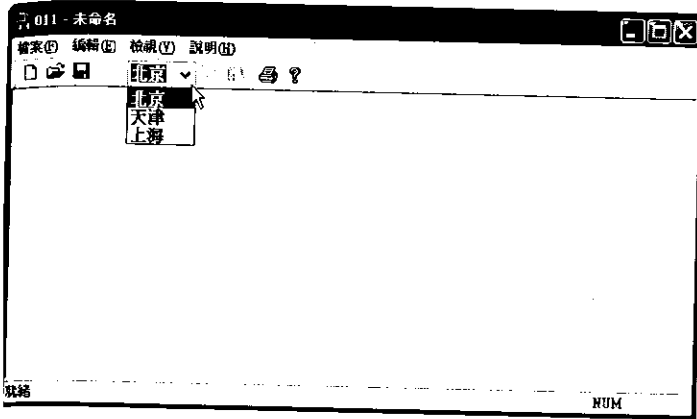


如何在工具列中加入其他控制項

在一些常用軟體如 Word 等，常會出現帶下拉式列示方塊控制項的工具列。在本實例中，我們討論如何在工具列中加入下拉式列示方塊控制項。程式執行結果，如圖 11-1 所示。



◀◀ 圖 11-1
執行結果

技術要點

- ▶ 要在工具列中放入別的控制項視窗，首先應該設計一個足夠大的間隔來放置控制項，然後再手工將所要建立的控制項放入到該位置。
- ▶ 在工具列中增加新的控制項，關鍵在於覆寫主框架的 `OnCreate()` 函式。
- ▶ 手工建立下拉式列示方塊控制項，需要呼叫下拉式列示方塊類別的 `Create` 函式設置控制項的屬性，再呼叫 `AddString` 函式為下拉式列示方塊控制項加入字串。

製作步驟

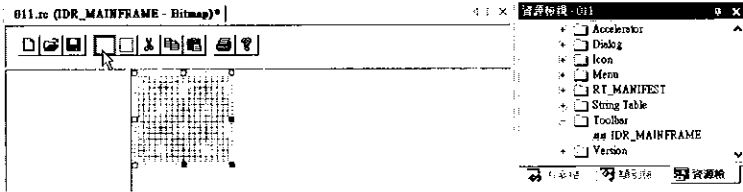
tep01 新建專案

啟動「Visual Studio .NET」，選取「檔案→新增→專案」指令，在專案類型視窗中選擇「Visual C++專案」，在範本框中選擇「MFC 應用程式」。輸入專案名稱（本專案我們用 011 當專案名稱），按一下「確定」鈕。

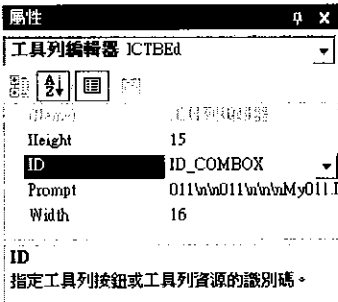
「應用程式類型」選擇「單一文件介面」，其他設置接受系統預設值，按一下「完成」鈕。

Step02 建立工具列下拉式組合方塊

按一下功能表「檢視→資源檢視」指令，打開資源檢視浮動面板。選中「ToolBar → IDR_MAINFRAME」，展開工具列編輯器，在原工具列最右側會有一空白的按鈕。我們用滑鼠將該按鈕拖曳移到要放置下拉方塊的位置，如圖 11-2 所示。並設置其屬性，如圖 11-3 所示。



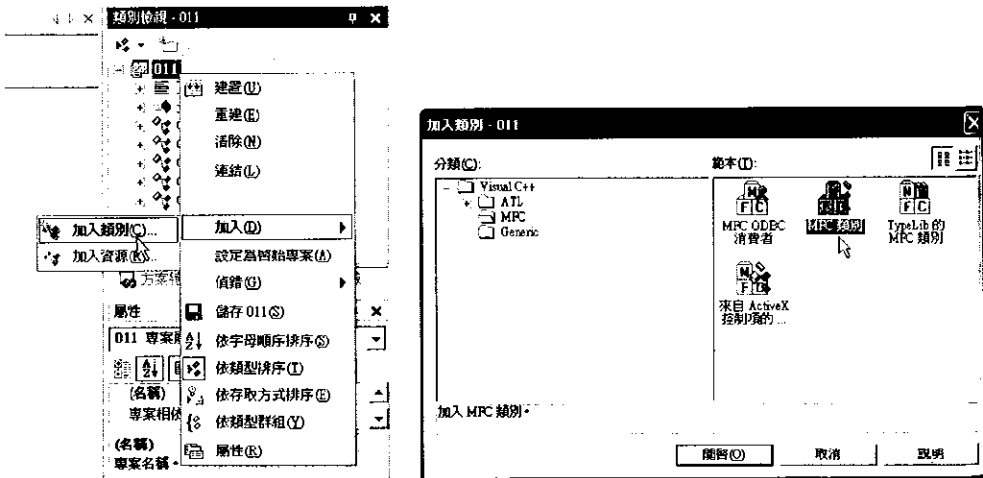
◀◀圖 11-2
將新的按鈕移到要
放下拉方塊的位置



◀◀圖 11-3
在屬性面板中將 ID 改成 ID_COMBO

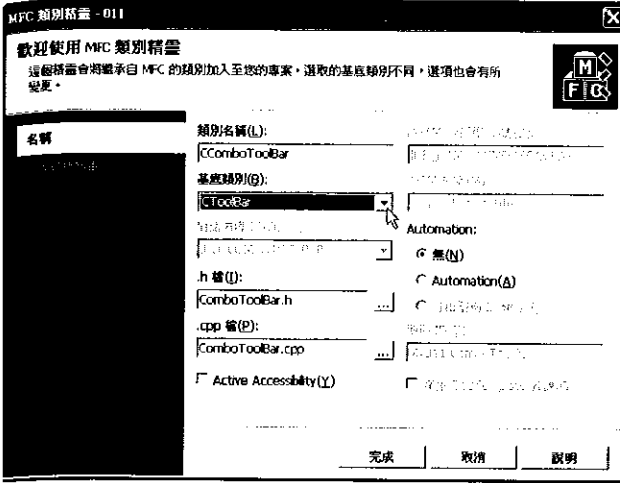
Step03 建立新類別 CComboToolBar

選取「檢視→類別檢視」，以滑鼠右鍵點選專案名稱，如圖 11-4 所示，選取「加入 →加入類別」指令。接著在對話方塊中選取「MFC 類別」，並按下開啟鈕。



▲ 圖 11-4 加入類別

接著會出現 MFC 類別精靈，類別名稱取名為 CComboToolBar，基底類別選 CToolBar，其他則保留預設值，按下完成鈕加入。



◀◀ 圖 11-5
加入新類別 CComboToolBar

Step04 編修及加入程式碼

編輯修改 ComboToolBar.h 檔，宣告成員變數（加粗的部份）：

```
public
    CComboBox m_combobox;
```

接著開啟專案的 MainFrm.h 檔，加入 Include 和變更工具列為 CComboToolBar：

```
// MainFrm.h : CMainFrame 類別的介面
#include "ComboToolBar.h"
.....
protected: // 控制列內嵌的成員
    CStatusBar m_wndStatusBar;
    //CToolBar m_wndToolBar; //原來工具列宣告
    CComboToolBar m_wndToolBar; //有加入下拉方塊的新工具列
.....
```

```

ComboToolBar.h* | ComboToolBar.cpp
CComboToolBar
1: #pragma once
2:
3:
4: // CComboToolBar
5:
6: class CComboToolBar : public CToolBar
7: {
8:     DECLARE_DYNAMIC(CComboToolBar)
9:
10: public:
11:     CComboToolBar();
12:     virtual ~CComboToolBar();
13: public:
14:     CComboBox m_combobox;
15: protected:
16:     DECLARE_MESSAGE_MAP()
17: };
18:
ComboToolBar.h* | ComboToolBar.cpp MainFrm.h
CMainFrame
17: public:
18:
19: // 覆寫
20: public:
21:     virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
22:
23: // 程式碼操作
24: public:
25:     virtual ~CMainFrame();
26: #ifdef _DEBUG
27:     virtual void AssertValid() const;
28:     virtual void Dump(CDumpContext& dc) const;
29: #endif
30:
31: protected: // 控制列內嵌的成員
32:     CStatusBar m_wndStatusBar;
33:     //CToolBar m_wndToolBar; //原來工具列宣告
34:     CComboToolBar m_wndToolBar; //有加入下拉方塊的新工具列
35:
36: // 產生的訊息對應函式
37: protected:
38:     afx_msg int OnCreate(LPCREATESTRUCT lpCreateStruct);
39:     DECLARE_MESSAGE_MAP()
40: };
41:
42:

```

▲ 圖 11-5 編修 ComboToolBar.h 和 MainFrm.h 兩個標頭檔的程式碼

Step05 到 MainFrm.cpp 編修及加入程式碼

開啟 MainFrm.cpp 檔，編修 OnCreate 函式（加粗的部份）：

```

// MainFrm.cpp : CMainFrame 類別的實作
.....

int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct) {
    if (CFrameWnd::OnCreate(lpCreateStruct) == -1)
        return -1;

    ..... (中間省略)

    // TODO: 如果您不要可以固定工具列，請刪除這三行
    m_wndToolBar.EnableDocking(CBRS_ALIGN_ANY);
    EnableDocking(CBRS_ALIGN_ANY);
    DockControlBar(&m_wndToolBar);

    CRect rect;

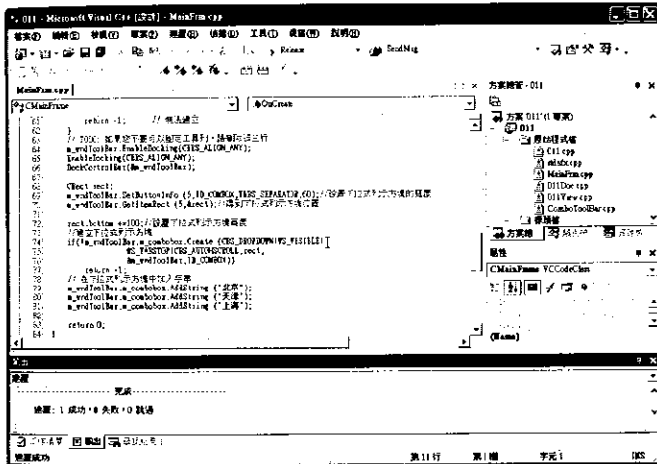
    //設置下拉式列示方塊的寬度
    m_wndToolBar.SetButtonInfo (5, ID_COMBOX, TBBS_SEPARATOR, 60);
    m_wndToolBar.GetItemRect (5, &rect); //得到下拉式列示方塊位置

```

```



rect.bottom +=100;//設置下拉式列示方塊高度
//建立下拉式列示方塊
if(!m_wndToolBar.m_combobox.Create (CBS_DROPDOWN|WS_VISIBLE|
WS_TABSTOP|CBS_AUTOHSCROLL,rect,
&m_wndToolBar,ID_COMBOX))
return -1;
// 在下拉式列示方塊中加入字串
m_wndToolBar.m_combobox.AddString ("北京");
m_wndToolBar.m_combobox.AddString ("天津");
m_wndToolBar.m_combobox.AddString ("上海");
return 0;
}

```



◀圖 11-6
到 MainFrm.cpp 編修及加入
程式碼

Step06 編譯執行程式

產生了程式後，可選取「建置→建置方案」指令或按一下圖示  編譯程序。
接著選取「偵錯→開始」指令或按一下圖示  執行程式。

歸納注解

本實例實現了工具列中下拉式列示方塊（Combo）控制項的添加，掌握了這一點，就可以在工具條中增加各種各樣的控制項，如增加選項（Radio）控制項、核取方塊（Check）控制項、按鈕（按下、複選和單選）、進度指示控制項、編輯框、靜態控制項和新的日期/時間控制項...等。