

究竟什麼是多重文件介面，讓它們登台亮相之前，先解釋兩個名詞：

- ▶▶ 單一文件介面 (SDI, Single Document Interface)：SDI 表示一次只能開啟一份文件，例如使用「記事本」。
- ▶▶ 多重文件介面 (MDI, Multiple Document Interface)：能同時編輯多份文件，例如 MS Word。有多份文件時，還能決定它們是以「並排顯示」或「分割」，做不同的排列組合。

12-1 建立 MDI 表單

產生 SDI 文件只要在 Windows 系統下，可以出現於螢幕任何地方！如果是 MDI 文件就得有所考量，所有 MDI 文件只能在 MDI 父視窗的工作區域內顯示，接受 MDI 父視窗的管轄。舉個簡單例子，使用 VS 2019 軟體時，關閉某個專案，執行環境（父視窗）並不會關閉。由 MDI 父視窗下所開啟的視窗稱為「子視窗」（Child Window），父視窗只會有一個，子視窗也無法改變成父視窗。由於子視窗接受父視窗的管轄，因此沒有「最大化」、「最小化」和視窗大小的調整。

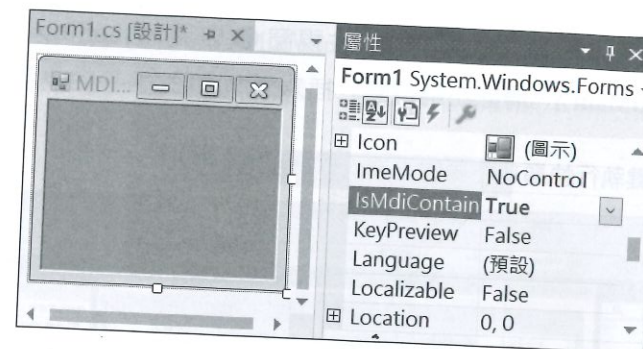
前面章節專案都是以 SDI 表單來運作，這意味著一個專案只會開啟一個表單。如何建立 MDI 父表單？屬性「IsMDIContainer」的『True/False』用來決定它是否成為 MDI 表單；製作程序如下：

STEP 01 產生 MDI 父表單。建立一般表單後，將屬性「IsMDIContainer」變更為「True」來產生 MDI 父表單，進一步作為 MDI 子視窗的容器。

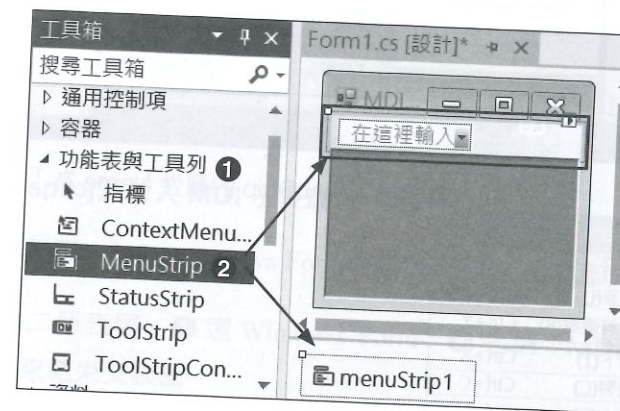
STEP 02 加入 MDI 子表單。同樣加入一般表單，藉由屬性「MDIParent」來指定 MDI 父表單。

範例《CH1201.csproj》建立 MDI 父表單

STEP 01 建立 Windows Form 專案；將表單的屬性 IsMDIContainer 變更為「True」來產生 MDI 父表單。



STEP 02 加入 MenuStrip 控制項建立一個簡單的功能表。① 展開工具箱的功能表與工具列；再以滑鼠雙擊 ② MenuStrip 控制項，它會加到表單頂部和表單底部的「匣」。

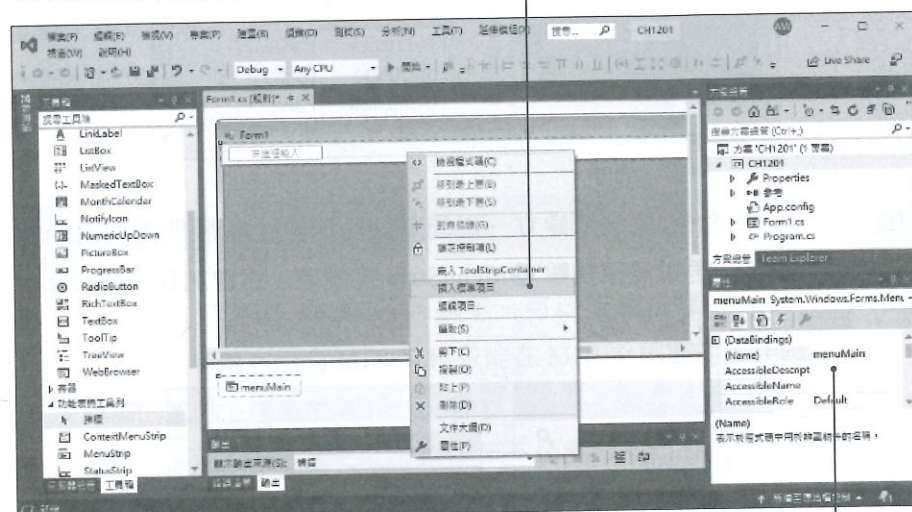


步驟說明

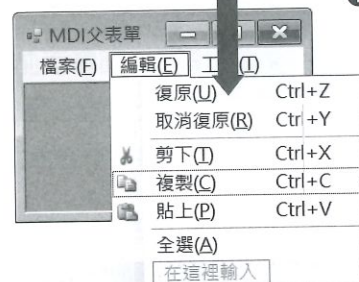
- 當表單變成 MDI 父表單時，背景色會變成深灰色，跟一般的表單不相同。
- 產生 MDI 父表單之後，要有功能表做搭配，執行時「檔案 > 新增」指令，才能在 MDI 父表單之下，產生一個 MDI 子表單。
- 以「標準項目」快速產生一個簡易的功能表。

STEP 03 選取 MenuStrip 控制項，從屬性視窗 ❶ 變更 MenuStrip 的屬性 Name 為「menuMain」，滑鼠左鍵單擊「插入標準項目」。

❷ 按滑鼠右鍵執行快顯功能表的「插入標準項目」指令

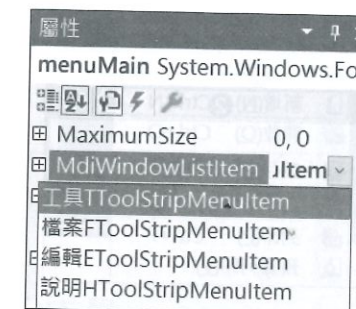


❶ 變更 MenuStrip 的屬性 Name 為「menuMain」



簡易的功能表

STEP 04 將 MenuStrip 控制項的 MdiWindowListItem 屬性指定給「工具」功能表，讓作用中的子表單能取得焦點，並以核取記號顯示。



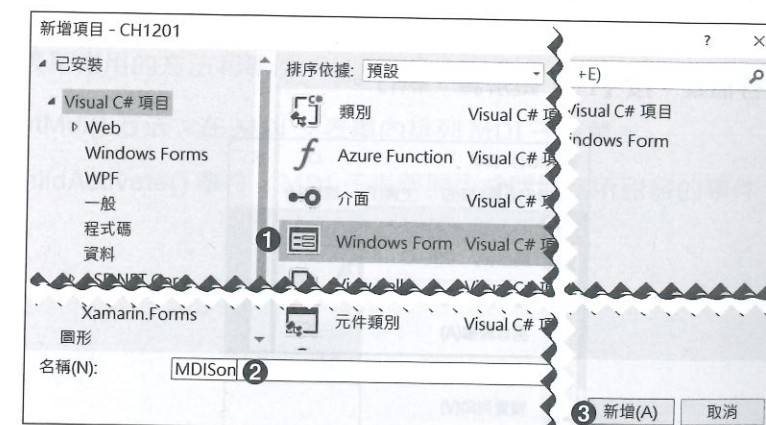
12-1-1 加入 MDI 子表單

完成父視窗的建立後，接著就是加入 MDI 子視窗。必須加入第二個表單來成為 MDI 子視窗的樣版。步驟如下：

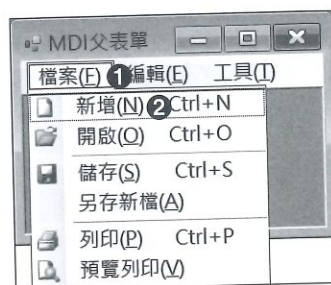
範例《CH1201.csproj》加入 MDI 子表單

STEP 01 執行「專案 > 加入 Windows Form」指令，進入「新增項目」交談窗。

STEP 02 加入第二個表單；❶ 選 Windows Form、❷ 名稱「MDISon」、❸ 按「新增」鈕來結束交談窗。



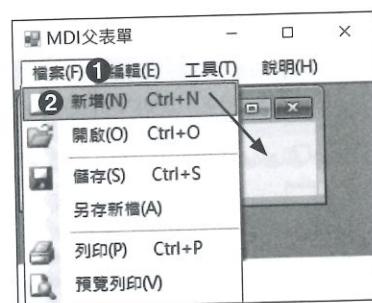
STEP 03 回到 MDI 父表單，展開 ❶ 檔案功能表，滑鼠雙擊 ❷ 「新增」項目，進入程式碼編輯器。



STEP 04 撰寫程式碼，把 MDIChild 表單的 MdiParent 屬性透過程式碼指派 Form1。

```
01 private void 新增ToolStripMenuItem_Click(
02     object sender, EventArgs e)
03 {
04     // 建立子表單
05     MDISon oneChild = new MDISon();
06     // 將目前的表單指定為 oneChild 的 MDI 父表單
07     oneChild.MdiParent = this;
08     oneChild.Size = new Size(50, 100);
09     // 記錄子表單的數量
10     int count = this.MdiChildren.Length;
11     // 設定子表單的標題
12     oneChild.Text = $"我是子表單-{count.ToString()}";
13     oneChild.Show(); // 顯示 MDI 子表單
14 }
```

STEP 05 儲存檔案，按【F5】鍵解譯、執行。



【程式解說】

- 第 5、7 行：依據加入 MDIChild 來實體化子表單物件；將建立的子表單利用 MdiParent 屬性加入父表單中。
- 第 10、12 行：計算子表單的數量，利用 Text 屬性將新加入子表單以「我是子表單 X」顯示 MDI 子表單標題列。
- 如同新增檔案一般，執行「檔案>新增」指令，就能產生新的 MDI 子表單，而且指令每執行一次就能產生 MDI 子表單。

12-1-2 MDI 表單的成員

既然是表單，代表它們成員眾多，表【13-1】乃針對 MDI 父、子表單簡單介紹常用屬性。

○ 表【13-1】MDI 表單常用性

MDI 子表單屬性	說明
IsMdiChild	屬性值 true 會建立一個 MDI 子表單
MdiParent	MDI 表單中指定子表單的父表單
ActiveMdiChild	取得目前作用中的 MDI 子表單
IsMdiContainer	是否要將表單建立為 MDI 子表單的容器
MdiChildren	傳回以此表單為父表單的 MDI 子表單陣列

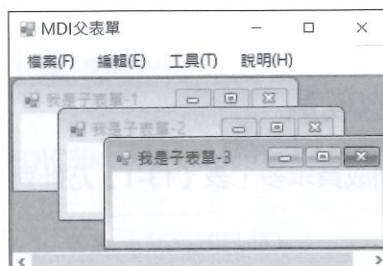
MDI 表單常用的方法和事件：

- LayoutMdi() 方法：在 MDI 父表單內排列 MDI 子表單。
- MdiChildActivate() 事件：MDI 子表單開啟或關閉時所引發的事件。

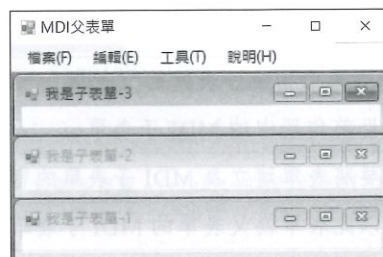
12-1-3 MDI 子表單排列方式

MDI 表單在執行階段可以擁有多個 MDI 子表單，LayoutMdi() 方法能指定其排列方式，列舉它的常數值：

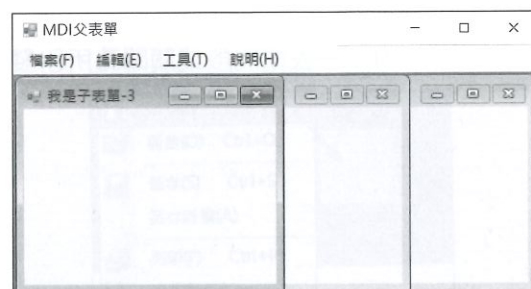
- ▶▶ Arrangelcons：將最小化的 MDI 子表單以圖示排列。
- ▶▶ Cascade：所有 MDI 子視窗重疊 (Cascade) 於 MDI 父表單工作區，這是子表單預設的排列方式。



- ▶▶ TileHorizontal：所有 MDI 子視窗水平並排 MDI 父表單工作區。



- ▶▶ TileVertical：所有 MDI 子視窗垂直並排 MDI 父表單工作區。



撰寫程式碼來排列 MDI 子表單，敘述如下：

```
this.LayoutMdi (MdiLayout.TileHorizontal); // 水平排列
this.LayoutMdi (MdiLayout.TileVertical); // 垂直排列
this.LayoutMdi (MdiLayout.Cascade); // 重疊排序
```

12-2 滑鼠事件

操作表單或控制項時，按一下滑鼠，移動滑鼠所引發的事件，皆屬於滑鼠事件。只不過前面章節都以 Click 事件為主！對於 Windows 應用程式來說，透過滑鼠來接收，並以事件處理常式進行相關程序是非常普遍的動作。在認識滑鼠事件處理之前，先認識發生滑鼠事件時會一併傳送的資訊，部份事件的事件處理者是 MouseEventArgs，它繼承自 System.EventArgs，且位於命名空間 System.Windows.Forms 下，MouseEventArgs 類別擁有五個公用屬性，先以表【12-2】列出一些常見滑鼠事件。

○ 表【12-2】常見的滑鼠事件

事件名稱	說明
Click()	按下滑鼠左鍵一下時就會觸發的事件。
DoubleClick	雙擊滑鼠左鍵時就會觸發的事件。
MouseMove	移動滑鼠時就會觸發的事件。
MouseDown	按下滑鼠按鍵時就會觸發的事件。
MouseUp	放開滑鼠按鍵時就會觸發的事件。
MouseLeave	滑鼠游標離開控制項時就會觸發的事件。
MouseEnter	滑鼠游標進入控制項時就會觸發的事件。