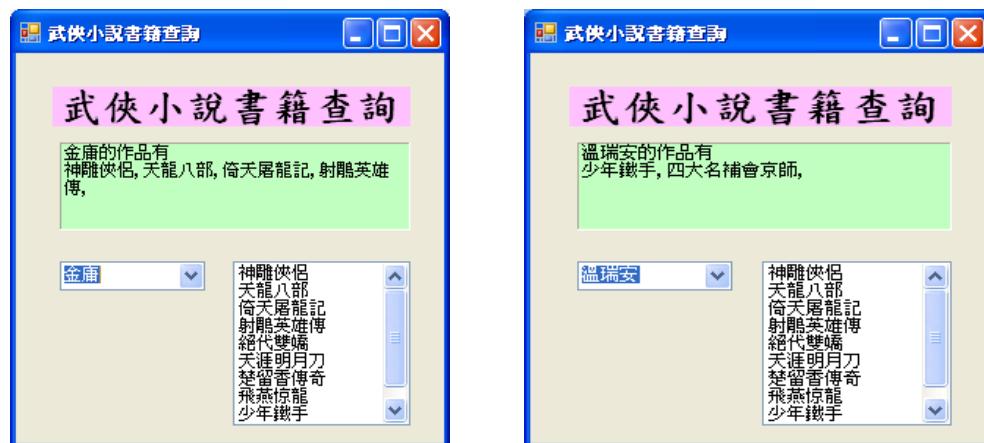
2. SelectedIndexChanged 事件

- 下拉式清單控制項預設的事件是 **SelectedIndexChanged** 事件。
- 當控制項的項目被點選時，就會觸動本事件。
- 使用者點選下拉式清單控制項的項目時，文字方塊中的內容也會改變，會同時觸動 **TextChanged** 事件。
- 兩個事件的先後順序是 **TextChanged** 先觸動，然後 **SelectedIndexChanged** 事件接著觸動。
- 通常將使用者選擇項目時要處理的程式碼寫在 **SelectedIndexChanged** 事件處理函式中。



範例演練

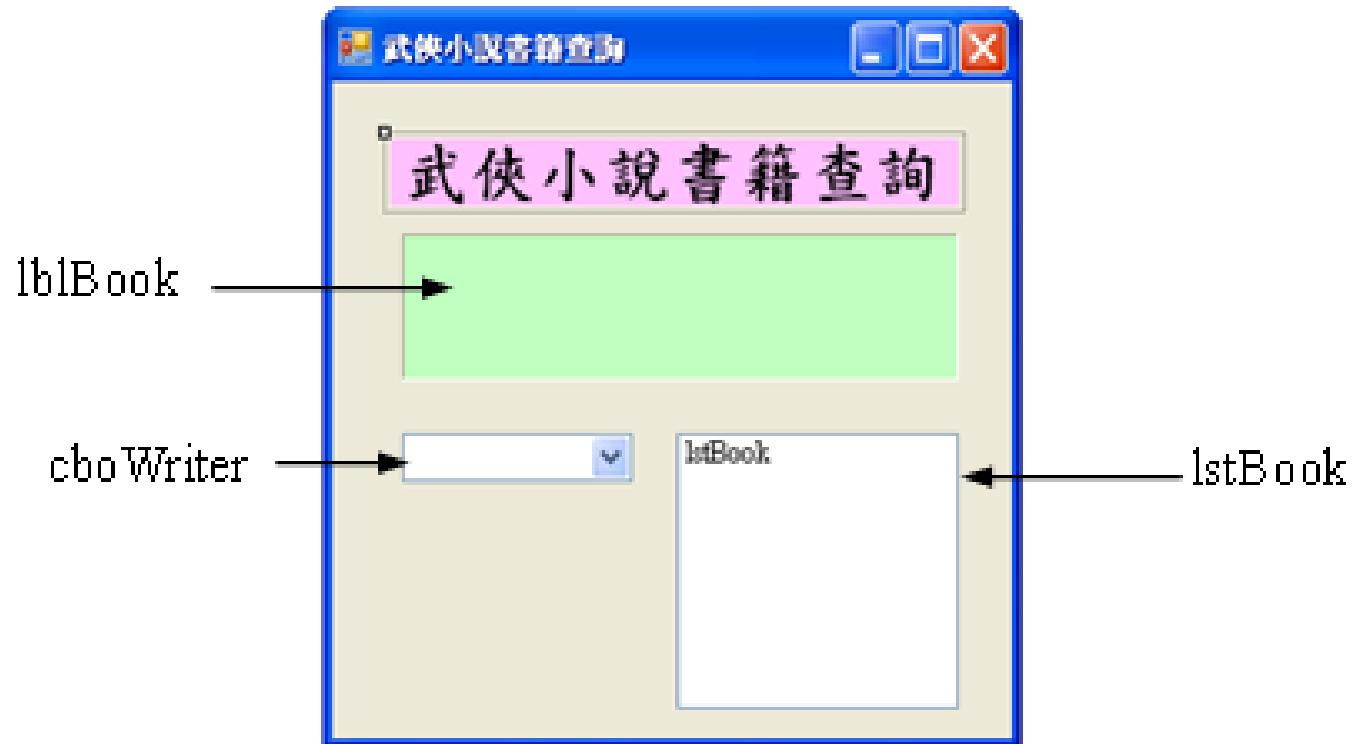
■ 在表單左邊有一個下拉式清單控制項，其中有四個武俠小說作者。右邊有一個清單控制項，有十個武俠小說著作的項目。當按下拉式清單控制項選擇作者後，標籤控制項上面會顯示該作者名字以及著作名稱。所有初值設定均在表單的 Form1_Load 事件處理函式中設定。



1. 作者：金庸
著作：神雕俠侶, 天龍八部,
倚天屠龍記, 射鵰英雄傳"
2. 作者：古龍
著作：絕代雙嬌, 天涯明月刀,
楚留香傳奇
3. 作者：臥龍生
著作：飛燕驚龍
4. 作者：溫瑞安
著作：少年鐵手, 四大名補會京師



■ Step1 設計輸出入介面





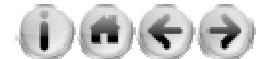
■ Step2 撰寫程式碼

```
// FileName : SearchBook.sln

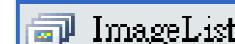
01 string[] Writer = new string[] { "金庸", "古龍", "臥龍生", "溫瑞安" };
02 string[][] Book = new string[4][];
03
04 private void Form1_Load(object sender, EventArgs e)
05 {
06     cboWriter.Items.AddRange(Writer);
07     Book[0] = new string[] { "神雕俠侶", "天龍八部", "倚天屠龍記", "射鵰英雄傳" };
08     Book[1] = new string[] { "絕代雙嬌", "天涯明月刀", "楚留香傳奇" };
09     Book[2] = new string[] { "飛燕驚龍" };
10     Book[3] = new string[] { "少年鐵手", "四大名補會京師" };
11     for (int i = 0; i <= Book.GetUpperBound(0); i++)
12     {
13         lstBook.Items.AddRange(Book[i]);
14     }
15 }
16
```



```
17 private void cboWriter_SelectedIndexChanged(object sender, EventArgs e)
18 {
19     lblBook.Text = Writer[cboWriter.SelectedIndex] + "的作品有\n";
20     for (int i = 0; i <= Book[cboWriter.SelectedIndex].GetUpperBound(0); i++)
21     {
22         lblBook.Text += Book[cboWriter.SelectedIndex][i] + ", ";
23     }
24 }
```



10.2 ImageList 影像清單控制項



- 影像清單控制項 可預先儲存很多圖檔，等需要時再將影像清單 指定給某個控制項，更換欲顯示的圖形。
- 通常 ImageList 中儲存的圖檔都是小圖示，以免佔用過多的 記憶空間。
- 工具箱中的 Label、LinkLabel、Button、RadioButton、 CheckBox、TabControl、TreeView 等控制項都有 ImageList 屬性，只要在表單上有建立 ImageList 控制項，在設計階段都會出現 ImageList 屬性下拉式清單中。
- 可在表單上建立多組影像清單控制項，以供上面控制項依需求選用。
- 在執行階段 PictureBox 控制項也可選取 ImageList 清單的圖片 置入 PictureBox 控制項上。
- 將 `imageList1` 第一張圖置入 `pictureBox1` 控制項上，寫法：
`pictureBox1.Image = imageList1.Images[0];`



10.2.1 ImageList 常用屬性

屬性	說明
Images	進入影像集合編輯器，載入或刪除清單中的圖片，其索引值由 0 開始。
ImageSize	設定儲存圖片的大小，預設為 16x16，最大值為 256x256。
ColorDepth	設定儲存圖片的色彩位元數。屬性值有： ① Depth4Bit ② Depth8Bit(預設值) ③ Depth16Bit ④ Depth32Bit



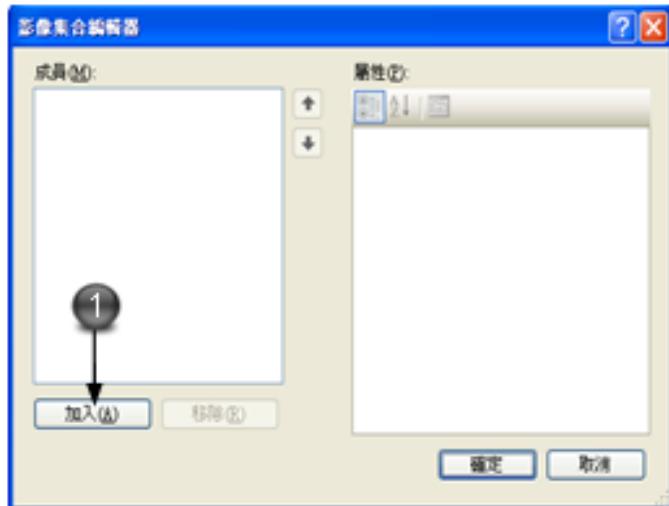
10.2.2 ImageList 圖檔的載入與移除

- **ImageList** 影像清單控制項儲存的圖檔，可透過 **ImageSize** 屬性調整成相同的大小，圖形大小或比例最好相同，以免造成變形。
- 儲存的圖檔，可透過 **ColorDepth** 屬性調整成相同的色彩位元數，建議設定以所有圖片清單中色彩最高者為準，以免色彩失真。



一. 如何在設計階段載入圖檔

- 在 **ImageList** 中加入和移除小圖片，操作方法類似 **PictureBox**。因為 **ImageList** 可以載入多張圖片，因此會出現「影像集合編輯器」來協助管理圖片。
 1. 在屬性視窗中的 **Images** 屬性值的右邊 鈕上按一下，出現左下圖「影像集合編輯器」。





- 若要移除影像清單中的圖像，選取要移除的圖檔，再按 鈕即可。
- 若要移動圖檔在集合中的順序，先選取要移動的圖檔，再按 或 鈕來調整該圖檔在集合中的位置。



二. 如何在執行階段載入圖檔

- 使用 **Add** 和 **Image.FromFile** 方法或 **new Bitmap()** 物件來設定 **Images**屬性值，達到載入圖檔的目的。
- 語法：

語法1：// path為檔案路徑

imageList1.Images.Add(Image.FromFile(path));

語法2：

imageList1.Images.Add(new Bitmap(path));

[例1] 下面兩種寫法皆可載入ok.bmp圖檔。

- ① **imageList1.Images.Add(Image.FromFile("ok.bmp"));**
- ② **imageList1.Images.Add(new Bitmap("ok.bmp"));**

[例2] 若將 **imageList1** 第一張圖放入 **pictureBox1** 上顯示
pictureBox1.Image=imageList1.Images[0];



三. 如何在執行階段移除圖檔

- 使用 **RemoveAt**方法來移除**Images**集合中的一張圖片。語法：

imageList1.Images.RemoveAt(index);

- 如要移除所有圖片，可用**Clear**方法。語法：

imageList1.Images.Clear();



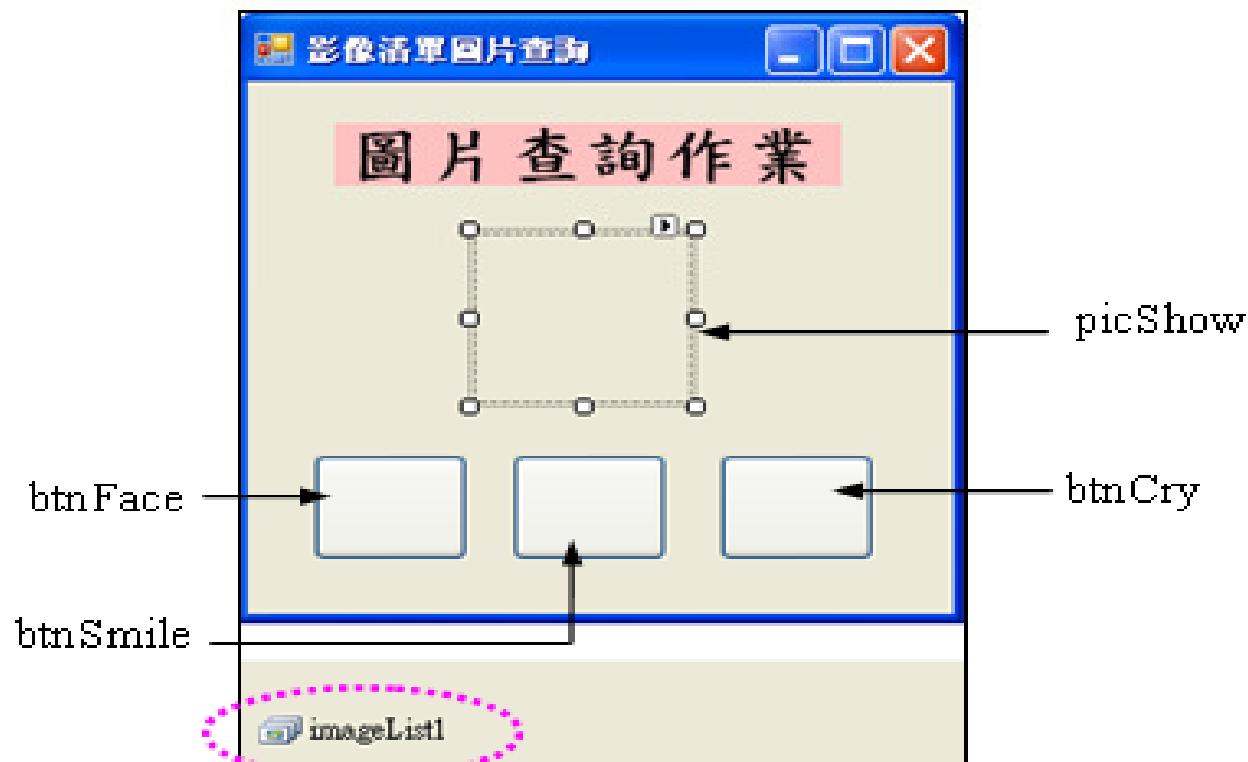
範例演練

當你在 、、 三個不同的按鈕上按一下，會將按鈕上的圖示顯示於圖片方塊控制項中，本例請用 **ImageList** 影像清單控制項存入上面三個小圖示檔 FACE0. ICO、FACE1.ICO、FACE2.ICO。





Step1 設計輸出入介面





Step2 撰寫程式碼

```
// FileName : ImageListDemo.sln
01 private void Form1_Load(object sender, EventArgs e)
02 {
03     picShow.Image = imageList1.Images[0];
04     picShow.SizeMode = PictureBoxSizeMode.StretchImage;
05 }
06
07 private void btnFace_Click(object sender, EventArgs e)
08 {
09     picShow.Image = imageList1.Images[0];
10 }
```



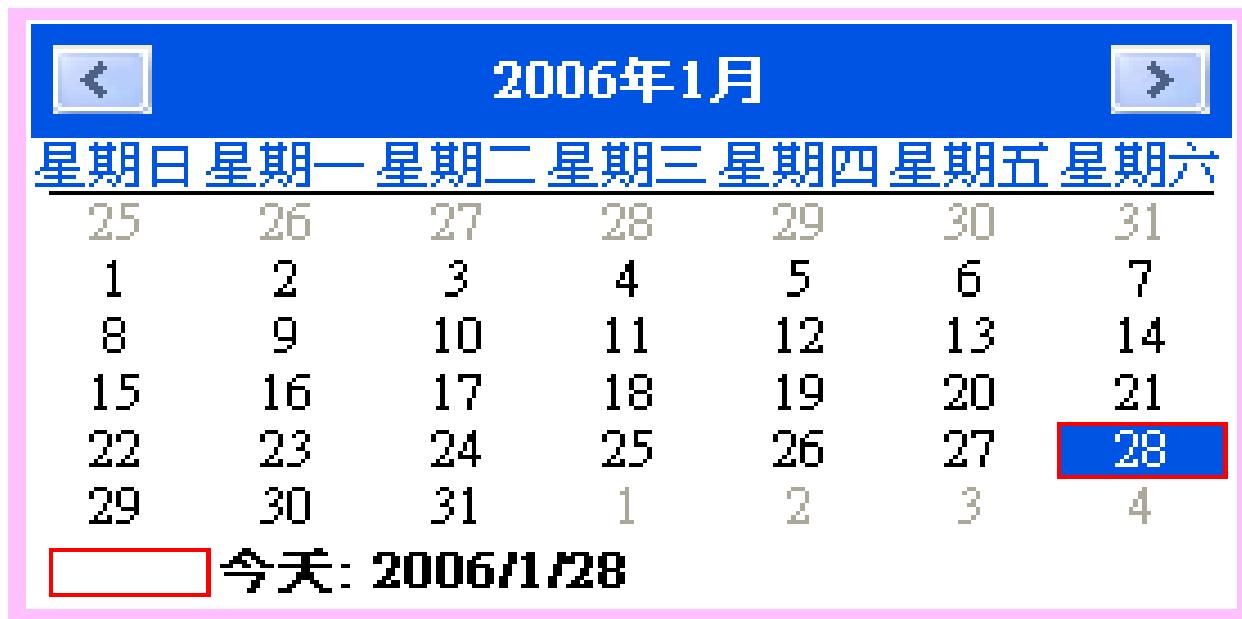
```
11
12 private void btnSmile_Click(object sender, EventArgs e)
13 {
14     picShow.Image = imageList1.Images[1];
15 }
16
17 private void btnCry_Click(object sender, EventArgs e)
18 {
19     picShow.Image = imageList1.Images[2];
20 }
```



10.3 MonthCalendar / DateTimePicker 日期時間控制項

10.3.1 MonthCalendar 月曆控制項

- 月曆控制項主要功能是顯示月曆，以及讓使用者選取日期區塊。





一. 如何建立 MonthCalendar 月曆控制項

1. 選取工具箱中  月曆控制項工具。
2. 在表單適當位置拖曳出月曆控制項，月曆控制項的大小是由系統內定，我們是無法改變。
3. 設定月曆控制項的相關屬性。



二. 如何設定**AnnuallyBoldedDates** 屬性

- 此屬性是一個日期的集合。
- 主要設定月曆控制項中每年哪些日期要用粗體字顯示
- **BoldedDates** 屬性也是一個日期的集合。
- 主要在設定月曆控制項中哪些日期要用粗體字顯示，而非每年要標示的日期。





三. MonthCalendar 月曆控制項常用屬性

屬性	說明
AnnuallyBoldedDates	設定月曆控制項中每年用粗體字顯示的日期。
BoldedDates	設定月曆中用粗體字顯示的日期陣列。
CalendarDimensions	設定月曆中月份欄和列的數目，預設值為 1,1。若設為 1,2 表示上下顯示兩個月份；若設為 2,2 表示上下左右共四個月。
TodayDate	今天的日期。
ShowToday	設定在月曆的下面是否顯示今天日期。 1. True : 顯示今天日期（預設值） 2. False : 不顯示今天日期
ShowTodayCircle	設定在月曆上是否圈出今天日期。 1. True : 框選出今天日期（預設值） 2. False : 不框選出今天日期



屬性	說明
ShowWeekNumbers	設定是否顯示週次。 1. False：不顯示週次（預設值） 2. True：顯示週次
MaxSelectionCount	設定可以連續選取日期的總天數，預設值為 7，代表最多只能選取 7 天。
SelectionStart	在程式執行階段，設定連續選取日期的起始日期。
SelectionEnd	在程式執行階段，設定連續選取日期的終點日期。
SelectionRange	設定連續選取日期的範圍。例如： 2009-12-18,2009-12-20
MaxDate	設定可以選擇日期的上限。
MinDate	設定可以選擇日期的下限。
ScrollChange	設定按月曆中 ↑、↓ 鈕改變的月數，系統預設值為 0（由系統內定）。



■ 四. MonthCalendar月曆控制項常用方法

1. AddAnnuallyBoldedDate方法

在程式執行中新增AnnuallyBoldedDates集合成員，要使用AddAnnually BoldedDate方法，其語法如下：

- `DateTime d=new DateTime(2009,9,2);`
- `monthCalendar1.AddAnnuallyBoldedDate(d);`

2. AddBoldedDate方法

在程式執行中新增BoldedDates集合成員，要使用AddBoldedDate方法，語法如下：

- `DateTime d=new DateTime(2006,9,2);`
- `monthCalendar1.AddBoldedDate(d);`



3. RemoveAllAnnuallyBoldedDate方法

在程式執行中移除所有 AnnuallyBoldedDates 集合成員，要使用 Remove AllAnnuallyBoldedDate 方法，語法：

```
monthCalendar1.RemoveAllAnnuallyBoldedDate();
```

4. RemoveAllBoldedDate方法

在程式執行中移除所有 BoldedDates 集合成員，要使用 RemoveAll BoldedDate 方法，語法：

```
monthCalendar1.RemoveAllBoldedDate();
```



五. MonthCalendar月曆控制項常用事件

1. DateChanged事件

DateChanged事件是MonthCalendar月曆控制項預設的事件，當使用者改變日期時會觸動本事件。例如使用者未選取日期，只是按 、 鈕改變的月份時，只會觸動本事件而不會觸動DateSelected事件。所以我們會將日期改變時，要處理的程式碼寫在DateChanged事件處理函式中。



2. DateSelected事件

當使用者選取日期時會觸動本事件，例如使用者點選一個日期，會先觸動**DateChanged**事件，接著觸動**DateSelected**事件。如果用拖曳方式選取連續日期，每拖曳過的日期就會觸發一次**DateChanged**事件，選取完才會觸動**DateSelected**事件。所以將使用者選取日期時，要處理的程式碼寫在**DateSelected**事件處理函式中。



10.3.2 DateTimePicker 日期挑選控制項



日期挑選控制項，和上一小節介紹的月曆控制項非常類似，常用屬性和事件都大致相同。只是日期挑選控制項多了文字方塊，可以直接輸入日期。而且月曆如下圖是採下拉式呈現，所以比較節省版面空間。



DateTimePicker 日期挑選控制項建立的方法，和 MonthCalendar 月曆控制項相同，所以就不再重複說明。



一. DateTimePicker 日期挑選控制項常用屬性

屬性	說明
Value	設定和讀取選取的日期和時間，其資料型別為 DateTime。
Format	設定文字方塊中日期的顯示格式。如下： ① Long：完整日期 <input type="button" value="2009年 8月15日"/> (預設值) ② Short：簡短日期 <input type="button" value="2009/ 8/15"/> ③ Time：時間 <input type="button" value="上午 11:45:29"/> ④ Custom：以自訂格式來顯示日期時間值。須配合 CustomFormat 屬性才有效 <input type="button" value="八月 15, of the year 2009"/>
CustomFormat	用來格式化顯示日期或時間的自訂格式字串，當 Format 屬性設為 Custom 才有效。若將此屬性設為：「 MMM dd, 'of the year' yyyy 」，在日期挑選控制項的顯示格式如下： <input type="button" value="八月 15, of the year 2009"/> 程式中的寫法如下： <code>dateTimePicker1.Format = DateTimePickerFormat.Custom;</code> <code>dateTimePicker1.CustomFormat = "MMM dd, 'of the year' yyyy";</code>



ShowCheckBox	設定是否顯示核取方塊。 ① False：不顯示  (預設值) ② True：顯示 
Checked	當 ShowCheckBox 值為 True 時，可以用來判斷使用者是否已經選取一個日期。若值為 True，則表示已選取。
ShowUpDown	設定是否上下微調按鈕  ，讓使用者點選年、月、日後按  鈕切換日期。 ① False：不顯示 (預設值) ② True：顯示 



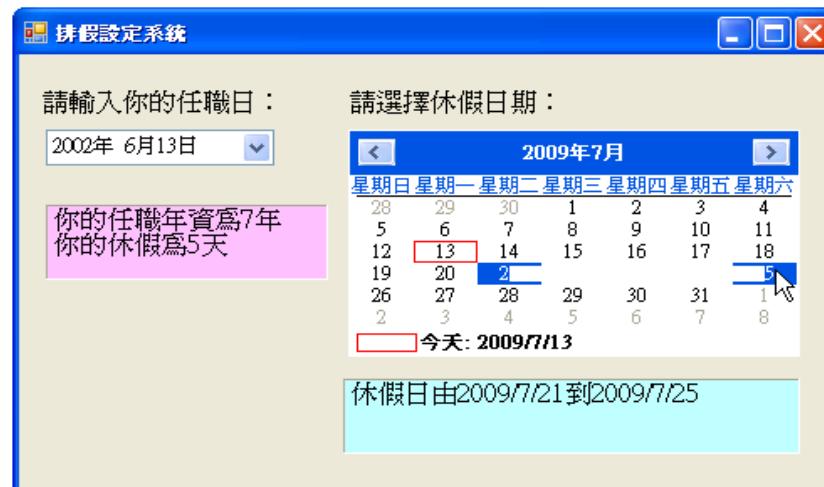
二. DateTimePicker 日期挑選控制項常用事件

- **ValueChanged**事件是DateTimePicker 日期挑選控制項預設的事件
- 當使用者改變日期時間時會觸動本事件。
- 將日期改變時，要處理的程式碼寫在**ValueChanged**事件處理函式中。



範例演練

■ 使用者輸入任職日期後，會計算出年資，並顯示可休假的日數（滿1~5年可休假3天、5年以上5天、未滿1年不能休假）。當使用者允許休假時，月曆才可以操作。程式中規定使用者要選取連續的休假日期，且不可以選取當天以前的日期。當使用者選取日期後，就顯示休假的起訖日期，若休假天數未滿也會顯示訊息。





■ Step1 設計輸出入介面





Step2 撰寫程式碼

```
// FileName : CalendarDemo.sln  
01 int vacation_days;  
02  
03 private void Form1_Load(object sender, EventArgs e)  
04 {  
05     monthCalendar1.Enabled = false;  
06     monthCalendar1.MinDate = DateTime.Now;  
07     dateTimePicker1.MaxDate = DateTime.Now;  
08     lblYears.Text = "";  
09     lblVacation.Text = "";  
10 }
```



```
12 private void dateTimePicker1_ValueChanged(object sender, EventArgs e)
13 {
14     int years;
15     years = DateTime.Now.Year - dateTimePicker1.Value.Year;
16     vacation_days = 0;
17     if (years > 5)
18     {
19         vacation_days = 5;
20     }
21     else if (years >= 1 && years <= 5)
22     {
23         vacation_days = 3;
24     }
25     else if (years < 0)
26     {
27         dateTimePicker1.Value = DateTime.Now;
28         years = 0;
29     }
30     if (vacation_days > 0)
31     {
32         monthCalendar1.Enabled = true;
33         monthCalendar1.MaxSelectionCount = vacation_days;
34         lblVacation.Text = "請選擇連續" + vacation_days + "天為休假日";
35     }
```



```
36     else
37     {
38         monthCalendar1.Enabled = false;
39         lblVacation.Text = "年資太淺！請繼續為公司努力！！";
40     }
41     lblYears.Text = "你的任職年資為" + years + "年" + Environment.
42                               NewLine + "你的休假為" + vacation_days + "天";
43
44     private void monthCalendar1_DateChanged(object sender,
45                                         DateRangeEventArgs e)
46     {
47         lblVacation.Text = "休假日由" +
48                         monthCalendar1.SelectionStart.ToShortDateString() + "到" +
49                         monthCalendar1.SelectionEnd.ToShortDateString();
50     }
51 }
```



10.4 ScrollBar / VscrollBar / TrackBar 滾軸控制項

- 滾軸類的控制項主要是用來讓使用者以拖曳方式來改變程式的設定值。
- 使用者可直覺且親切地操作。
- 可避免輸入錯誤的數值。
- 控制台鍵盤設定，可拖曳軌跡棒改變延遲的時間。





10.4.1 HscrollBar / VscrollBar 水平垂直捲軸



HScrollBar

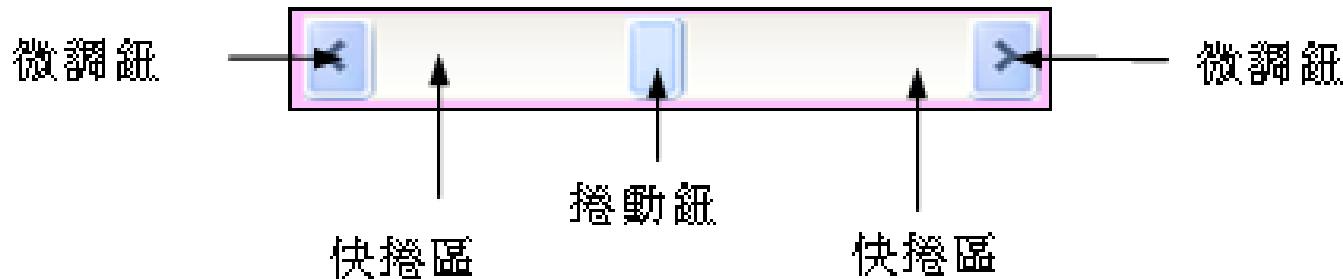
水平捲軸和

VScrollBar

垂直捲軸控制項



視窗小於內容都會出現捲軸





一. HScrollBar / VScrollBar 水平和垂直捲軸常用屬性

屬性	說明
Value	設定和讀取捲動紐位置的數值，預設值為 0。
Maximum	設定捲軸的最大值，預設值為 100。
Minimum	設定捲軸的最小值，預設值為 0。
LargeChange	設定當使用者按快捲區時，捲動紐移動的數值，預設值為 10。
SmallChange	設定當使用者按微調紐時，捲動紐移動的數值，預設值為 1。



二. HScrollBar / VScrollBar常用事件

1. Scroll事件

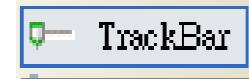
Scroll事件是捲軸控制項預設的事件，當使用者拖曳捲動鈕時會觸動本事件。

2. ValueChanged事件

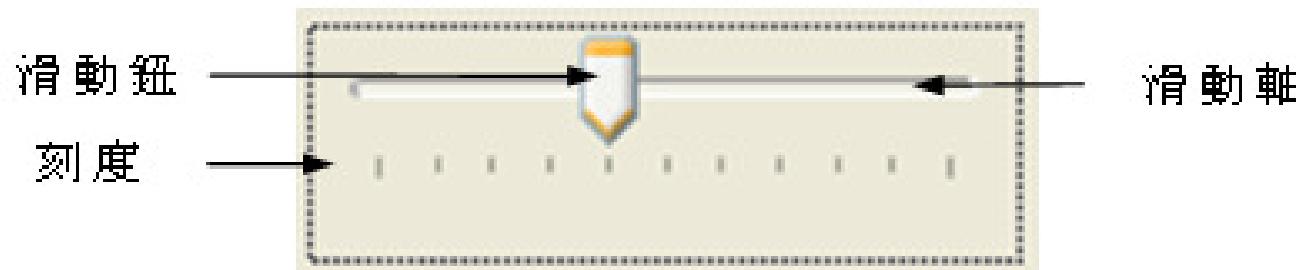
當使用者改變捲軸控制項的Value屬性值時會觸動本事件，例如按微調鈕或快捲區。



10.4.2 TrackBar 滑動桿



- 滑動桿控制項（或稱為軌跡棒）
- 功能和捲軸非常類似
- 外觀像音響面板放控制音量的滑動桿。

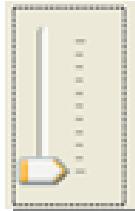
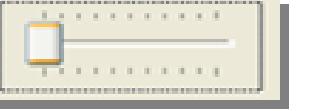




一. TrackBar 滑動桿常用屬性

屬性	說明
Value	設定和讀取滑動鈕位置的數值，預設值為 0。
Maximum	設定滑動桿控制項的最大值，預設值為 10。
Minimum	設定滑動桿控制項的最小值，預設值為 0。
LargeChange	設定當使用者按滑動軸或 PageUp、PageDown 鍵時，滑動鈕移動的數值，預設值為 5。
SmallChange	設定當使用者按方向鍵時，滑動鈕移動的數值，預設值為 1。注意滑動桿控制項必須取得駐點 (Focus)，按鍵才有作用。
TickFrequency	設定滑動桿控制項刻度的距離，屬性值為正整數，預設值為 1。



Orientation	<p>設定滑動桿控制項滑動軸的方向。</p> <p>Horizontal (水平) - 預設值 </p> <p>Vertical (垂直) </p>
TickStyle	<p>設定滑動桿控制項刻度顯示的位置。</p> <p>① BottomRight (預設值) </p> <p>② None (無刻度) </p> <p>③ TopLeft (上面或左邊) </p> <p>④ Both (兩邊都有) </p>



二. TrackBar 常用事件

1. Scroll 事件

Scroll 事件是滑動桿控制項預設的事件，當使用者拖曳滑動鈕時會觸動本事件。

2. ValueChanged 事件

當使用者改變滑動桿控制項的 **Value** 屬性值時會動本事件，例如按方向鍵或滑動軸。



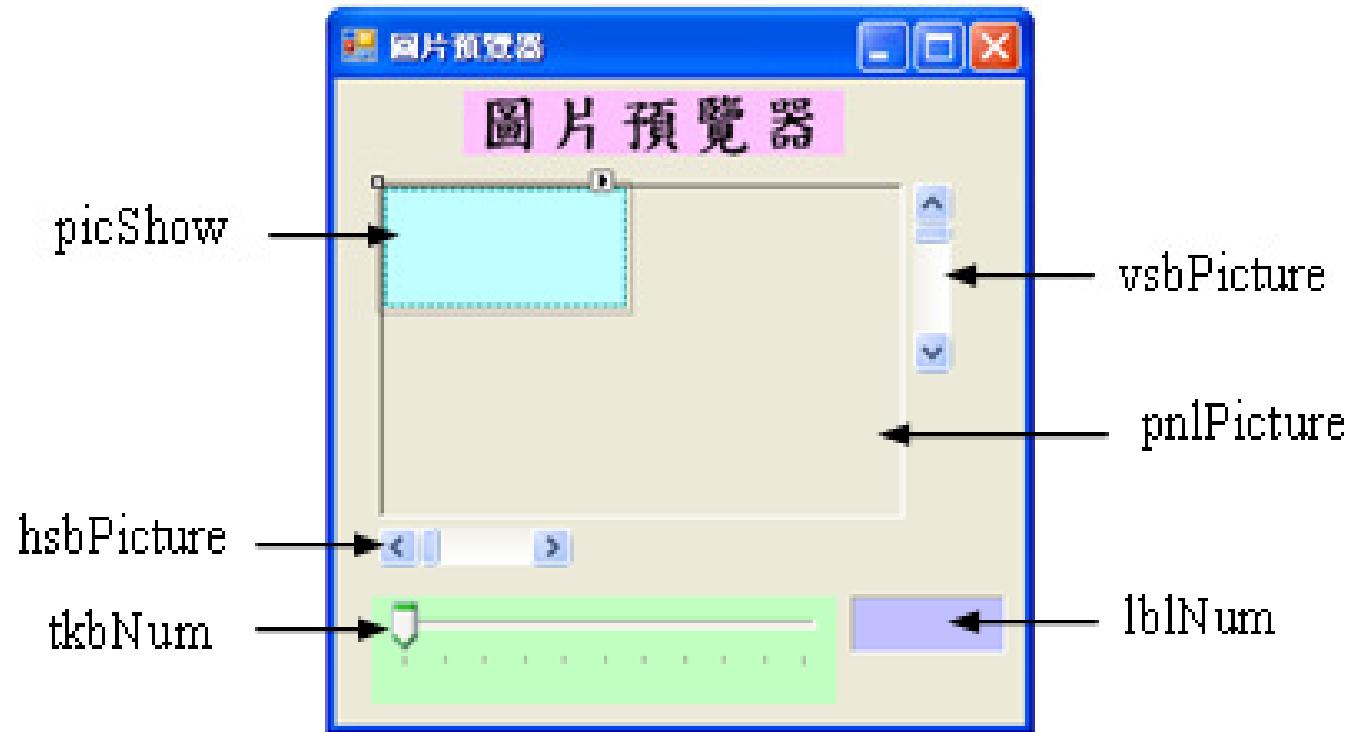
範例演練

設計一個簡單的圖片預覽器，在表單下方有一個滑動桿
拖曳滑動鈕可以切換三張不同的圖片，預設為顯示第一張。
切換圖片後，操作水平和垂直捲軸可以預覽圖片。
若圖片較小，則捲軸沒有作用。





■ Step1 設計輸出入介面





Step2 撰寫程式碼

```
// FileName : ScrollbarDemo.sln
```

```
01 private void Form1_Load(object sender, EventArgs e)
02 {
03     hsbPicture.Width = pnlPicture.Width;
04     vsbPicture.Height = pnlPicture.Height;
05     tkbNum.Minimum = 1;
06     tkbNum.Maximum = 3;
07     picShow.Image = Image.FromFile("picture1.jpg");
08     hsbPicture.Maximum = picShow.Width - pnlPicture.Width;
09     vsbPicture.Maximum = picShow.Height - pnlPicture.Height;
10     lblNum.Text = "第 1 張圖";
11 }
```



```
13 private void tkbNum_ValueChanged(object sender, EventArgs e)
14 {
15     picShow.Image = Image.FromFile("picture" + tkbNum.Value.ToString() + ".jpg");
16     picShow.Left = 0; picShow.Top = 0;
17     if (picShow.Width > pnlPicture.Width)
18     {
19         hsbPicture.Maximum = picShow.Width - pnlPicture.Width;
20         hsbPicture.Enabled = true;
21     }
22     else
23     {
24         hsbPicture.Enabled = false;
25     }
```



```
26 if (picShow.Height > pnlPicture.Height)
27 {
28     vsbPicture.Maximum = picShow.Height - pnlPicture.Height;
29     vsbPicture.Enabled = true;
30 }
31 else
32 {
33     vsbPicture.Enabled = false;
34 }
35 hsbPicture.Value = 0; vsbPicture.Value = 0;
36 lblNum.Text = "第 " + tkbNum.Value.ToString() + " 張圖";
37 }
```



39 **private void hsbPicture_Scroll(object sender, ScrollEventArgs e)**

40 {

41 picShow.Left = -hsbPicture.Value;

42 }

43

44 **private void vsbPicture_Scroll(object sender, ScrollEventArgs e)**

45 {

46 picShow.Top = -vsbPicture.Value;

47 }



10.5 RichTextBox 豐富文字方塊

- 該控制項提供類似Microsoft Word文書處理應用程式的顯示和文字管理功能。
- TextBox 無法處理具格式化的功能。
- 豐富文字方塊提供格式化功能：
 - 選取變更文字的格式
 - 選取變更文字的前景色與背景色、
 - 調整段落格式、
 - 建立項目符號清單、
 - 執行連結、
 - 載入與儲存Rich Text Format (RTF) 或純文字檔。



■ 一. RichTextBox 常用屬性

屬性	說明
BackColor	取得或設定 RichTextBox 控制項的背景色。
BorderStyle	取得或設定 RichTextBox 控制項的框線樣式。
DetectUrls	取得或設定 RichTextBox 控制項有超連結文字的部份是否加上藍色底線。
Font	取得或設定 RichTextBox 控制項所使用的文字字型。
ForeColor	取得或設定 RichTextBox 控制項的前景色。
SelectedText	取得或設定 RichTextBox 控制項所選取的文字。
SelectionBackColor	取得或設定 RichTextBox 控制項中所選取文字的背景色。
SelectionBullet	取得或設定目前 RichTextBox 控制項的插入點或選取範圍是否套用項目符號樣式。
SelectionColor	取得或設定 RichTextBox 控制項中所選取文字的前景色。



SelectionFont	取得或設定 RichTextBox 控制項中所選取文字的字型。
SelectionIndent	取得或設定 RichTextBox 控制項開始選取的行縮排長度 (以 Pixel 像素為單位)。
SelectionLength	取得或設定 RichTextBox 控制項中選取的字元數目。



■ 二. RichTextBox 常用方法

方法	說明
Copy	將 RichTextBox 控制項中目前選取的範圍複製到「剪貼簿」。
Cut	將 RichTextBox 控制項中目前選取的範圍移至「剪貼簿」。
LoadFile	將指定的檔案內容載入 RichTextBox 控制項內。 常用有下列兩種寫法： ① 將 c:\data.txt 純文字檔載入到 richTextBox1 內，其寫法如下： <code>richTextBox1.LoadFile (“c:\\data.txt”,RichTextBoxStreamType.PlainText);</code> ② 將 c:\data.rtf 檔(Rich Text Format)載入到 richTextBox1 內，其寫法如下： <code>richTextBox1.LoadFile (“c:\\data.rtf”,RichTextBoxStreamType.RichText);</code>
Paste	將「剪貼簿」的內容貼到 RichTextText 控制項指內的位置。
SaveFile	將 RichTextBox 控制項的內容儲存至指定的檔案。常用有下列兩種寫法： ① richTextBox1 內的資料儲存至 c:\data.txt 純文檔，其寫法如下： <code>richTextBox1.SaveFile (“c:\\data.txt”,RichTextBoxStreamType.PlainText);</code> ② richTextBox1 內的資料儲存至 c:\data.rtf 檔，其寫法如下： <code>richTextBox1.SaveFile (“c:\\data.rtf”,RichTextBoxStreamType.RichText);</code>
SelectAll	選取 RichTextBox 控制項中的所有文字。



三. RichTextBox常用事件

- 若 RichTextBox 內的文字包含超連結：
 - 將 DetectUrls = true，使該控制項的文字有連結的部份顯示藍色加底線。
 - 透過 LinkClicked 事件來執行與超連結相關的工作
 - 當在 RichTextBox 控制項有連結部份按一下會觸發 LinkClicked 事件
 - 此時透過下面程式連結到指定的網站：

```
private void richTextBox1_LinkClicked(object sender, LinkClickedEventArgs e)
{
    System.Diagnostics.Process.Start(e.LinkText);
}
```



範例演練

■ 使用 **RichTextBox**、**Button**、**Label**、**ComboBox** 等控制項 製作簡易記事本，功能說明：

① 前景色 黑

此清單可用來設定黑、紅、綠、藍四種豐富文字方塊所選取的文字前景色。

② 背景色 白

此清單可用來設定白、紅、綠、藍四種豐富文字方塊所選取的文字背景色。

③ 大小 10

此清單可用來設定文字大小，其文字大小有8、10、12、14、16、18、20。



④ 字體 新細明體

此清單可用來設定文字字體，其字體有新細明體、
細明體、標楷體、Arial。

⑤ 存檔

按下此鈕將豐富文字方塊的內容儲存至與執行檔
相同路徑下的 MyFile.rtf 檔內。

⑥ 開檔

按下此鈕將與執行檔相同路徑下的 MyFile.rtf 檔的
內容全部載入到豐富文字方塊內。

⑦ 清除

按下此鈕將豐富文字方塊的內容全部清除。

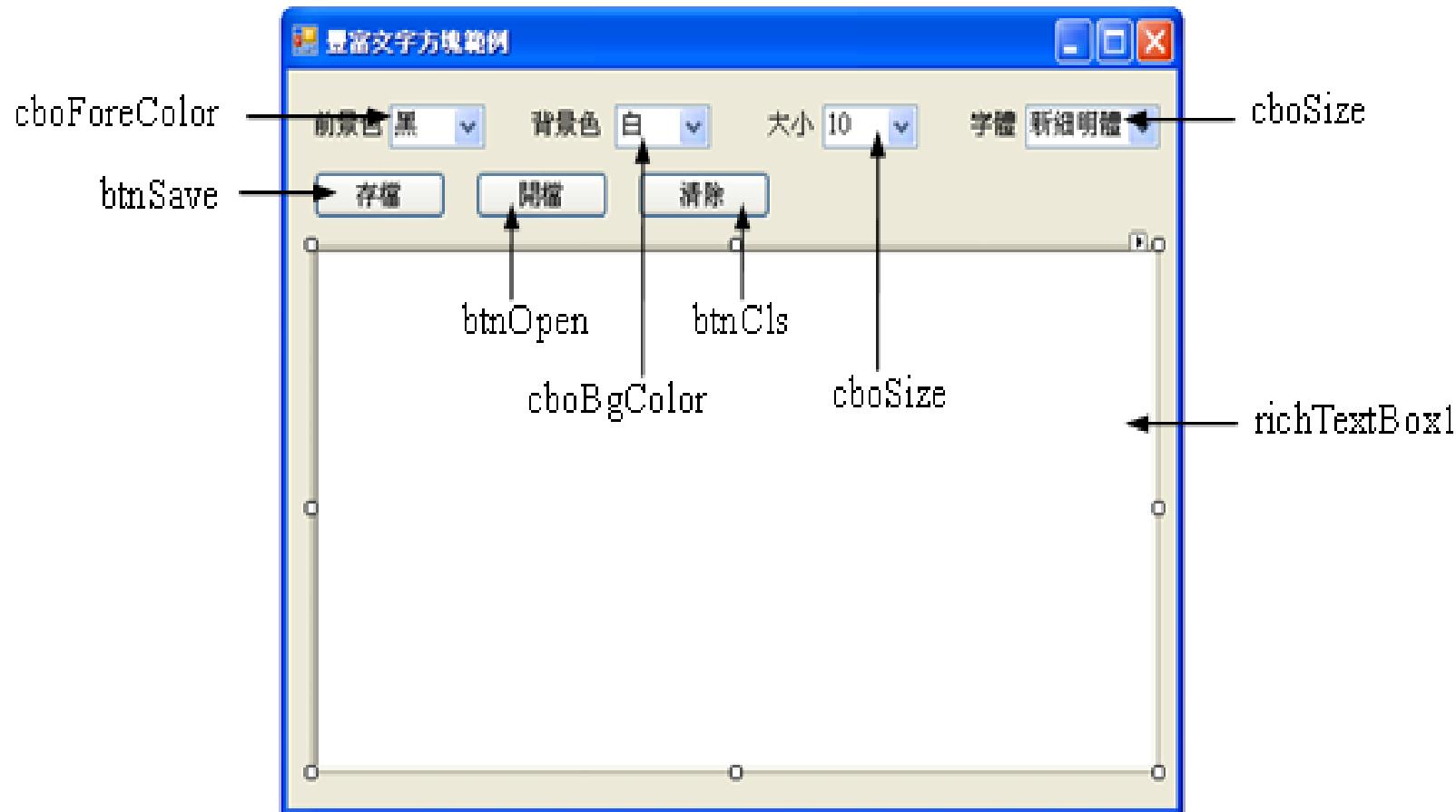


執行情形





■ Step1 設計輸出入介面





Step2 撰寫程式碼

```
// FileName : RichTextBoxDemo.sln
01 int FontSize ;
02 string FontName;
04 private void Form1_Load(object sender, EventArgs e)
05 {
06     FontSize = 10;
07     FontName = "新細明體";
08     richTextBox1.Font = new Font(FontName, FontSize);
09 }
11 private void cboForeColor_SelectedIndexChanged(object sender,
    EventArgs e)
12 {
13     if (cboForeColor.Text == "黑")
14     {
15         richTextBox1.SelectionColor = Color.Black;
16     }
```



```
17 else if (.cboForeColor.Text == "紅")
18 {
19     richTextBox1.SelectionColor = Color.Red;
20 }
21 else if (.cboForeColor.Text == "綠")
22 {
23     richTextBox1.SelectionColor = Color.Green;
24 }
25 else if (.cboForeColor.Text == "藍")
26 {
27     richTextBox1.SelectionColor = Color.Blue;
28 }
29 }
30
```



```
31 private void cboBgColor_SelectedIndexChanged(object sender, EventArgs e)
32 {
33     if (cboBgColor.Text == "白")
34     {
35         richTextBox1.SelectionBackColor = Color.White ;
36     }
37     else if (cboBgColor.Text == "紅")
38     {
39         richTextBox1.SelectionBackColor = Color.Red;
40     }
41     else if (cboBgColor.Text == "綠")
42     {
43         richTextBox1.SelectionBackColor = Color.Green;
44     }
45     else if (cboBgColor.Text == "藍")
46     {
47         richTextBox1.SelectionBackColor = Color.Blue;
48     }
49 }
```



```
51 private void cboSize_SelectedIndexChanged(object sender, EventArgs e)
52 {
53     FontSize = Convert.ToInt32(cboSize.Text);
54     richTextBox1.SelectionFont = new Font(FontName, FontSize);
55 }
56
57 private void cboFont_SelectedIndexChanged(object sender, EventArgs e)
58 {
59     FontName = cboFont.Text;
60     richTextBox1.SelectionFont = new Font(FontName, FontSize);
61 }
```



```
63 private void btnSave_Click(object sender, EventArgs e)
64 {
65     richTextBox1.SaveFile("MyFile.rtf", RichTextBoxStreamType.RichText );
66     MessageBox.Show("存檔成功!!!");
67 }
68
69 private void btnOpen_Click(object sender, EventArgs e)
70 {
71     richTextBox1.LoadFile("MyFile.rtf", RichTextBoxStreamType.RichText );
72     MessageBox.Show("開檔成功!!!");
73 }
74
75 private void btnCls_Click(object sender, EventArgs e)
76 {
77     richTextBox1.Text = "";
78 }
```