豐富文字方塊與工具列

CHAPTER 13

13.1 RichTextBox 豐富文字方塊控制項

由於 TextBox 文字方塊工具無法處理具有格式的文字，此時就必須使用 RichTextBox 豐富文字方塊工具。RichTextBox 控制項是繼承至 TextBox 控制項，因此具有 TextBox 所有屬性和方法，還提供更進階的格式化功能，例如：字型和顏色的設定、存取 Rich Text 格式檔 (簡稱 RTF 檔)…等。使用 TextBox 控制項的應用程式只要稍加修改，就可以改用 RichTextBox 控制項。

RichTextBox 控制項提供一些屬性，可以將格式套用於控制項內選取的字元或段落。若要變更某部分字元或段落的格式，只要先選取該部分文字，就能套用指定格式，而且設定後輸入的文字都會沿用相同的格式，一直到變更設定或者選取控制項文件中的不同區段為止。另外，RichTextBox 控制項不像 TextBox 控制項受到 64K 字元容量的限制。下表是 RichTextBox 控制項常用的成員：

成員名稱	說　　明
AutoWordSelection 屬性 (預設值 False)	True：表示只要使用者在控制項內，有文字的任何部分快按兩下，就可以自動選取整個字(Word)。
SelectionBullet 屬性 (限執行階段使用)	是否要建立項目符號清單。如右圖所示。 richTextBox1.SelectionBullet = true; • 第一章 • 第二章 • 第三章
SelectionFont 屬性 (限執行階段使用)	可設定或取得選取文字的字型樣式(如：標準、粗體、斜體…等)，也可以設定字型大小和字型。

成員名稱	說　　明
SelectionColor 屬性 (限執行階段使用)	可設定或取得選取文字的色彩。
DetectUrls 屬性	如果 RichTextBox 控制項內的文字含有和網站相連的文字超連結時，可將屬性值設為 True, 會將控制項的超連結文字以藍色加底色的格式顯示。使用者點按超連結文字時，會觸發 LinkClicked 事件。
SelectionIndent 屬性 (限執行階段使用)	調整段落的格式，此屬性設定段落左邊縮排的距離。相當於 Word 尺規的左邊縮排尺標。(以 Pixel 為單位)
SelectionHangingIndent 屬性(限執行階段使用)	調整段落的格式，此屬性設定段落首行縮排的距離。相當於 Word 尺規的首行凸排尺標。(以 Pixel 為單位)
SelectionRightIndent 屬性 (限執行階段使用)	調整段落的格式，此屬性設定段落右邊縮排的距離。相當於 Word 尺規的右邊縮排尺標。(以 Pixel 為單位)
Copy 方法	將控制項中選取的文字複製到「剪貼簿」中。
Cut 方法	將控制項中選取的文字剪下，並複製到「剪貼簿」中。
Paste 方法	將「剪貼簿」中的文字複製到控制項中。
LoadFile 方法	將現有的 RTF 或 ASCII 文字檔載入控制項。 [例] 將 ASCII 文字檔載入到 richTextBox1 內： richTextBox1.LoadFile("C:\\My\\Testdoc.txt", RichTextBoxStreamType.PlainText) ; [例]將 RTF 文字檔載入到 richTextBox1 內： richTextBox1.LoadFile("C:\\My\\Testdoc.rtf", RichTextBoxStreamType.RichText);
SaveFile 方法	可將控制項中文字與格式，以指定檔案儲存為 RTF 或 ASCII 文字格式，其用法和 LoadFile()方法類似。
Find 方法	用來在該控制項的文字內尋找指定的字串。若有找到，傳回找到字串第一個字元的索引位置，並反白顯示找到的文字；若找不到傳回 -1。
Clear 方法	將控制項內的文字全部刪除。
LinkClicked 事件	用來執行和連結相關聯的工作。 [例] 連結到指定的網站或網頁： System.Diagnostics.Process.Start(e.LinkText) ;
TextChagned 事件	當該控制項的內容有任何改變都會觸發此事件，將此事件內的程式區段執行一次。

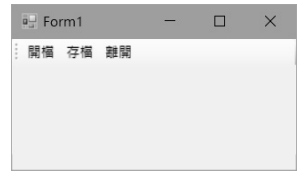
13.2 ToolStrip 工具列控制項

在視窗應用程式環境下，會將常用的功能以按鈕圖示組合成一個工具列，掛在功能表的下方以方便快速選取，譬如 Visual Studio 整合開發環境(IDE)的標準工具列就是如此。透過工具箱中的 ToolStrip 工具便可輕易做出自訂工具列，ToolStrip 控制項預設放在表單的標題欄正下方，可以在此工具列設定圖示及文字。

一. ToolStrip 工具列容器內功能項目常用成員

ToolStrip 工具列控制項是一個容器，是由一些按鈕、標籤、下拉式按鈕、文字方塊所組合而成的集合。執行階段若要在工具列新增選項，可透過 Items.Add()方法來達成。譬如欲在 toolStrip1 工具列控制項後面新增三個文字按鈕，其寫法如下：

```
toolStrip1.Items.Add("開檔");
toolStrip1.Items.Add("存檔");
toolStrip1.Items.Add("離開");
```



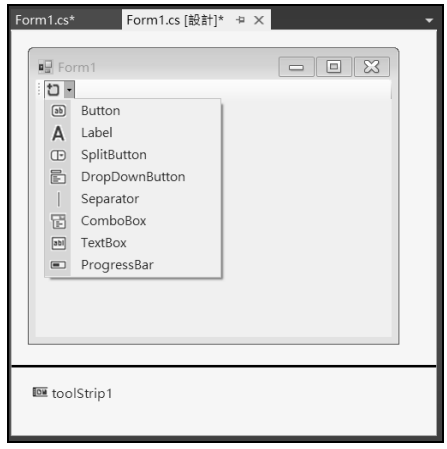
ToolStrip 工具列容器中的控制項，例如按鈕、標籤、下拉式清單…等控制項，其常用成員說明如下：

成員名稱	說明
Alignment 屬性	設定工具圖示由左或右開始排起。預設值：Left。
AutoToolTip 屬性	決定是否使用 Text 或 ToolTipText 當工具列的提示。
Checked 屬性	表示 ToolStripButton 是已按下或未按下。
Enabled 屬性	設定該工具圖示是否有效，預設值為 True。

成員名稱	說明
DisplayStyle 屬性	① None:空白圖示。 ② Text：只顯示文字當圖示。 ③ Image：只顯示影像當圖示(預設值)。 ④ ImageAndText：同時顯示文字與圖像當圖示。
Image 屬性	當 DisplayStyle 屬性有 Image 屬性設定時，Image 屬性才有效。
ImageScaling 屬性	用來設定所選取的圖像是否要調整和 ToolStrip 一樣大小。預設值為 SizeToFit 會縮成和 ToolStrip 一樣大小。None：不做調整。
ImageTransparentColor 屬性	將指定的顏色設成透明。
RightToLeft 屬性	用來表示是否使用 RTL 語言方式由右至左繪製。 ① 預設值為 No，表示按照預設方式先圖後文字。 ② 若設為 Yes，表示先文字後圖。
Text 屬性	若 DisplayStyle 屬性有 Text 設定時，此屬性才有效。
TextDirection 屬性	用來設定文字顯示的方向： ① Horizontal ： (預設值) ② Vertical 90： ③ Vertical 270：
TextImageRelation 屬性	用來設定圖片相對於文字的相對位置： ① ：ImageBeforeText(預設值) ② ：TextBeforeImage ③ ：TextAboveImage ④ ：TextAboveImage
ToolTipText 屬性	設定當滑鼠移到該工具圖示上時欲顯示的提示訊息。

二. ToolStrip 工具列容器內控制項的建立

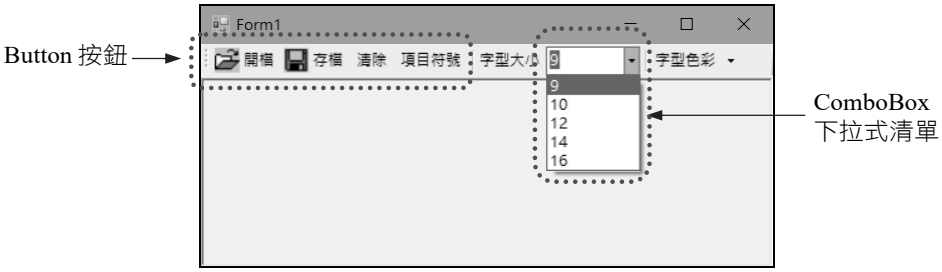
當在工具箱的 ToolStrip 工具上快按兩下，會在表單標題欄的正下方產生一個空白的工具列，如下圖所示，ToolStrip 工具列內有些控制項可供選用。



【工具選項說明】

工具選項	說明
Button 按鈕工具圖示	產生按鈕控制項，按鈕上可有影像、文字、影像和文字並存。
Label 標籤工具圖示	產生標籤控制項，標籤上可有影像、文字、影像和文字並存。
SplitButton 分隔鈕工具圖示	產生標準按鈕和下拉按鈕的組合。
DropDownButton 下拉按鈕工具圖示	產生按鈕的下拉選項，可指定 DropDownItem 屬性值來顯示清單項目。
Seperator 分隔線	產生分隔線，用來將同性質工具加以區隔。
ComboBox 下拉式清單	產生下拉式清單控制項。
TextBox 文字方塊	產生文字方塊控制項。
ProgressBar 進度棒	產生進度棒控制項。

接著練習如何在工具列上建立按鈕、標籤、下拉式清單、分隔鈕…等控制項。

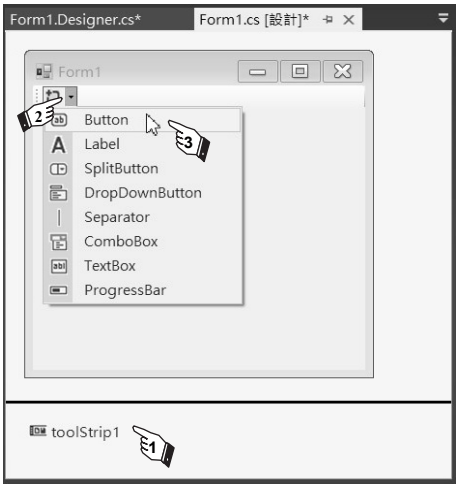


操作步驟

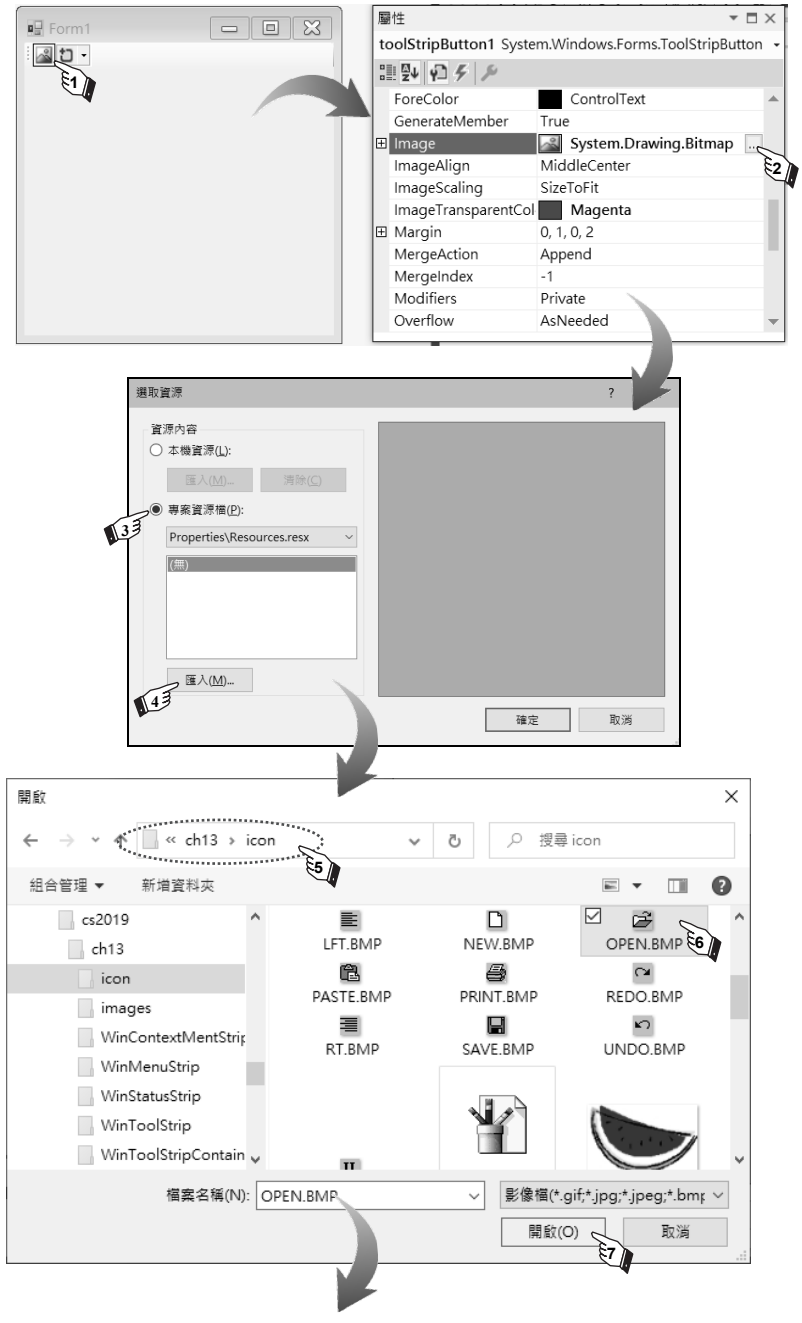
Step 01 在工具箱的 ToolStrip 工具上快按兩下，加入物件名稱(Name 屬性)為 toolStrip1 工具列控制項。

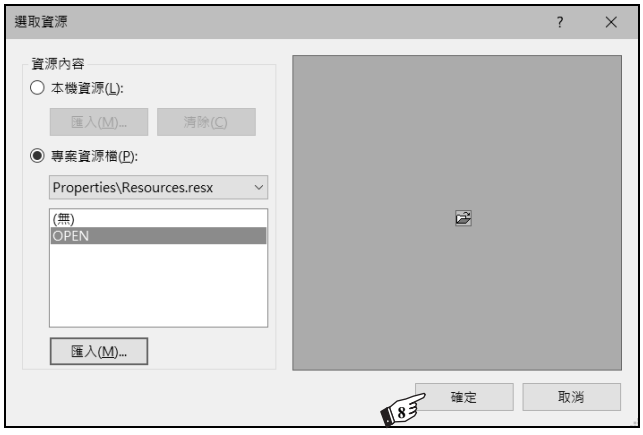
Step 02 在 toolStrip1 工具列內新增含有影像和文字的 開檔 按鈕

① 先選取 toolStrip1 的 下拉鈕，接著再由清單中選取「Button」，表示使用按鈕控制項當工具列的第一個圖示按鈕。

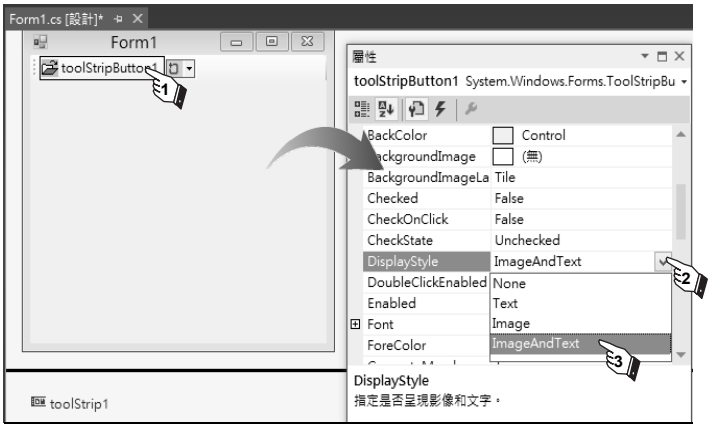


② 依照下圖數字順序操作，將 toolStrip1 工具列的第一個按鈕圖示改成 圖示。(OPEN.BMP 置於書附範例 ch13/Icon 資料夾下)

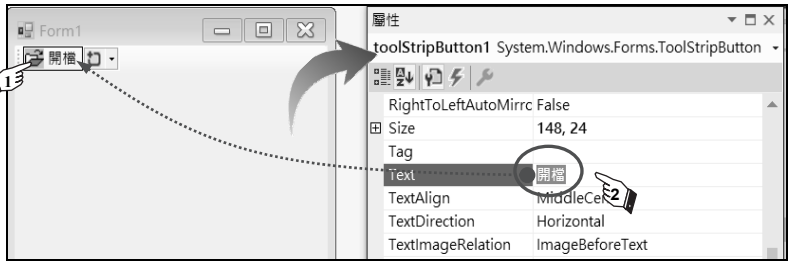




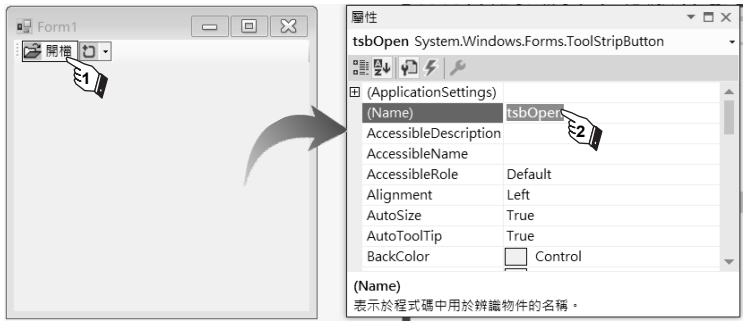
③ 在下圖選取 toolStrip1 中的 按鈕，透過屬性視窗將此按鈕控制項的 DisplayStyle 屬性設為「ImageAndText」，讓該按鈕上面可同時顯示文字及圖示。



④ 在下圖選取 toolStrip1 中的 按鈕，透過屬性視窗將此按鈕控制項的 Text 屬性設為「開檔」，讓該按鈕以 開檔 顯示。



⑤ 選取 toolStrip1 中的 開檔 按鈕，透過屬性視窗將此按鈕控制項的 Name 屬性(物件名稱)更名為「tsbOpen」。



Step 03 重複 Step 02，在 toolStrip1 工具列內新增含有影像和文字的 存檔 按鈕

① 在 toolStrip1 新增「Button」按鈕控制項當工具列的第二個圖示按鈕。

② 將 toolStrip1 工具列的第二個按鈕圖示改成 圖示(SAVE.BMP)。

③ 將 toolStrip1 中 按鈕的 DisplayStyle 屬性設為「ImageAndText」，讓該按鈕上面可同時顯示文字及圖示。

④ 將 toolStrip1 中 按鈕的 Text 屬性設為 "存檔"，讓該按鈕以 存檔 顯示。

⑤ 將 toolStrip1 中的 存檔 按鈕的 Name 屬性(物件名稱)更名為「tsbSave」。

Step 04 重複 Step 02，在 toolStrip1 工具列內新增「清除」文字按鈕

① 在 toolStrip1 新增「Button」按鈕控制項當工具列的第三個按鈕。

② 選取 toolStrip1 的第三個按鈕，將該按鈕的 DisplayStyle 屬性設為「Text」，使該按鈕以文字顯示。

③ 選取 toolStrip1 的第三個按鈕，將該按鈕的 Text 屬性設為 "清除"。

④ 選取 toolStrip1 的第三個按鈕，將該按鈕的 Name 屬性(物件名稱)設為「tsbCls」。

Step 05 重複 Step 02，在 toolStrip1 工具列內新增「項目符號」文字按鈕

① 在 toolStrip1 新增「Button」按鈕控制項當工具列的第四個按鈕。

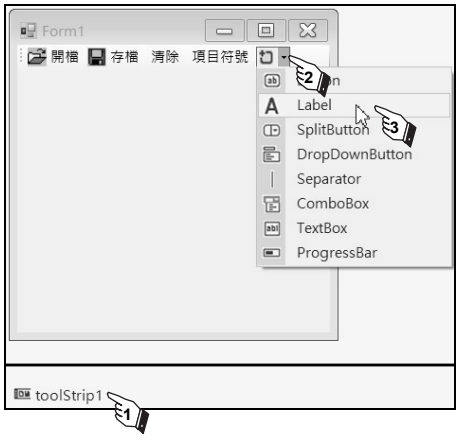
② 選取 toolStrip1 的第四個按鈕，將該按鈕的 DisplayStyle 屬性設為「Text」，使該按鈕以文字顯示。

③ 選取 toolStrip1 的第四個按鈕，將該按鈕的 Text 屬性設為 "項目符號"。

④ 選取 toolStrip1 的第四個按鈕，將該按鈕的 Name 屬性(物件名稱)設為「tsbBullet」。

Step 06 在 toolStrip1 工具列內新增「字型大小」標籤。

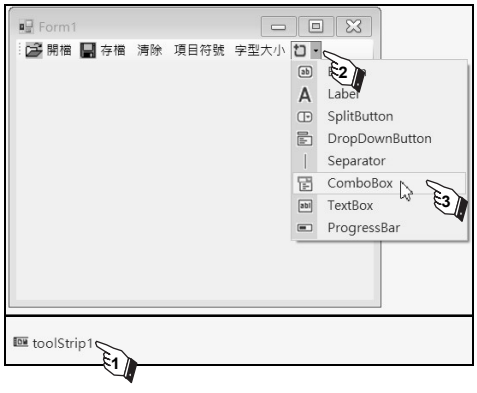
① 先選取 toolStrip1 的 下拉鈕，接著再由清單中選取「Label」，表示在 toolStrip1 內加入一個標籤控制項。



② 將此標籤的 Text 屬性設為 "字型大小"。

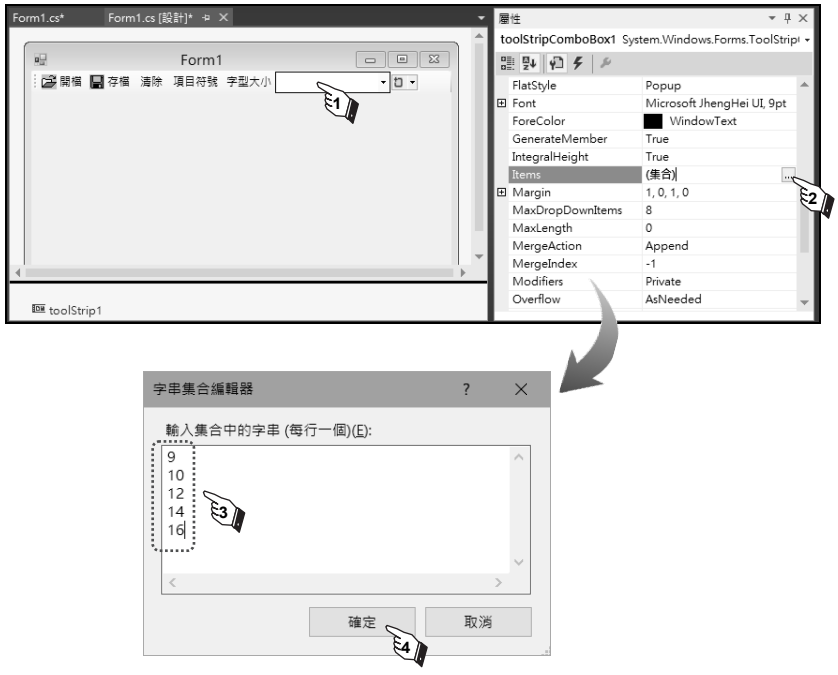
Step 07 在 toolStrip1 工具列內新增下拉式清單，清單項目有 9、10、12、14、16。

① 先選取 toolStrip1 的 下拉鈕，接著再由清單中選取「ComboBox」，表示在 toolStrip1 工具列加入下拉式清單。



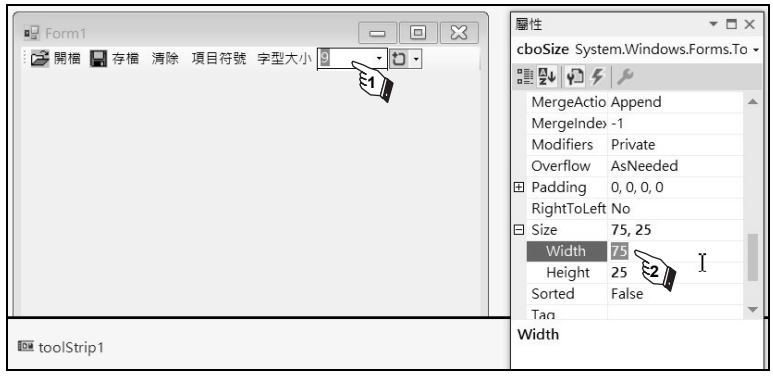
② 選取 toolStrip1 的下拉式清單，將該清單的 Text 屬性設為 "9"，表示該下拉式清單預設項目為「9」。

③ 選取 toolStrip1 的下拉式清單，然後按下下拉式清單 Items 屬性的 ... 鈕開啟「字串集合編輯器」視窗，然後依序建立下拉式清單的項目有 9、10、12、14、16。



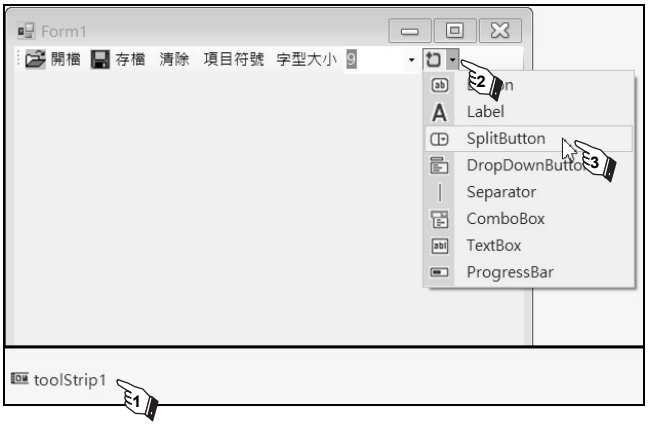
④ 選取 toolStrip1 中的下拉式清單，透過屬性視窗將此下拉式清單控制項的 Name 屬性(物件名稱)更名為「cboSize」。

⑤ 如下圖操作，若 toolStrip1 的下拉式清單太寬，可將該控制項 Size 子屬性 Width 設為「75」。



Step 08 在 toolStrip1 工具列內新增 SplitButton 分隔鈕。

① 如下圖，先選取 toolStrip1 的 下拉鈕，接著再由清單中選取「SplitButton」，表示在 toolStrip1 內加入 SplitButton 分隔鈕。



② 選取 toolStrip1 中的 SplitButton，將此控制項的 Name 屬性(物件名稱)更名為「tsbFontColor」。

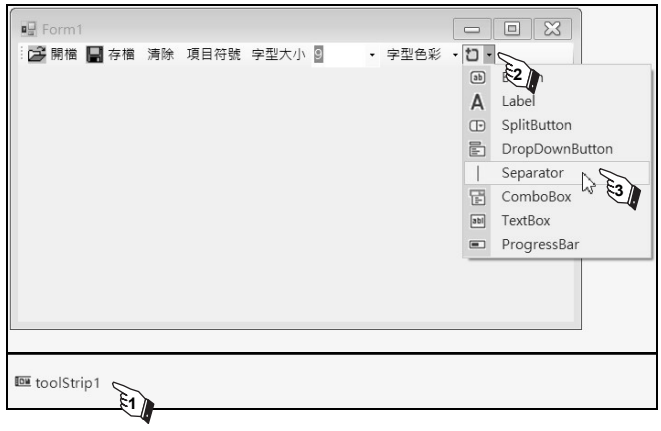
③ 選取 toolStrip1 中的 SplitButton(即 tsbFontColor 分隔鈕)，將此控制項的 DisplayStyle 屬性設為「Text」，讓該按鈕上面可同時顯示文字。

④ 選取 toolStrip1 中的 SplitButton(即 tsbFontColor 分隔鈕)，將此控制項的 Text 屬性設為 "字型色彩"。

⑤ 選取 toolStrip1 中的 "字型色彩" 分隔按鈕，接著依序在 在這裡輸入 上輸入「黑」、「紅」、「綠」、「藍」四個功能項目。

Step 09 在 toolStrip1 工具列內新增分隔圖示。

① 如下圖，先選取 toolStrip1 的 下拉鈕，接著再由清單中選取「Separator」，即在 toolStrip1 內加入 Separator 分隔圖示。



② 重複上述動作，在 toolStrip1 內再新增一個 Separator 分隔圖示。

③ 使用滑鼠分別將分隔圖示，拖曳到「字型大小」和「字型色彩」圖示的左邊，如下圖指定的位置。

範例：WinToolStrip.sln

使用豐富文字方塊與本節所建立的工具列，加上下列功能設計一個簡易的文書編輯程式。各工具按鈕功能說明如下：

① 開檔：將 GOTOP.rtf 檔的內容載入到豐富文字方塊內。(將書附檔案 GOTOP.rtf 檔複製到目前專案的 bin\Debug 資料夾下)

② 存檔：將豐富文字方塊的內容寫回 GOTOP.rft 檔內。

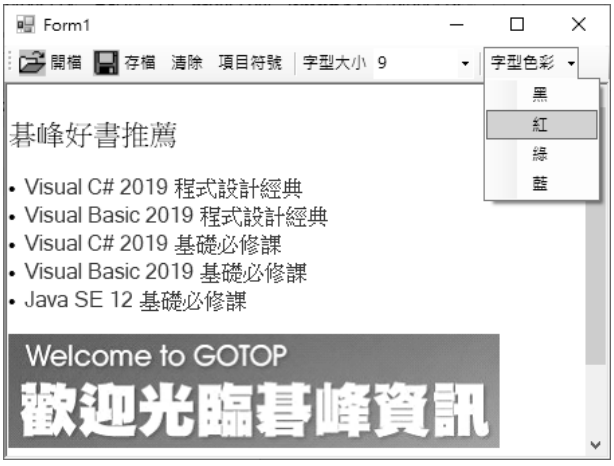
③ 清除：將豐富文字方塊的內容全部清除。

④ 項目符號：將選取的文字建立項目符號清單。

⑤ 字型大小下拉式清單：依下拉式清單設定選取文字的字型大小。

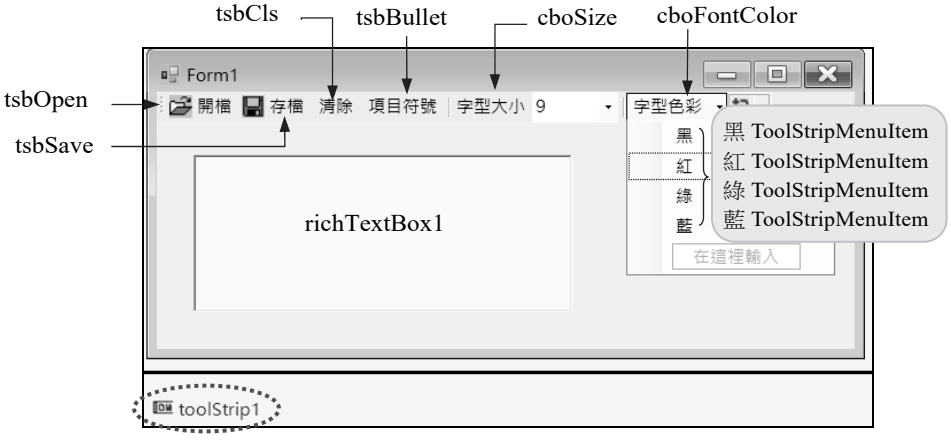
⑥ 字型色彩分隔鈕：設定選取文字的前景色彩。

執行結果



操作步驟

Step **01** 設計表單的輸出入介面：



Step **02** 撰寫程式碼

程式碼 FileName:Form1.cs

```
namespace WinToolStrip
{
    public partial class Form1 : Form
    {
        public Form1()
        {
```

```
07            InitializeComponent();
08        }
09        // ===  表單載入時執行
10        private void Form1_Load(object sender, EventArgs e)
11        {
12            // 豐富文字方塊填滿整個表單
13            richTextBox1.Dock = DockStyle.Fill;
14        }
15        // === 按工具列的 <開檔> 鈕時執行
16        private void tsbOpen_Click(object sender, EventArgs e)
17        {
18            // 使用 try{...}catch{...}來補捉沒有檔案可能發生的例外
19            try
20            {
21                // 將 GOTOP.rtf 檔的內容載入到 richTextBox1 豐富文字方塊內
22                richTextBox1.LoadFile("GOTOP.rtf",
                                RichTextBoxStreamType.RichText);
23            }
24            catch (Exception ex)
25            {
26                MessageBox.Show(ex.Message);
27            }
28        }
29        // ===  按工具列的 <存檔> 鈕時執行
30        private void tsbSave_Click(object sender, EventArgs e)
31        {
32            // 將 richTextBox1 豐富文字方塊內的資料儲存到 GOTOP.rtf 檔
33            richTextBox1.SaveFile("GOTOP.rtf",
                   RichTextBoxStreamType.RichText);
34        }
35        // ===  按工具列的 <清除> 鈕時執行
36        private void tsbCls_Click(object sender, EventArgs e)
37        {
38            richTextBox1.Clear();        //執行 Clear()方法
39        }
40        // ===  按下工具列的 <項目符號> 鈕時執行
41        private void tsbBullet_Click(object sender, EventArgs e)
42        {
```

```
43            richTextBox1.SelectionBullet = !richTextBox1.SelectionBullet;
44        }
45        // ===  字型下拉式清單 SelectedIndex 屬性值改變或選取下拉式清單時執行
46        private void cboSize_SelectedIndexChanged
              (object sender, EventArgs e)
47        {
48            // 設定選取字型的樣式
49            richTextBox1.SelectionFont =
                  new Font(richTextBox1.Font.FontFamily.ToString(),
                  float.Parse(cboSize.Text), richTextBox1.Font.Style);
50        }
51        // ===  按 [黑] 項目時執行
52        private void 黑 ToolStripMenuItem_Click(object sender, EventArgs e)
53        {
54            // richTextBox1 豐富文字方塊被選取部份字型色彩設為黑色
55            richTextBox1.SelectionColor = Color.Black;
56        }
57        // ===  按 [紅] 項目時執行
58        private void 紅 ToolStripMenuItem_Click(object sender, EventArgs e)
59        {
60            // richTextBox1 豐富文字方塊被選取部份字型色彩設為紅色
61            richTextBox1.SelectionColor = Color.Red;
62        }
63        // ===  按 [綠] 項目時執行
64        private void 綠 ToolStripMenuItem_Click(object sender, EventArgs e)
65        {
66            // richTextBox1 豐富文字方塊被選取部份字型色彩設為綠色
67            richTextBox1.SelectionColor = Color.Green ;
68        }
69        // ===  按 [藍] 項目時執行
70        private void 藍 ToolStripMenuItem_Click(object sender, EventArgs e)
71        {
72            // richTextBox1 豐富文字方塊被選取部份字型色彩設為藍色
73            richTextBox1.SelectionColor = Color.Blue;
74        }
75    }
76 }
```