

1-5 ListBox 與 ComboBox

1-5-1 前言

在"1-4 RadioButton 與 CheckBox"節中已介紹選擇控制項最常見的 2 種，RadioButton(選項按鈕)與 CheckBox(查核方塊)，但如果希望將資料或選項以清單的方式呈現的話，則 ListBox(清單方塊)與 ComboBox(下拉式清單方塊)就是最佳的選擇。兩者的使用者介面如下圖 1-5-1。從圖中可發現兩者最大不同在於清單項目的呈現方式，ListBox 中會一次呈現多個項目，若清單項目超過 ListBox 的高度，則會輔以垂直捲軸讓使用者上下拉動來顯示其他的清單項目。至於 ComboBox 則必須點選右端箭頭方會跳出一個下拉清單顯示清單項目。雖然使用者介面有相當大的差異，但兩者所提供的屬性、事件、方法幾乎一模一樣。



圖 1-5-1 ListBox 與 ComboBox 使用者介面

1-5-2 學習目標

- 熟悉 ListBox(清單方塊)類別常用的屬性、事件與方法。
- 熟悉 ComboBox(下拉式清單方塊)類別常用的屬性、事件與方法。

1-5-3 ListBox(清單方塊)

功能簡述

允許使用者在清單中進行單一或多重點選取的控制項。

建構函數

ListBox()

建立一個 ListBox 類別的新物件。例如 `ListBox LST = new ListBox();` 會產生一個名為 LST 的 ListBox 物件。

常用屬性

SelectionMode	SelectionMode 列舉型別。取得或設定 ListBox 的選取模式。共有 4 種選項，說明如下： 1.None：沒有可選取項目，此時 SelectedIndex 屬性永遠為-1。 2.One：只能選取一個項目。預設為 One。 3.MultiSimple：可以選取多重項目，按一下滑鼠或按下空白鍵可選取或取消選取清單中的項目。 4.MultiExtended：可以選取多重項目，且使用者可以用 SHIFT、CTRL 和方向鍵進行選取。
Items	ListBox.ObjectCollection 類別，請參閱下面"四、ListBox.ObjectCollection"說明。此一屬性可讓您取得對於目前儲存在 ListBox 中之項目清單的參考。有了這個參考，您可以加入項目、移除項目和取得集合中的項目計數。由於此屬性是一個類別，所以有自己的屬性與方法，使用語法是： 物件名稱.Items.屬性(或方法)。 例如想知道 ListBox1 清單中共有多少項目，這時必須讀取 Items 中的 Count 的屬性，所以指令為： ListBox1.Items.Count 細節請參閱下面"四、ListBox. ObjectCollection "的部份。
SelectedIndex SelectedItem	若 ListBox 是單選，則 SelectedIndex 此屬性會傳回或設定被選取的項目之索引值，請注意索引值從 0 開始，也就是第 1 項(最上面)的索引值為 0，若未選取任何選項其值為-1。SelectedItem 則會傳回被選取的項目的內容。
SelectedIndices SelectedItems	若 ListBox 可複選，則 SelectedIndices 此屬性包含 ListBox 中所有目前選取項目之以零起始的索引。SelectedItems 此屬性則含有 ListBox 中目前所有選取的項目。下面的程式可將所有選取項目的索引及內容顯示出來。 For (int i = 0; i <= ListBox3.SelectedItems.Count - 1 ; i++) { MessageBox.Show(ListBox3.SelectedIndices(i)); MessageBox.Show(ListBox3.SelectedItems(i)); }
Sorted	ListBox 中的項目是否要排序，也就是俗稱的字典順序，所謂字典順序請參閱下面一段文字： Lexicographic order is sometimes called dictionary order. When applied to permutations, lexicographic order is increasing numerical order (or equivalently, alphabetic order for lists of symbols). For example, the permutations of {1,2,3} in lexicographic order are 123, 132, 213, 231, 312, and 321. When applied to subsets, two subsets are ordered by their smallest elements. For example, the subsets of {1,2,3}

	<p>in lexicographic order are {}, {1}, {1,2}, {1,2,3}, {1,3}, {2}, {2,3}, {3}.</p> <p>來源：http://mathworld.wolfram.com/LexicographicOrder.html</p>
HorizontalBar	是否要顯示橫式捲軸
MultiColumn	當 ListBox 的高度不足以顯示所有項目時，是否要以多欄的方式來顯示，例如在 ListBox 中加入下面 5 個選項："AAA"、"BBB"、"CCC"、"DDD"、"EEE"，若 MultiColumn 為 False 則顯示圖 1-5-2；若 MultiColumn 為 True 則顯示如圖 1-5-3。



圖 1-5-2 MultiColumn 為 False

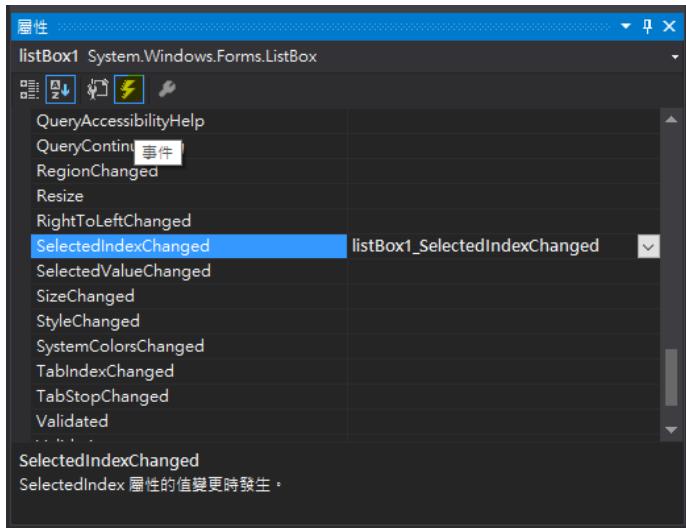


圖 1-5-3 MultiColumn 為 True

常用事件

SelectedIndexChanged	當 ListBox 中被選取的項目有改變時所觸發的事件。
----------------------	------------------------------

可由屬性視窗中事件按鈕選取事件名稱(如下圖)快點兩下快速產生事件程式碼。



1-5-4 ListBox.ObjectCollection

功能簡述

表示 ListBox 中的項目集合。Items 雖然是 ListBox 的一個屬性，但其實是另一種類別物件，類別名稱就是 `ListBox.ObjectCollection`。既然是一個類別當然就有自己的屬性與方法，例如 `ListBox1.Items.Count` 可得知 `ListBox1` 清單中共有多少項目、`ListBox1.Items.Add("莊利吉")`可在 `ListBox1` 的最後新增一個"莊利吉"選項。

建構函數

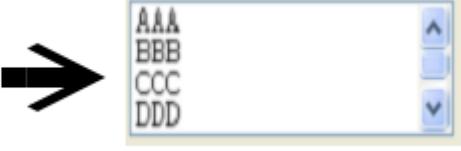
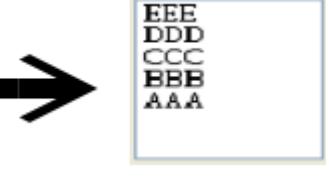
當產生 ListBox 物件時會自動建立 <code>ListBox.ObjectCollection</code> 物件，且掛到 <code>ListBox</code> 的 <code>Items</code> 屬性下。但若想自行產生可利用下面建構函數：

```
public ObjectCollection(
    ListBox owner
)
```

常用屬性

Count	Integer 型別。取得集合中的項目數目。
IsReadOnly	Boolean 型別。只能讀取，指出集合是否為唯讀。
Item	Object 類別。取得或設定 ListBox 中位於指定索引處的項目。例如要傳回 ListBox1 的第 1 個選項，可使用如下指令： ListBox1.Items.Item(0) 請注意 ListBox 中的索引值從 0 開始，也就是要傳回第 1 項則索引為 0。

常用方法

Clear	將 ListBox 中所有的項目清除。
Add Insert	此兩方法都是加入一個新項目到 ListBox 中，差別的是 Add 固定加入到 ListBox 中的最下端，而 Insert 却可指定插入的位置，例如 ListBox1.Items.Insert(0, "AAA");將一個新選項 AAA 加入到 ListBox1 的最上端，因為索引值 0 代表第一個。下圖 1-5-4 與圖 1-5-5 都是在 ListBox 中加入下面 5 個選項："AAA"、"BBB"、"CCC"、"DDD"、"EEE"，但圖 1-5-4 使用 Add 方法、圖 1-5-5 使用 Insert 指定插入在最前端。 <pre>ListBox1.Items.Clear() ListBox1.Items.Add("AAA") ListBox1.Items.Add("BBB") ListBox1.Items.Add("CCC") ListBox1.Items.Add("DDD") ListBox1.Items.Add("EEE")</pre> 
	
Remove RemoveAt	此兩方法都是從 ListBox 中移除某一項目，差別的是 Remove 方法要指定項目內容，例如 ListBox1.Items.Remove("AAA")。而 RemoveAt 要指定索引值，例如 ListBox2.Items.RemoveAt(0)會刪除 ListBox2 中的第一項。

補充說明

ListBox 中的項目除可使用上述的 Items.Add()、Items.RemoveAt()等方法加入刪除外，也可以在設計階段於 ListBox 控制項中 Items 屬性的地方，按一下右邊的流覽按鈕，就可出現選項的字串編輯器，直接於此輸入每一項的顯示文字，如下圖 1-5-6。

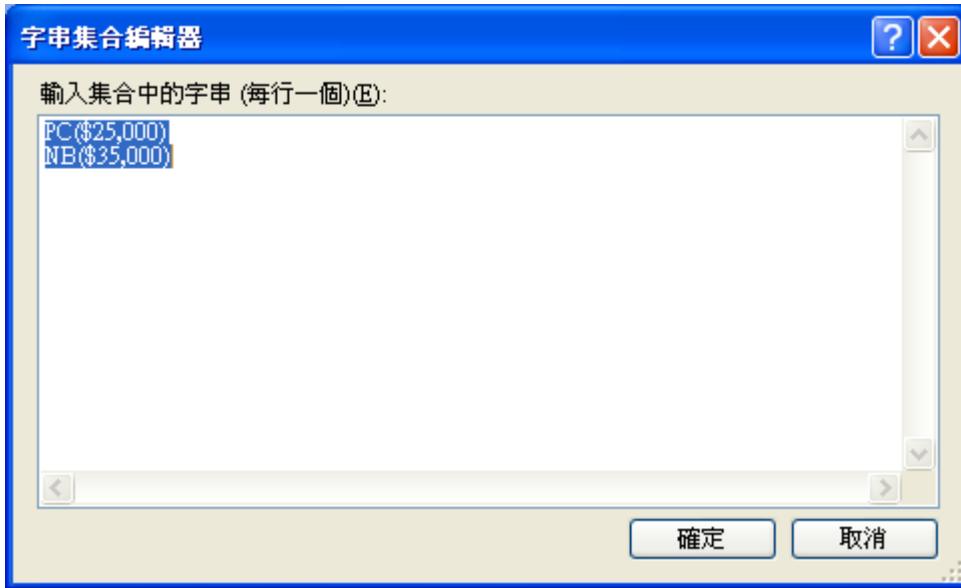


圖 1-5-6

1-5-5 ComboBox(下拉式清單方塊)

ComboBox 所提供的屬性、事件與方法與 ListBox 幾乎相同，但 ComboBox 本質上是 TextBox 與 ListBox 的結合，也就是說可以讓使用者如同 TextBox 般輸入選項的文字，也可以如 ListBox 般讓使用者只能以點選的方式來輸入。至於要以哪種型式來呈現，則看設計時將 DropDownListStyle 屬性設定成哪一種值。下面列出常用屬性、方法與事件，其中與 ListBox 相同者不再重複說明。

1.建構函數

ComboBox()	建立一個 ComboBox 類別的新物件。例如 ComboBox Cob = New ComboBox(); 會產生一個名為 Cob 的 ComboBox 物件。
------------	---

2.常用屬性

幾乎與 ListBox 相同，較有差異者如下：

DropDownStyle	ComboBoxStyle 列舉型別。決定選項的選取方式，有下列 3 種值： 1.Simple：相當於整數值 0。文字部分為可編輯的，清單部分是永遠可見的。使用者可以直接從 TextBox 輸入，也可以從 ListBox 中點選，介面如下圖 1-5-7。
---------------	--



圖 1-5-7

2.DropDown：相當於整數值 1。會固定顯示文字方塊，但點選向下箭頭時才會顯示清單。使用者可以直接從 TextBox 輸入，也可以從下拉清單中點選，介面如下圖 1-5-8。這是預設樣式。



圖 1-5-8

3.DropDownList：相當於整數值 2。使用者介面與設定成 ComboBoxStyle.DropDown(整數值 1)相同，但使用者只能從下拉式選單中點選不能在 TextBox 輸入。

3. 常用事件

SelectedIndexChanged	當 ListBox 中被選取的項目有改變時所觸發的事件。
----------------------	------------------------------

1-5-6 範例設計與說明

設計一個購物結帳模擬的程式，功能與"1-4RadioButton 與 CheckBox"的範例相同，但使

用 ListBox 與 ComboBox 來作為電腦與贈品選項，說明如下。

1. 執行畫面



圖 1-5-9

2. 物件名稱說明



圖 1-5-10

3. 演算法說明

- (1) 將二種電腦圖片與三種贈品圖片加入至專案成為資源
 - (2) 要調整圖片配合 PictureBox 大小，所以兩個 PictureBox：PB1 與 PB2 的 SixeMode 屬性設為 StretchImage。ListBox1 選取模式為預設，即每次只能選一樣。ComboBox1 設定成只能下拉選曲，不能以文字輸入。
 - (3) 在 Load 事件加入 "ASUS PC \$25,000"、"ACER NB \$35,000" 兩個選項到 ListBox1 中，並預設 "ASUS PC \$25,000" 為選取。加入 "影碟"、"公仔"、"背包" 三個選項到 ComboBox1 中，並預設 "影碟" 為選取。
 - (4) 設計一個程序 CalcAmount() 用來計算應付金額顯示於 Amount 中
 - (5) 在 ListBox1 的 SelectedIndexChanged 事件中檢查哪個選項被選取然後將對應圖片顯示於 PB1 中，然後呼叫 CalcAmount() 顯示應付金額。程式架構如下：
- ```
private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
 if (listBox1.SelectedIndex == 0)
 if (listBox1.SelectedIndex == 1)
 if (listBox1.SelectedIndex == 2)
}
```

```

else if (listBox1.SelectedIndex == 1)
CalcAmount();
}

```

- (6) 在 ComboBox1 的 SelectedIndexChanged 事件中檢查哪個選項被選取然後將對應圖片顯示於 PB2 中。

#### 4. 程式說明

```

namespace 購物結帳模擬3
{
 public partial class Form1 : Form
 {
 public Form1()
 {
 InitializeComponent();
 PB1.BorderStyle = BorderStyle.FixedSingle;
 PB1SizeMode = PictureBoxSizeMode.StretchImage;
 PB2.BorderStyle = BorderStyle.FixedSingle;
 PB2.SizeMode = PictureBoxSizeMode.StretchImage;
 listBox1.SelectionMode = SelectionMode.One;
 listBox1.Items.Clear();
 listBox1.Items.Add("ASUS PC $25,000");
 listBox1.Items.Add("ACER NB $35,000");
 listBox1.SelectedIndex = 0;
 comboBox1.DropDownStyle = ComboBoxStyle.DropDownList;
 comboBox1.Items.Add("影碟");
 comboBox1.Items.Add("公仔");
 comboBox1.Items.Add("背包");
 comboBox1.SelectedIndex = 0;
 CalcAmount();
 }

 private void CalcAmount()
 {
 if (listBox1.SelectedIndex == 0) Amount.Text = "25000";
 else if (listBox1.SelectedIndex == 1) Amount.Text = "35000";
 }

 private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
 {
 if (listBox1.SelectedIndex == 0) PB1.Image = Properties.Resources.ASUS_PC;
 else if (listBox1.SelectedIndex == 1) PB1.Image =
Properties.Resources.ACER_6291;
 CalcAmount();
 }

 private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
 {
 if (comboBox1.SelectedIndex == 0) PB2.Image = Properties.Resources.海潮之聲;
 else if (comboBox1.SelectedIndex == 1) PB2.Image = Properties.Resources.娃娃;
 else if (comboBox1.SelectedIndex == 2) PB2.Image = Properties.Resources.背包;
 }
 }
}

```

```
}
```

```
}
```

## 1-5-7 習題

修改上述範例在表單新增兩個 CheckBox："加購記憶體 1000 元"、"刷卡 5% 手續費"，顯示應付金額時須無條件捨棄小數部份，執行畫面如下。



圖 1-5-11