

## 9-4-9 複選按鈕 (CheckBox)

### 使用時機

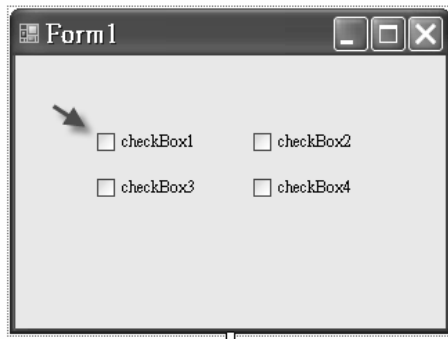
當需要在一組資料中選擇多項資料時。

### 常見屬性

屬性名稱	預設值	說明
Name	checkBox1	控制項物件名稱，最後數字是序號，同一控制項第二次出現就是 2
Autosize	true	依文字多寡及字型大小自動調整複選按鈕大小
BackColor	Window (白)	複選按鈕底色
Checked	False	是否被勾選
Font	新細明體, 9pt	顯示資料的字型
ForeColor	WindowText (黑)	顯示資料的顏色
Image	(無)	複選按鈕顯示的圖檔檔名
ImageAlign	MiddleCenter	圖檔顯示位置
Location		複選按鈕位置
Size		複選按鈕大小，Autosize 屬性值為 false 才有用
Text	CheckBox 1	複選按鈕顯示的資料
TextAlign	Left	複選按鈕顯示的資料選靠邊擇
Visible	True	是否顯示複選按鈕，可先設計好，由程式選擇顯示時機

## 常見事件

事件名稱	啟動時機
CheckedChanged	當複選按鈕被點選時



提示

複選按鈕必須多個一起使用，一次可點選多個，也就是點選一個時，原來點選的不會自動取消，要取消必須再點選一次。如果按鈕要預設為『點選』，checked 屬性值要設為 true，要知道何者被點選，同樣是檢查 checked 屬性值是否為 true。

範例

Hobbies 專案

設計一個男女學歷調查問卷，以文字標籤顯示問卷結果：

## 事件函式

按下按鈕時顯示問卷結果。

```

217         this->PerformLayout();
218     }
219 }
220 #pragma endregion
221 private System.Void button1_Click(System.Object^ sender, System.EventArgs^ e) {
222     String^ data="";
223     if (radioButton1->Checked==true)
224         data="男性";
225     else
226         data="女性";
227     if (checkBox1->Checked==true)
228         data+="國中";
229     if (checkBox2->Checked==true)
230         data+="高中";
231     if (checkBox3->Checked==true)
232         data+="大學";
233     if (checkBox4->Checked==true)
234         data+="研究所";
235
236     label3->Text=data+"畢業";
237 }
238 }
239 }
240
241

```

## 9-4-10 群組框 (GroupBox)



群組框可以將相關控制項加框美化，但重要的應用是：如果使用兩組以上單選按鈕時必須使用群組框來分組。



### 常見屬性

屬性名稱	預設值	說明
Name	groupBox1	控制項物件名稱，最後數字是序號，同一控制項第二次出現就是 2
Autosize	False	依文字多寡及字型大小自動調整群組框大小
BackColor	Control (灰)	群組框底色
BackgroundImage	(無)	群組框底圖
Font	新細明體, 9pt	群組名稱的字型
ForeColor	WindowText (黑)	群組名稱的顏色

屬性名稱	預設值	說明
Location		群組框位置
Size		群組框大小，Autosize 屬性值為 false 才有用
*Text	groupBox 1	群組框顯示的群組名稱
Visible	True	是否顯示群組框，可先設計好，由程式選擇顯示時機



群組框。



最常用的屬性是群組名稱 `Text`，用來標示該組資料的性質，如果不填，群組框就成為封閉的線框。本例加上群組框同時也給了群組名稱，原先的性別、學歷文字標籤就沒必要存在。

## 9-4-11 圖片框 (PictureBox)

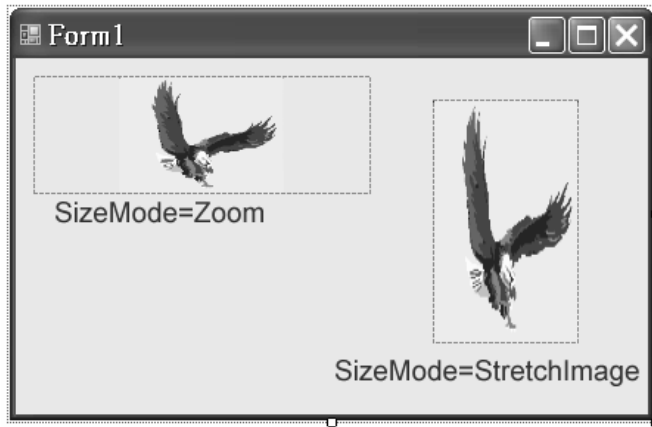


顯示圖檔資料時。



常見屬性

屬性名稱	預 設 值	說 明
Name	pictureBox1	控制項物件名稱，最後數字是序號，同一控制項第二次出現就是 2
BackColor	Control( 灰 )	圖片框底色
BackgroundImage	( 無 )	圖片框底圖
*BorderStyle	None	外框型式 None：無外框 Fixed3D：立體外框 FixedSingle：單線外框
Enabled	True	圖片框是否有效，可先設計好，由程式選擇有效時機
*Image	( 無 )	圖片檔檔名
Location		圖片框位置
Size		圖片框大小，Autosize 屬性值為 false 才有用
*SizeMode	Normal	圖片在圖片框內調整方式 Normal：圖片在圖片框左上角，不做調整 StretchImage：圖片調整與圖片框同大 AutoSize：圖片框調整與圖片同大 CenterImage：圖片置於圖片框中間 Zoom：圖片等比例放大至高或寬同圖片框高或寬
Visible	True	



Eagle 以圖片框顯示一隻老鷹圖片，配合計時器使用達成動畫效果。



★計時器事件每隔 1 秒啟動，每次根據 count 值換 pictureBox1 的圖片檔，由於 count 值必須連續，所以將 count 宣告成全域變數。

```
Eagle::Form1 Form1_Load(System::Object ^sender, System::Event
91 this->Controls->Add(this->pictureBox1);
92 this->Name = L"Form1";
93 this->Text = L"動畫";
94 this->Load += gcnew System::EventHandler(this, &Form1::Form_Load);
95 (cli::safe_cast<System::ComponentModel::ISupportInitialize ^>(this->pictureBox1))->EndInit();
96 this->ResumeLayout(false);
97
98 }
99 #pragma endregion
100 private: System::Void timer1_Tick(System::Object^ sender, System::EventArgs^ e) {
101     switch (count)
102     {
103     case 0:
104         pictureBox1->Image = Image::FromFile("e:\\c++\\code\\ch09\\Eagle\\eagle2.gif");
105         count = 1;
106         break;
107     case 1:
108         pictureBox1->Image = Image::FromFile("e:\\c++\\code\\ch09\\Eagle\\eagle3.gif");
109         count = 2;
110         break;
111     case 2:
112         pictureBox1->Image = Image::FromFile("e:\\c++\\code\\ch09\\Eagle\\eagle1.gif");
113         count = 0;
114         break;
115     }
116
117 }
118 private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e) {
119
```

★如果只換圖片只能見到老鷹只有動作沒有移動，因此計時器事件中加上圖片位置隨時間改變。

```
pictureBox1->Left += 12;
pictureBox1->Top += 2;
```

★如果要有一點立體感，老鷹由遠而近飛來，因此計時器事件中加上圖片大小隨時間改變：（要讓圖片大小能縮放，pictureBox 的SizeMode 屬性要設為 StretchImage。）

```
pictureBox1->Height += 3;
pictureBox1->Width += 4;
```

```

Eagle::Form1 timer1_Tick(System::Object^ sender, System
103
104 private: System::Void timer1_Tick(System::Object^ sender, System::EventArgs^ e) {
105     switch (count)
106     {
107         case 0:
108             pictureBox1->Image = Image::FromFile("e:\\c+\\code\\ch09\\Eagle\\ea
109             count = 1;
110             break;
111         case 1:
112             pictureBox1->Image = Image::FromFile("e:\\c+\\code\\ch09\\Eagle\\ea
113             count = 2;
114             break;
115         case 2:
116             pictureBox1->Image = Image::FromFile("e:\\c+\\code\\ch09\\Eagle\\ea
117             count = 0;
118             break;
119     }
120     pictureBox1->Left += 12;
121     pictureBox1->Top += 2;
122
123     pictureBox1->Height += 3;
124     pictureBox1->Width += 4;
125 }
126
127 private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e) {
128
129
130
131

```

## 9-4-12 選項清單框 (ListBox)



當輸入資料有一定的選項時可使用選項框，避免輸入資料不合規定。



常見屬性

屬性名稱	預設值	說明
Name	listBoxBox1	控制項物件名稱，最後數字是序號，同一控制項第二次出現就是 2
BackColor	Window (白)	選項清單底色
BorderStyle	Fixed3D	外框式樣 None：沒有外框 Fixed3D：立體外框 FixedSingle：單線外框
Font	新細明體, 9pt	顯示的字型
ForeColor	WindowText (黑)	字型的顏色



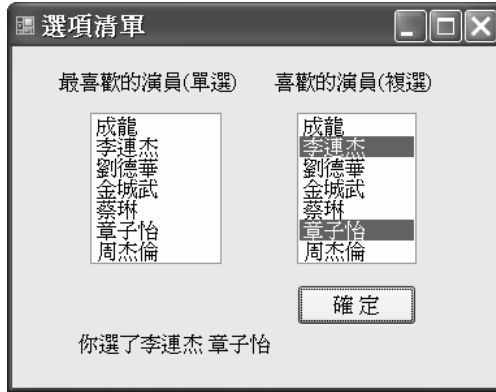
屬性名稱	預 設 值	說 明
ItemHeight	12	選項個數
Items	( 集合 )	定義選項清單內的選項
Location		選項清單位置
*SelectionMode	one	選取方式... <b>One</b> : 以滑鼠單選 <b>MultiSimple</b> : 以滑鼠一次一項多項複選 <b>MultiExtended</b> : 以滑鼠配合控制按鍵多項複選，按 <b>Shift</b> 鍵點選頭尾選項，按 <b>Ctrl</b> 鍵跳項連續點選多項。
Size		選項清單大小
Sorted	False	選項是否需要排序
Visible	True	是否顯示選項清單，可先設計好，由程式選擇顯示時機

## 常見事件

事件名稱	啟動時機
SelectedIndexChanged	當選項清單選項被點選時



選項清單中選擇喜歡的演員，左邊清單為單選，右邊清單可複選。



單選清單( SelectionMode 屬性值設為 one )可以使用 SelectedIndexChanged ( ) 函式檢查。複選清單 ( SelectionMode 屬性值設為 MultiSimple 或 MultiExtended ) 由於點選多次，一般以按鈕事件取出被選取的項目。

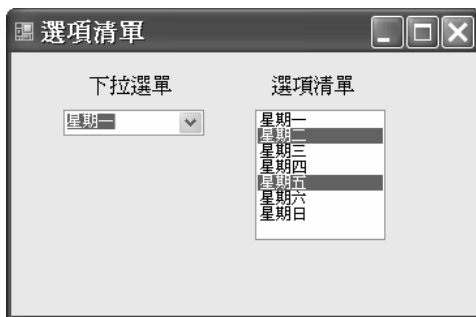
```

162 #pragma endregion
163
164 // 複選
165 private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
166     String^ data="";
167     // Loop 所有的選項
168     for (int i=0; i<listBox2->Items->Count; i++)
169         // 判斷是否被點選
170         if (listBox2->GetSelected(i))
171             // 取得點選項目的名稱
172             data+=listBox2->Items[i]->ToString()+" ";
173     label3->Text="你選了"+data;
174 }
175
176 // 單選
177 private: System::Void listBox1_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e) {
178     String^ data="";
179     // 取得點選項目的名稱
180     data=listBox1->SelectedItem->ToString();
181     label3->Text="你選了"+data;
182 }
183
184 private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e) {
185     label3->Text="";
186 }
187 };
188
189
190

```

## 🔄 下拉選單與選項清單框比較

下拉選單比較不佔版面位置且只能單選，選項清單框雖然佔版面位置，但可以複選。所以，如果是單選的輸入可視版面編排選擇下拉選單或選項清單框，如是複選的輸入，就得選擇選項清單框了。



## 9-4-13 日期時間選擇 (DateTimePicker)



日期與時間的輸入幾乎是每一個應用程式會用到的，如果以一般文字輸入框設計又不加格式化與檢查是非常簡單，如果要設計成非常人性化又不會出錯的輸入物件可要吃掉非常多的程式碼。這種情況當然會有現成的控制項可用，那就是日期時間選擇 (dateTimePicker) 控制項。



### 常見屬性

屬性名稱	預設值	說明
Name	dateTimePicker1	控制項物件名稱，最後數字是序號，同一控制項第二次出現就是 2
CalendarFont	新細明體,9pt	月曆字體字型
CalendarForeColor	(黑)	月曆字體顏色
CalendarMonthBackground	(白)	月曆月份背景顏色
CalendarTitleBackColor	(藍)	月曆標題列背景顏色
CalendarTitleForeColor	(白)	月曆標題列字體顏色

屬性名稱	預 設 值	說 明
*Format	Long	顯示及輸入格式 Long- 長日期格式，如 April 16, 2008 Short- 短日期格式，如 16/04/2008 Time- 時間格式，如 6:10:34 PM
MinDate	01/01/1753	容許最小的日期輸入值
MaxDate	31/12/9998	容許最大的日期輸入值
Location		控制項位置
Size		控制項大小
ShowUpDown	False	上下移動小按鈕
*Value		顯示及輸入的資料
Visible	True	是否顯示控制項，可先設計好，由程式選擇顯示時機

### 常見事件

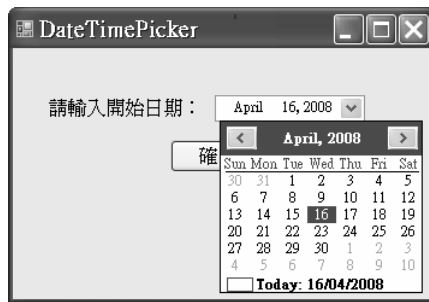
事件名稱	啟動時機
ValueChanged	當控制項被點選時

### 使用說明

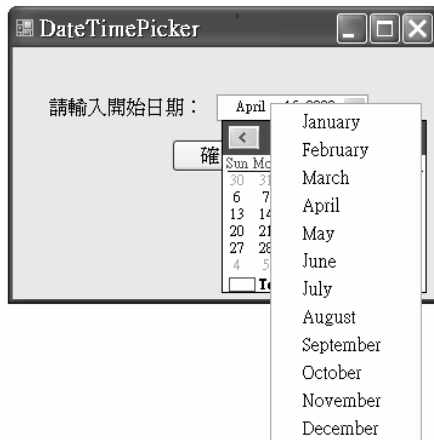
將日期時間選擇控制項於表單上拖放好位置，會出現日期的下拉選單，如下圖所示：



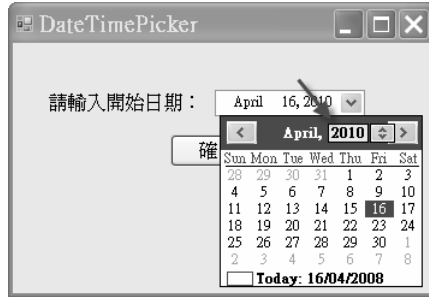
點擊日期的下拉選單會出現一個月曆般的日期輸入畫面，如下圖所示：



在月曆上翻頁或點擊月份可選擇月份、如下圖所示：



在月曆上點擊年份可點選年份、如下圖所示：



如果要輸入時間資料，可將日期時間選擇控制項的 `Format` 屬性值設定為 `Time`，並將 `ShowUpDown` 屬性值設定為 `True`，會出現時間的設定選單，如下圖所示：



**範例** DateTime

以日期時間選擇控制項輸入日期：



- ★計算輸入日期到年底的日數。
- ★計算輸入日期 100 天後的日期。

★計算年初到輸入日期的日數。

```
155 this->PerformLayout();
156
157 }
158 #pragma endregion
159
160 private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
161     // 取輸入的日期
162     DateTime date1 = dateTimePicker1->Value;
163
164     // 計算輸入日期到年底的日數
165     DateTime date2 = DateTime( date1.Year, 12, 31);
166     TimeSpan diff = date2.Subtract( date1 );
167     String^ s = date1.Year + "/" + date1.Month + "/" + date1.Day;
168     label2->Text = s + " 到年底有 "+diff.Days+"天";
169
170     // 計算輸入日期100天後的日期
171     DateTime date3 = date1.AddDays(100);
172     label3->Text = s + " 100天後是"
173         + date3.Year + "/" + date3.Month + "/" + date3.Day;
174
175     // 計算年初到輸入日期的日數
176     date2 = DateTime( date1.Year, 1, 1);
177     diff = date1.Subtract( date2 );
178     label4->Text = "年初到 "+s+"有 "+diff.Days+" 天";
179
180 }
181
182 private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e) {
183 }
```