

How About

# COM

&

# ActiveX Control

With Visual C++ 6.0

Author:

周敬斐

Curtis CHOU

mahler@ms16.hinet.net

This document can be freely release and distribute without modify.

<b>ACTIVE X CONTROLS 基本概念</b> .....	<b>3</b>
使用 ACTIVE X 的好處 .....	3
使用 MFC ACTIVE X CONTROL WIZARD .....	3
MFC ACTIVE X CONTROLS WIZARD 產生的程式 .....	4
MFC 開發的好處 .....	4
註冊控制項 .....	5
ACTIVE X 控制項的繪製 .....	5
使用 ONDRAW 在執行週期與設計週期 .....	5
修改 ONDRAW() 函數 .....	6
<b>ACTIVE X 控制項屬性設定</b> .....	<b>6</b>
內建屬性(STOCK PROPERTIES) .....	6
增加內建屬性到控制項內 .....	6
CLASS WIZARD 為你做了什麼? .....	9
自訂屬性 .....	9
加入自訂屬性 .....	10
使用 GET/SET METHOD 來實作控制項屬性 .....	11
如何建立唯獨的屬性或是唯寫的屬性? .....	12
如何建立參數化的屬性? .....	13
存取控制項周圍屬性 .....	14
<b>ACTIVE X 控制項方法</b> .....	<b>16</b>
加入內建方法 .....	16
使用 CLASS WIZARD 加入內建方法 .....	16
使用 CLASS WIZARD 加入自訂方法 .....	16
<b>ACTIVE X 控制項事件(ACTIVE X CONTROL EVENT)</b> .....	<b>18</b>
加入內建事件 .....	18
加入內建事件 .....	18
加入自訂事件(CUSTOM EVENT) .....	20
<b>建立 ACTIVE X 控制項之屬性對話方塊</b> .....	<b>20</b>
什麼是 ACTIVE X 控制項屬性對話方塊? .....	20
設計一個預設的屬性頁 .....	22
在屬性頁與控制項交換資料 .....	24
加入內建屬性頁 .....	26
加入自訂屬性頁 .....	27
<b>實作單純的中華民國時間控制項</b> .....	<b>30</b>



# ActiveX Controls 基本概念

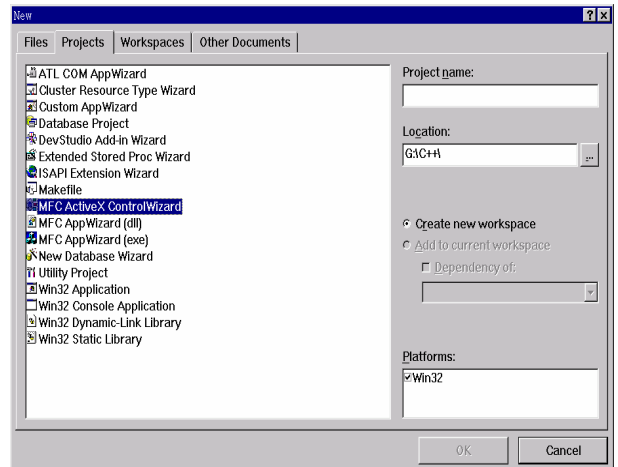
## 使用 ActiveX 的好處

跨越軟體平台?

開發者只要開發一次程式?

網際網路網頁相容與增強?

使用者有多樣化的選擇?



## 使用 MFC ActiveX Control Wizard

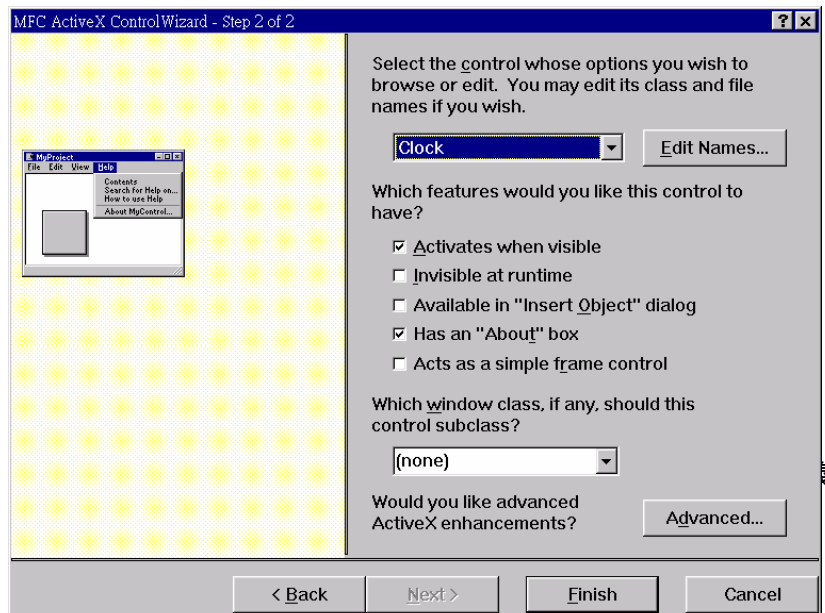
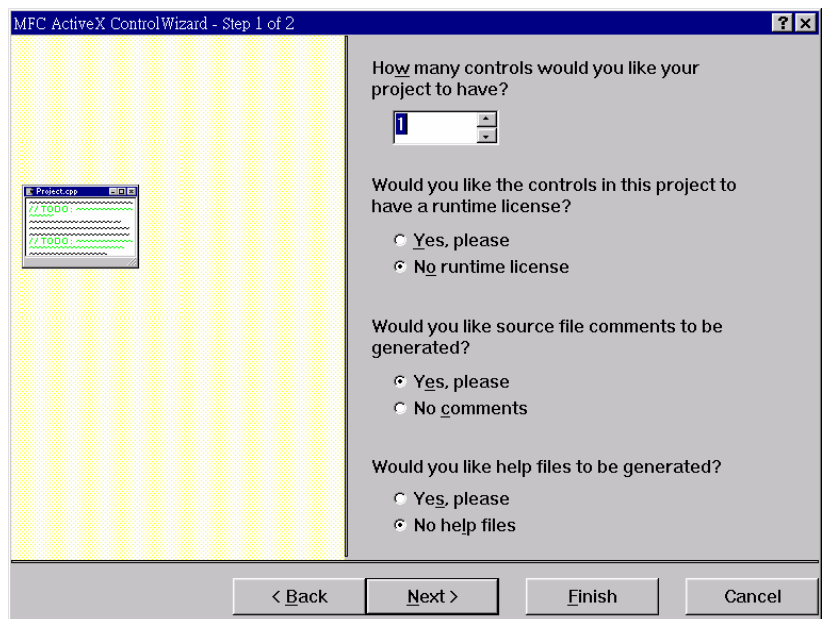
1. File -> New -> Project
2. Choose MFC ActiveX Control Wizard

3. 第一步? 選擇專案控制項數

目? 最多 99 個?

- 甲、設定控制項是否需要 run-time license
- 乙、設定是否要對程式碼產生註解

丙、是否要產生 Help File



#### 4. 第二步

甲、修改控制項名稱

乙、設定控制項額外功能

丙、使用誰為 Window 視窗

### MFC ActiveX Controls Wizard 產生的程式

類別名稱	敘述
Control module	用來初始化控制項動態連結函式庫 DLL? 通常不需要修改? 預設類別名稱為 CyyyApp
Control	控制項實做類別? 包含控制項的服務? 繪圖? 與互動的程式碼? 預設類別名稱為 CyyyCtrl
Property Page	此為控制項的屬性頁? 此屬性頁提供使用者對控制項的設定? 預設類別名稱 CyyyPropPage

### MFC 開發的好處

MFC 提供了一類別 COleControl? 此為控制項物件的基底類別? 此類別封裝了所有 ActiveX 所



需之方法？而 COleControl 的基底類別為 CWnd？所以具備所有 CWnd 相關的方法及屬性？

## 註冊控制項

控制項要使用之前必須先在 Window 的 Registry 中先註冊？而控制項需要在

HKEY\_CLASSES\_ROOT 區域被註冊？而在 HKEY\_CLASSES\_ROOT？CLSID 存在一組 128bit

十六進位的數字？而這一組數字我們又稱 GUID？Global Unique Identifier?? 一般的 DLL 可以

透過 RegSvr32.exe 來註冊所發展的 DLL？

## ActiveX 控制項的繪製

虛擬函數 OnDraw(CDC\* pDC , const CRect& rcBounds, const CRect& rcInvalid) ;

pDC 當發生 drawing 事件時所使用的 DC？

rcBounds 控制項大小？包含邊框？

rcInvalid 控制項失效的區域？

MFC 提供 COleControl::InvalidateControl 與 CWnd::Invalid 一樣意思？

## 使用 OnDraw 在執行週期與設計週期

執行週期？當控制項的應用程式容器送出 WM\_PAINT 之訊息時？其呼叫 COleControl::OnPaint()

產生一個 DC 並傳遞給 OnDraw()？而連帶引發 COleControl::OnDraw()？

設計週期？例如在 VB 中設計表單的時候？由於物件使處在無視窗的狀態下？所以無法得知

WM\_PAINT 訊息？所以這時候控制項改直接呼叫 COleControl::OnDraw()？



## 修改 OnDraw() 函數

不管我們的控制項是否有在運作? Active? 或是失效(InActive)? 應用程式架構(FrameWork)還是不停的呼叫 OnDraw()? 而當控制項有在運作的時候? OnDraw()收到的有效矩形之左上角為(0,0)? 而收到的 DC? Device Context? 會傳遞給包含控制項的子視窗? 假設控制項是沒有運作的话? 則收到的有效矩形之左上角就不一定為(0,0)? 而 DC 將會傳遞給該控制項的容器?

# ActiveX 控制項屬性設定

## 內建屬性(Stock Properties)

一般 ActiveX 控制項中常常會見到一些如文字顏色? 字型等等的屬性頁? 這些大部分都使用內建的屬性? 如下面列舉一樣?

BackColor	決定背景顏色
BorderStyle	邊框型態
FillColor	圖形填入顏色
Font	字串或文字使用的字型
Enabled	決定控制項是否有效
Text	控制項之文字
Caption	控制項之標題

## 增加內建屬性到控制項內

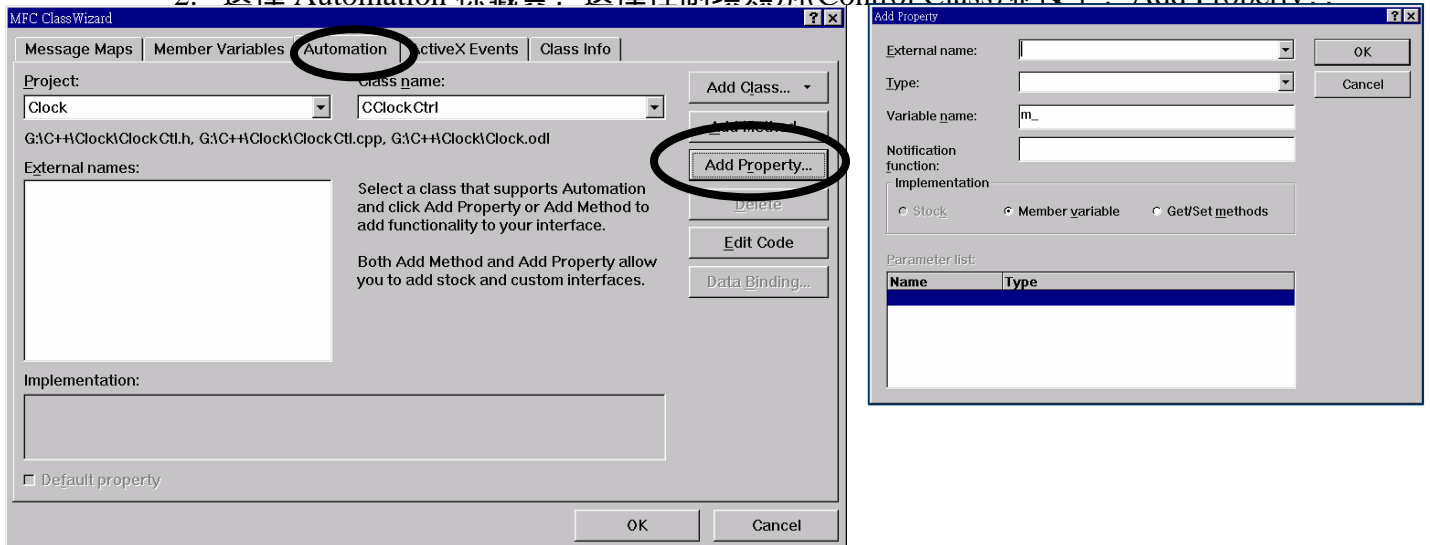
加入內建屬性到控制項內的方式主要使用 VC?? 之 ClassWizard?



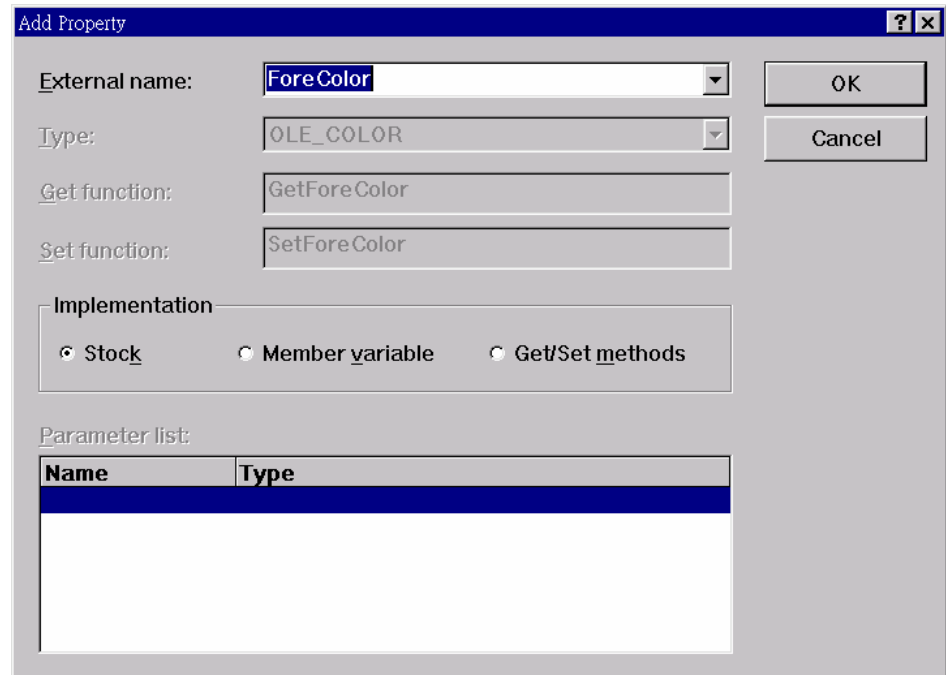
加入步驟?

1. 打開專案並啟動 ClassWizard (Ctrl + W)

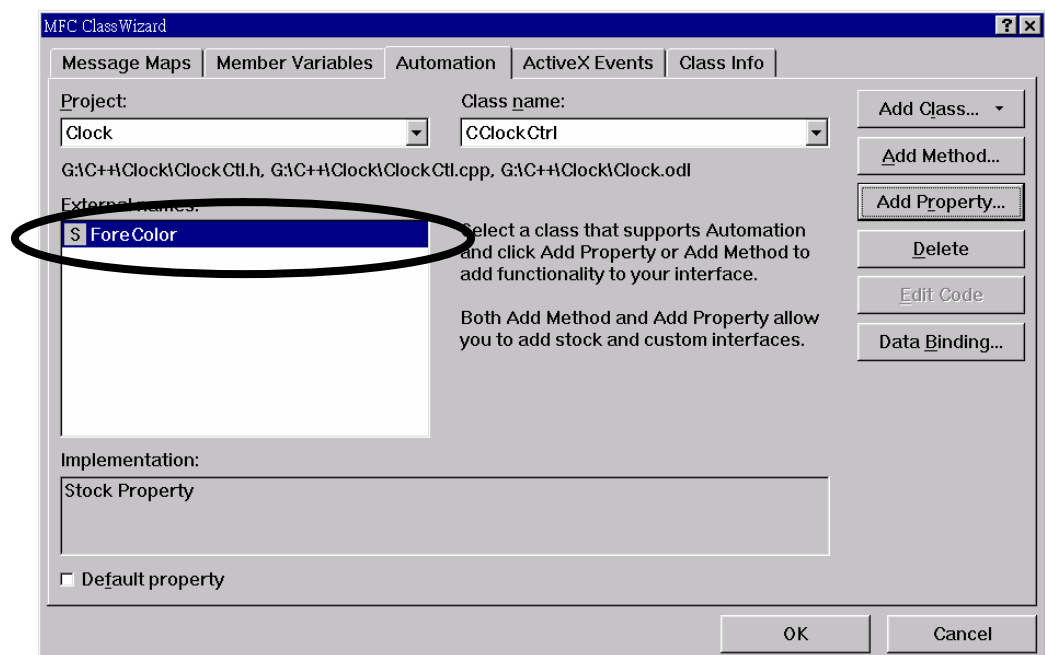
2. 選擇 Automation 標籤頁? 選擇控制項類別(Control Class)並按下? Add Property??



3. 在 External name 中選擇屬性名稱？並按下 OK？



4. 屬性增加成功後 ClassWizard 會出現下面畫面？





## ClassWizard 為你做了什麼？

當每一個你加入 ActiveX 控制項的內建屬性？ ClassWizard 會幫你加入下列的程式碼？

```
BEGIN_DISPATCH_MAP(CClockCtrl, COleControl)
   //{{AFX_DISPATCH_MAP(CClockCtrl)
    DISP_STOCKPROP_FORECOLOR()
    //}}AFX_DISPATCH_MAP
    DISP_FUNCTION_ID(CClockCtrl, "AboutBox", DISPID_ABOUTBOX, AboutBox, VT_EMPTY, VTS_NONE)
END_DISPATCH_MAP()
```

當你的控制項改變屬性設定的時候？ 會自動呼叫 COleControl::InvalidateControl()通知控制項該屬性已經更改？ 而 InvalidateControl 會自動送出相對應的訊息以呼叫相對應的處理函數？ 例如？ 以上面的例子？ 當我更改前景的顏色？ 控制項會自動觸發 InvalidateControl 並送出訊息引法

OnForeColorChanged()？

```
void CClockCtrl::OnForeColorChanged()
{
    // TODO: Add your specialized code here and/or call the base class
    MessageBox("ForeColor is Changed") ;
    COleControl::OnForeColorChanged();
}
```

## 自訂屬性

除了使用內建屬性？ 你也可以使用自訂的屬性？ VC?? 定義了下列四種不同的巨集使我們可以自行增加自己所需要的屬性？

型態	Dispatch map entry macro	使用時機
資料成員需被通知	DISP_PROPERTY_NOTIFY	當控制項的屬性質被更改時？ 會送出訊息引發相對應的處理函數？
資料成員(Member Variable)	DISP_PROPERTY	當控制項的屬性質被改變與否



		都無所謂?
Get/Set Method	DISP_PROPERTY_EX	當你需要在執行週期計算所需的屬性值? 或是實作唯獨或唯寫的屬性值時使用?
參數化	DISP_PROPERTY_PARAM	用來透過一個單一的控制項存取一組值?

Get 與 Set 方法提供了 ActiveX 控制項相當大的彈性? 而且大部分的人都會實作自己的方法使控制項屬性設定上更加完善?

## 加入自訂屬性

在使用內建屬性與自訂屬性的一個不同點是? 你必需要自行指定在 Dispatch map entry macro 中自己指定參數? 加入的方法如下?

1. 打開 ClassWizard
2. 在 Automation 標籤頁中選擇  
所需之 class name 並按下

? Add Property??

3. 在 External name 中打入你自訂的屬性名稱?

4. 在 Type 欄位中指定此屬性的型態?
5. 在 Implementation 標籤頁選擇 Member Variable 或是 Get/Set Method
6. 按下 OK 確定之後? 回到 ClassWizard 在按下 OK 結束?



## 使用 Get/Set Method 來實作控制項屬性

你也可以使用 Get/Set 方法來實作你的控制項屬性？

Get：當使用者呼叫控制項屬性時會被引發？

Set：當使用者要求更改屬性時被引發？

加入方法同上面的步驟？不過在 Add Property 對話方塊中做一點小修改？

The screenshot shows the 'Add Property' dialog box with the following configuration:

- External name: IsChecked
- Type: BOOL
- Get function: GetIsChecked
- Set function: SetIsChecked
- Implementation:  Get/Set methods
- Parameter list: (Empty table)

Name	Type
------	------

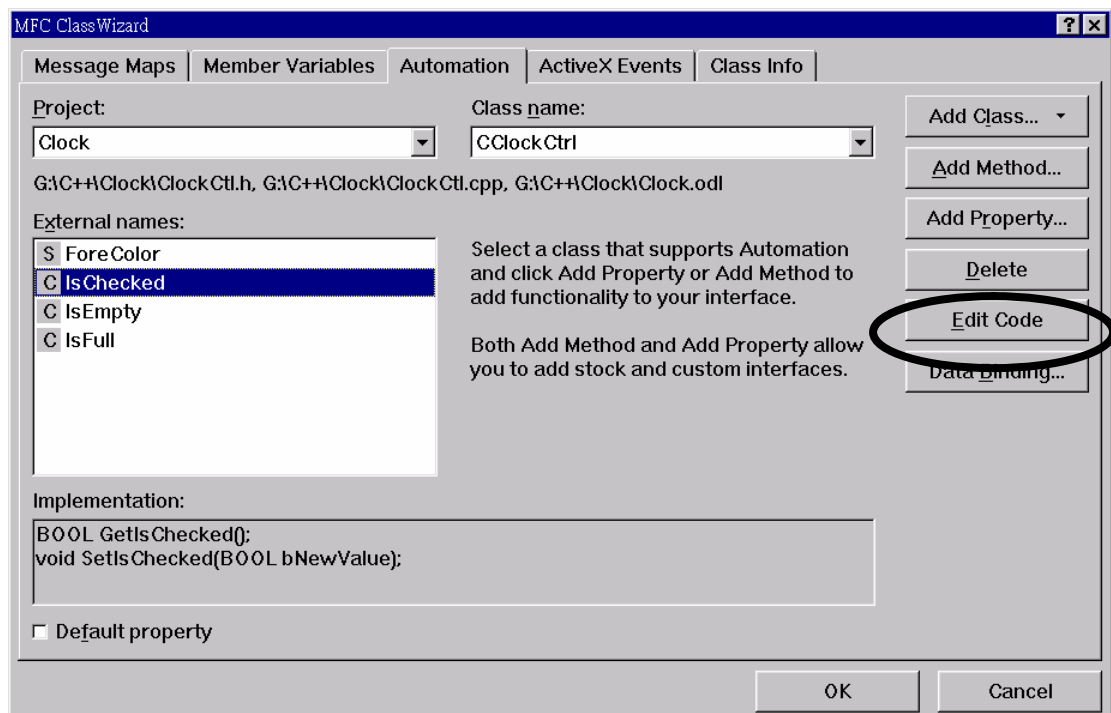


當你完成 Add Property 則你的 ClassWizard 應該會出現你剛剛加入的屬性? 你可以按下 Edit Code

按鈕? 就會自動跳到實作的程式碼? 這個例子產生的程式碼如下?

```
BOOL CClockCtrl::GetIsChecked()
{
    // TODO: Add your property handler here
    if (m_bPaied == TRUE)
        return TRUE;
    else
        return FALSE;
}

void CClockCtrl::SetIsChecked(BOOL bNewValue)
{
    // TODO: Add your property handler here
    m_bPaied = bNewValue ;
    SetModifiedFlag();
}
```



## 如何建立唯獨的屬性或是唯寫的屬性?

如果你要建立 Read-Only 或是 Write-Only 的屬性? 只要藉由刪除 Get 或是 Set 函數即可達成需

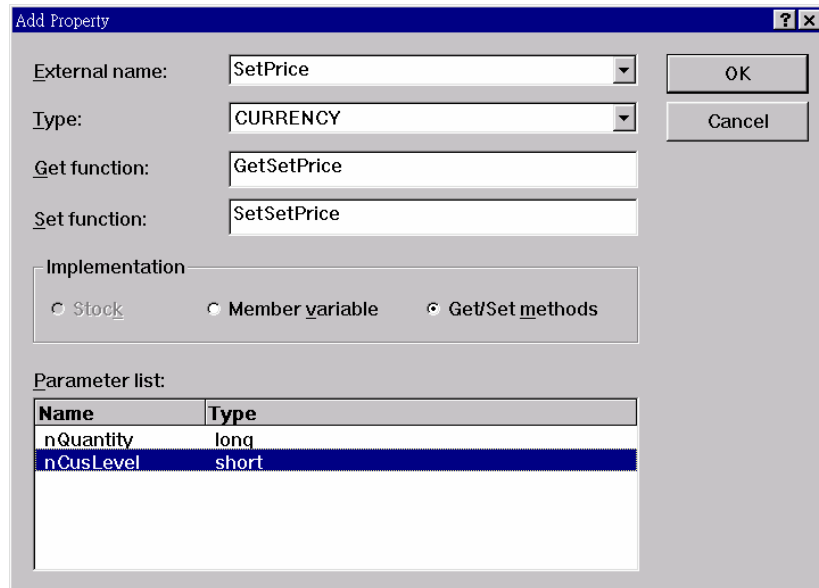
求? 如果需要一個 Read-Only 的屬性? 則刪除 Set 函數? 如果是要 Write-Only 則刪除 Get 函數?



## 如何建立參數化的屬性?

參數化屬性的建立和 Get/Set Method 建立方式一樣? 只是在 Add Property 對話視窗中最下方的

Parameter List 中指定你所需要的參數? 最多只有 15 個??



Name	Type
nQuantity	long
nCusLevel	short

完成之後產生的程式碼大致如下?

```
CURRENCY CClockCtrl::GetSetPrice(long nQuantity, short nCusLevel)
```

```
{  
    CURRENCY cyResult={0,0};  
    // TODO: Add your property handler here  
    return cyResult;  
}
```

```
void CClockCtrl::SetSetPrice(long nQuantity, short nCusLevel, CURRENCY newValue)
```

```
{  
    // TODO: Add your property handler here  
    SetModifiedFlag();  
}
```



## 存取控制項周圍屬性

當你的控制項位於特定的容器上被使用的時候？可能會需要去取得控制項容器的屬性？例如取得控制項所在之表單之顏色？這樣子好和自己的控制項顏色搭配？MFC 提供了一個函數 `COleControl::GetAmbientProperty()` 去存取周圍的屬性值？函數必須給予 Dispatch ID？而這些 Dispatch ID 被定義在 `OleCtl.h` 之中？有需要可以自行去查詢使用？

COleControl function	功用
<code>AmbientBackColor</code>	容器物件之背景顏色
<code>AmbientDiaplayName</code>	容器之名稱
<code>AmbientFont</code>	容器之字型屬性
<code>AmbientForeColor</code>	容器之前景色彩
<code>AmbientLocaleID</code>	區域性的 ID 以允許控制項自己辨識自己
<code>AmbientTextAlign</code>	容器物件之文字對齊屬性
<code>AmbientUserMode</code>	當控制項有兩種運作模式？執行模式？設計模式時使用
<code>AmbientUIDead</code>	允許控制項判斷容器是否要與使用者互動

完整的 Dispatch ID 必需要到 `OleCtl.h` 中查尋？我這裡列出一些在 `OleCtl.h` 中之內容？

```
#define DISPID_AUTOSIZE (-500)
#define DISPID_BACKCOLOR (-501)
#define DISPID_BACKSTYLE (-502)
#define DISPID_BORDERCOLOR (-503)
#define DISPID_BORDERSTYLE (-504)
#define DISPID_BORDERWIDTH (-505)
#define DISPID_DRAWMODE (-507)
#define DISPID_DRAWSTYLE (-508)
#define DISPID_DRAWWIDTH (-509)
```



#define DISPID_FILLCOLOR	(-510)
#define DISPID_FILLSTYLE	(-511)
#define DISPID_FONT	(-512)
#define DISPID_FORECOLOR	(-513)
#define DISPID_ENABLED	(-514)
#define DISPID_HWND	(-515)
#define DISPID_TABSTOP	(-516)
#define DISPID_TEXT	(-517)
#define DISPID_CAPTION	(-518)
#define DISPID_BORDERVISIBLE	(-519)
#define DISPID_APPEARANCE	(-520)



# ActiveX 控制項方法

## 加入內建方法

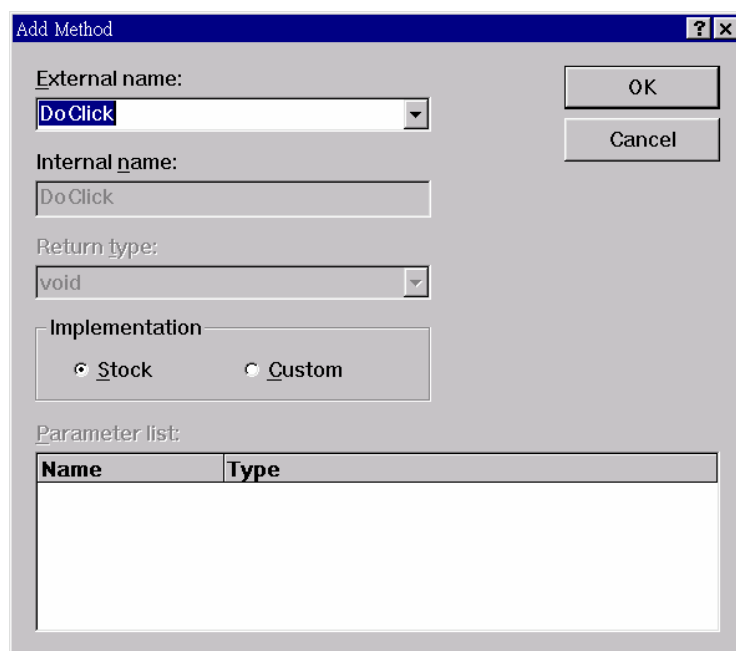
ClassWizard 提供了一些內建方法？例如 DoClick 和 Refresh？

DoClick() 當使用者按下滑鼠左鍵便引發 COleControl::DoClick()

Refresh() 使控制項重新繪製？類似 CWnd::Invalidate()？

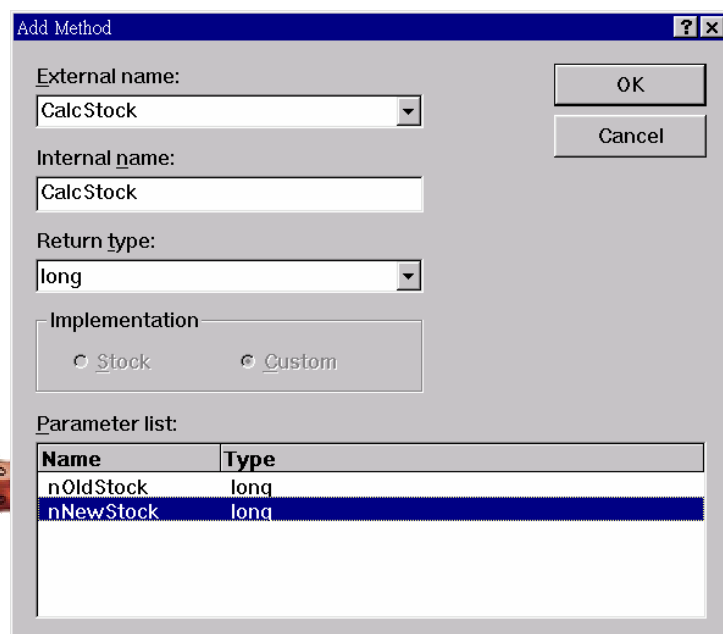
## 使用 ClassWizard 加入內建方法

1. 開啟 ClassWizard 並且選擇 Automation 標籤頁
2. 選擇好你想要加入的 Class 並按下？ Add Method？
3. 在 External name 中選擇你想要的內建方法？
4. 按下 OK 完成？並可選擇 Edit 以修改程式碼？



## 使用 ClassWizard 加入自訂方法

1. 開啟 ClassWizard 並且選擇 Automation 標籤頁





2. 選擇好你想要加入的 Class 並按下? Add Method?
3. 在 External name 中輸入你想要的使用的內建方法?
4. 而 Internal name 也會出現你輸入的名稱?
5. 填入方法的回傳型態(Return Type)?
6. 在 Parameter 中填入你想要加入此方法的參數列?
7. 按下 OK 完成? 並可選擇 Edit 以修改程式碼?



# ActiveX 控制項事件(ActiveX Control Event)

屬性與方法使控制項可以讓使用者進行一些要求或是活動? 而事件則是提供控制項與容器之間的溝通? 一個事件最多可以攜帶 15 個參數? 但是你不能透過事件傳回任何值給 ActiveX Control?

## 加入內建事件

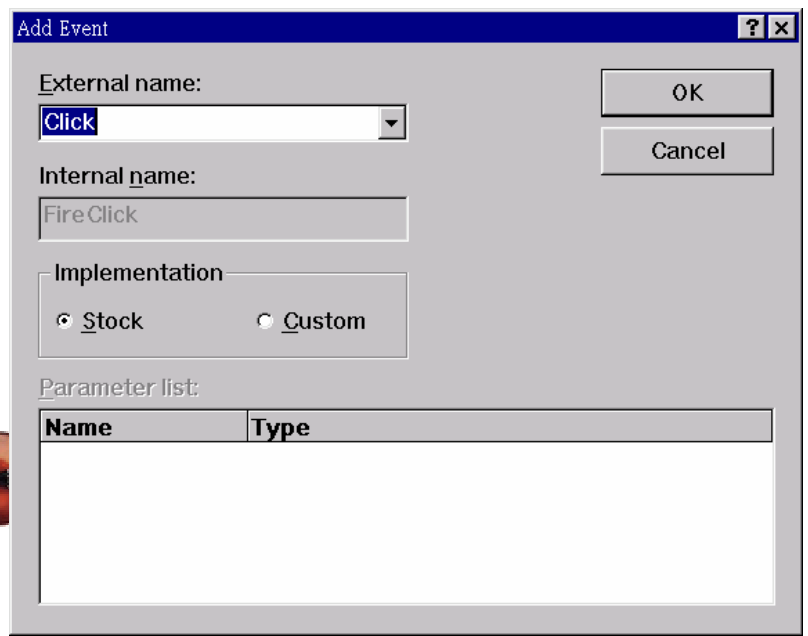
內建事件的訊息處理程式(Handler)由 COleControl 自己有定義預設的處理函數?

事件 Event	何時發生
Click	當使用者按下 Click
DbClick	當使用者按兩下滑鼠左鍵
Error	當控制項想要通知容器有錯誤發生
KeyDown	當控制項有焦點時使用者按下一個鍵
KeyUp	當控制項有焦點時使用者放開一個鍵
MouseDown	當按下滑鼠按鍵
MouseUp	當放開滑鼠按鍵
MouseMove	當移動滑鼠時

## 加入內建事件

1. 開啟 ClassWizard
2. 選擇 ActiveX Event 標籤頁? 並選擇

你想要的 CLASS NAME?



3. 按下[Add Event]? 並在 External name 中選擇你想要的內建事件?
4. 按下 OK 已完成加入事件?

完成加入內建物件則產生下列程式碼?

```
BEGIN_EVENT_MAP(CClockCtrl, COleControl)
   //{{AFX_EVENT_MAP(CClockCtrl)
    EVENT_STOCK_CLICK()
   //}}AFX_EVENT_MAP
END_EVENT_MAP()
```



## 加入自訂事件(Custom Event)

1. 開啟 ClassWizard? 並選擇 ActiveX Event 標籤頁

2. 選擇[Add Event]?

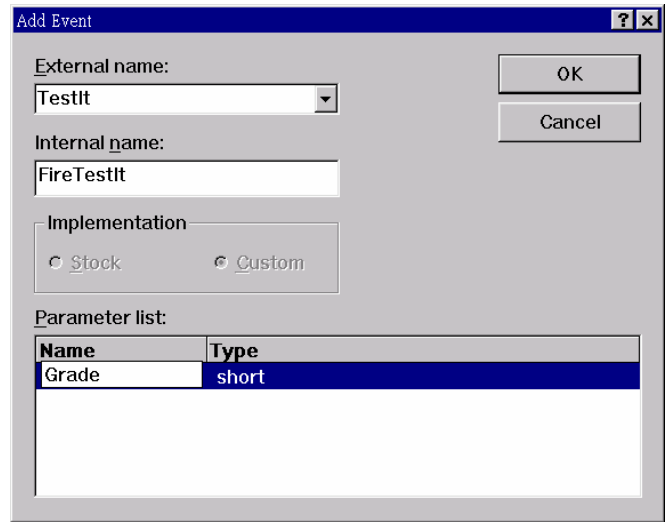
3. 在 External Name 中輸入你想要的事件名稱?

4. 而 Internal name 所表示的是事件觸發的函

數? 其中命名多為 FireXxxx? Xxxx 表示你的事件名稱??

5. 如果想要自己加入參數? 則到 Parameter 中加入所需要的參數即可?

6. 按下 OK 完成?



完成後在 EVENT MAP 中會加入 EVENT\_CUSTOM()的巨集?

```
BEGIN_EVENT_MAP(CClockCtrl, COleControl)
//{{AFX_EVENT_MAP(CClockCtrl)
EVENT_CUSTOM("TestIt", FireTestIt, VTS_I2)
EVENT_STOCK_CLICK()
//}}AFX_EVENT_MAP

END_EVENT_MAP() 中
```

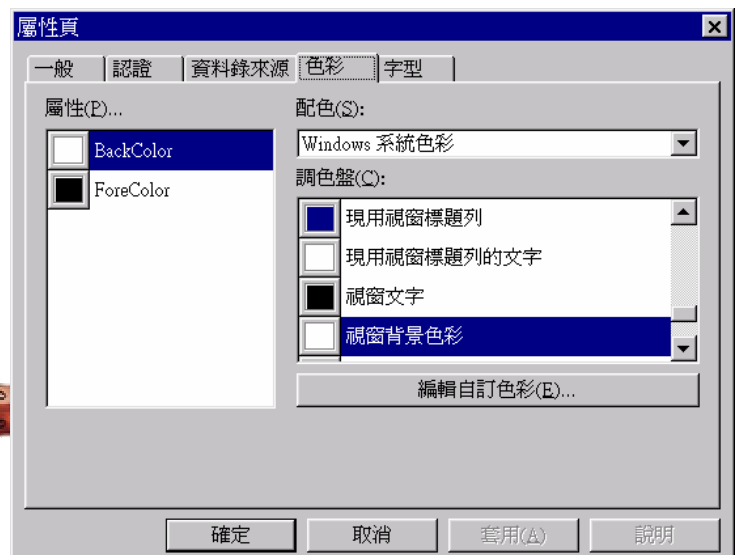
## 建立 ActiveX 控制項之屬性對話方塊

什麼是 ActiveX 控制項屬性對話

方塊?

為了方便使用控制項? 我們通常會提供

一組互動式的介面給控制項與使用者互



動？例如我們在 VB 中對一個 ActiveX 控制項按下右鍵並選擇屬性？你就可以看到一組對話方塊上面佈滿了該控制項所需要的屬性？

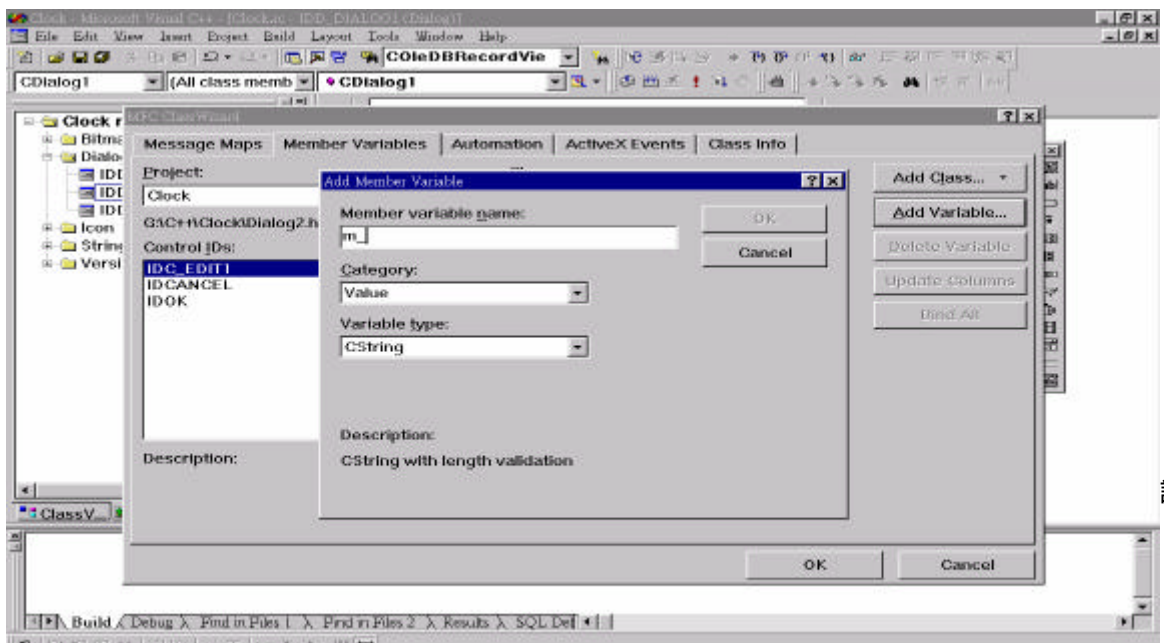
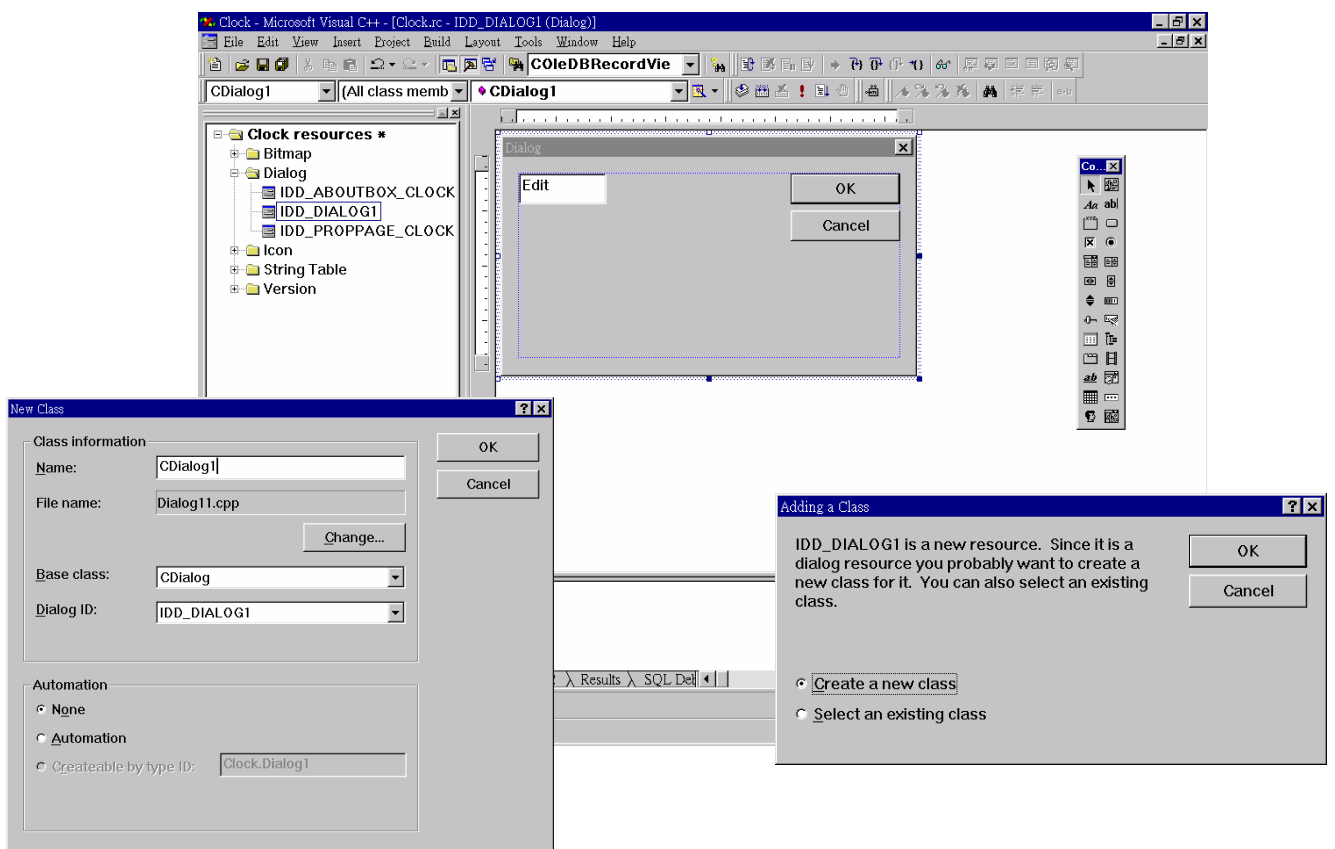
而在 MFC 中我們只需要的資源編輯器？去編輯我們所需要的對話方塊？並且建立幾個 Dialog Class 用來處理屬性對話視窗的訊息？就可以產生一個和右圖一樣的屬性視窗了？



# 設計一個預設的屬性頁

一個屬性對話視窗中可以擁有好幾個屬性頁(Property Page)? 而我們目前就是要來實作這些屬性頁? 建立的步驟如下?

1. 在資源編輯器中的 Dialog 區域加入一個新的 DIALOG
2. 為 DIALOG 加入一些控制項來控制我們的 ActiveX 控制項屬性
3. 使用 ClassWizard 建立 Dialog Class 並且加入適當的資料成員以維持資料?
4. 按下 Member Variable? 並在 Control ID 中? 選擇你想要加入的變數並按下[Add Variable]





當完成上述的步驟？你可以到該對話方塊的類別中去找一個叫做 DoDataExchange 的函數？

這個函數透過 DDX(Do Data Exchange)來對表單上的控制項與類別中的資料成員進行資料的交換？

也就是說？在表單上填入的資訊？一旦按下 OK 就會引發 DoDataExchange()並且把表單上的值更新到類別中相對應的資料成員？

例如？

```
void CDialog2::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
   //{{AFX_DATA_MAP(CDialog2)
    DDX_Text(pDX, IDC_EDIT1, m_Test1);
   //}}AFX_DATA_MAP
}
```

在 DDX 中有很多巨集可以使用？在此我就不多介紹？請自己參閱 MFC 相關書籍的 Property Dialog？

## 在屬性頁與控制項交換資料

DoDataExchange()適用來把對話方塊上的資料傳送到資料成員？而我們現在教各位使用 DDP 函數來把控制項的屬性資料成員與對話方塊的控制項屬性結合在一起？

```
DDP_Text(pDX , IDC_CAPTION , “Caption” )?
```

其中 pDX 為資料交換的指標？而 IDC? CAPTION 是表單上的物件 ID? 而 Catpion 則是 ActiveX 中的 Caption 屬性？相關的 DDP 函數我就列在下面表格？

DDP_CBIndex	連結 combo box 中所選擇的字串索引號碼
-------------	--------------------------





DDP_CBString	連結 combo box 中所選擇的字串
DDP_CBStringExact	結合控制項屬性與 combo box
DDP_Check	結合控制項屬性與 Check Box
DDP_LBIndex	連結 list box 之字串索引
DDP_LBString	連結 list box 中之字串
DDP_LBStringExact	結合控制項屬性與 list box?
DDP_Radio	結合控制項屬性與 Radio Button
DDP_Text	結合控制項屬性和文字



一個完整的 DDP 程式實例？

```
//{{AFX_DATA_MAP(CAlarmPropPage)
DDP_Text(pDX, IDC_ALARM_HOUR, m_AlarmHour, _T("AlarmHour"));
DDX_Text(pDX, IDC_ALARM_HOUR, m_AlarmHour);
DDV_MinMaxInt(pDX, m_AlarmHour, 0, 23);
DDP_Text(pDX, IDC_ALARM_MINUTE, m_AlarmMinute, _T("AlarmMinute"));
DDX_Text(pDX, IDC_ALARM_MINUTE, m_AlarmMinute);
DDV_MinMaxInt(pDX, m_AlarmMinute, 0, 59);
DDP_Text(pDX, IDC_ALARM_SOUND, m_AlarmSound, _T("AlarmSound"));
DDX_Text(pDX, IDC_ALARM_SOUND, m_AlarmSound);
DDP_Text(pDX, IDC_ALARM_COMMAND, m_AlarmCommand, _T("AlarmCommand"));
DDX_Text(pDX, IDC_ALARM_COMMAND, m_AlarmCommand);
DDP_Check(pDX, IDC_ALARM_SET, m_bAlarmSet, _T("AlarmSet"));
DDX_Check(pDX, IDC_ALARM_SET, m_bAlarmSet);
DDP_CBIndex(pDX, IDC_ALARM_TYPE, m_AlarmType, _T("AlarmType"));
DDX_CBIndex(pDX, IDC_ALARM_TYPE, m_AlarmType);
//}}AFX_DATA_MAP
DDP_PostProcessing(pDX);
```

## 加入內建屬性頁

MFC 提供了一些內建的屬性頁可以供我們使用？

內建物件頁	Class ID
Font chooser	CLSID_CFontPropPage
Picture chooser	CLSID_CPicturePropPage
Color chooser	CLSID_CColorPropPage

屬性頁的增加非常簡單？只要在 CXxxCtrl 中找到下面的屬性頁對應

```
BEGIN_PROPPAGEIDS(CClockCtrl, 3)
    PROPPAGEID(CClockPropPage::guid)
    PROPPAGEID(CAlarmPropPage::guid)
    PROPPAGEID(CLSID_CColorPropPage)
END_PROPPAGEIDS(CClockCtrl)
```

就可以自由加入你想要的屬性了？



## 加入自訂屬性頁

在加入屬性頁前？有幾點要注意？

1. 盡量不要有太多的屬性頁
2. 盡量不要讓屬性頁太複雜？頁與頁之間的關連性要低一點
3. 在同一個頁面最好有相同的關係

建立自訂屬性頁主要的步驟

1. 建立一個新的 Property Page
2. 在 Property Page 中加入控制項
3. 建立一個新的 Property Class
4. 把該 Property Page 加入控制項之屬性頁對應

以上是主要的建立步驟？細節我們在下面一一探討？

建立屬性頁對話方塊於資源編輯器中

1. 使用資源編輯器建立一個新的對話視窗資源
2. 命名該對話視窗的名稱為 IDD\_PROPPAGE\_Xxxx(Xxxx 為你想要的名稱)
3. 設定對話方塊的樣視為 child window 與 thin border
4. 把 TitleBar 與 Visible 的選項都清除
5. 把對話視窗的大小設定在 250 x 62 或是 250 x 110 個單位？這樣子做以避免

compiler 在編譯的過程中出現不可知的警告？

建立新的屬性頁類別

- i. . 啟動 ClassWizard



- ii. 並建立一個新的類別？其類別的父類別為 COlePropertyPage？
- iii. 按下 OK 並結束？

### 設定屬性頁的名稱與標題

1. 打開資源編輯器中的 String Table？並加入 ID？例如 IDS\_Xxxx\_PPG 其字串為  
Xxxx Property Page？
2. 再加入一個視窗標題的字串資源？例如 IDS\_Xxxx\_PPG\_CAPTION 其字串為  
Xxxx？
3. 使用屬性頁名稱註冊一個屬性頁？在 UpdateRegistry() 函數中？更新  
AfxOleRegisterPropertyPage 函數？才能使你的屬性頁可以作用？屬性頁必須透過  
註冊才可以使用？

```
BOOL CXxxx::CXxxxFactory::UpdateRegistry(BOOL bRegister)
{
// TODO: Define string resource for page type; replace '0' below with ID.
if (bRegister)
    return AfxOleRegisterPropertyPageClass(AfxGetInstanceHandle(), m_clsid, IDS_Xxxx_PPG);
else
    return AfxOleUnregisterClass(m_clsid, NULL);
}
```

4. 皆下來請在這個屬性頁類別的建構子中？修改一下？把初始化串列中的父建構子

COlePropertyPage() 的第二個參數加入剛剛我們設定的屬性頁標題 ID？

```
CXxxx::CXxxx() :
    COlePropertyPage(IDD, IDD_Xxxx_PPG_CAPTION)
{
//{{AFX_DATA_INIT(CXxxx)
// NOTE: ClassWizard will add member initialization here
// DO NOT EDIT what you see in these blocks of generated code !
//}}AFX_DATA_INIT
}
```



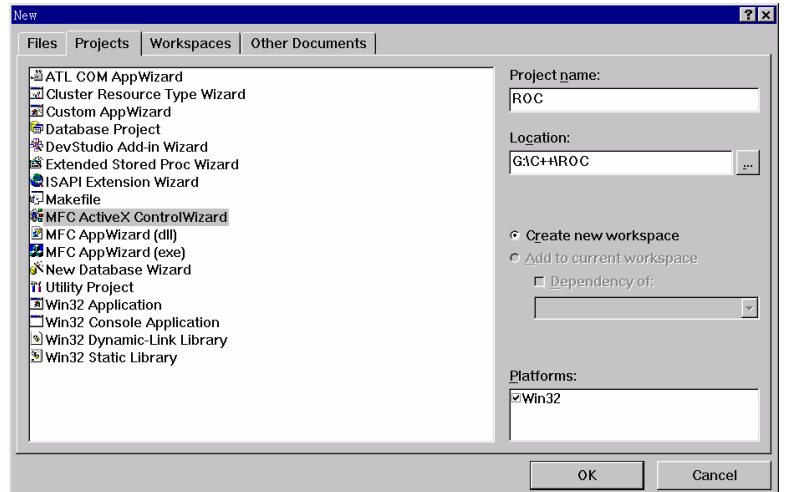
5.加入屬性頁到控制項屬性頁清單? map? 之中



# 實作單純的中華民國

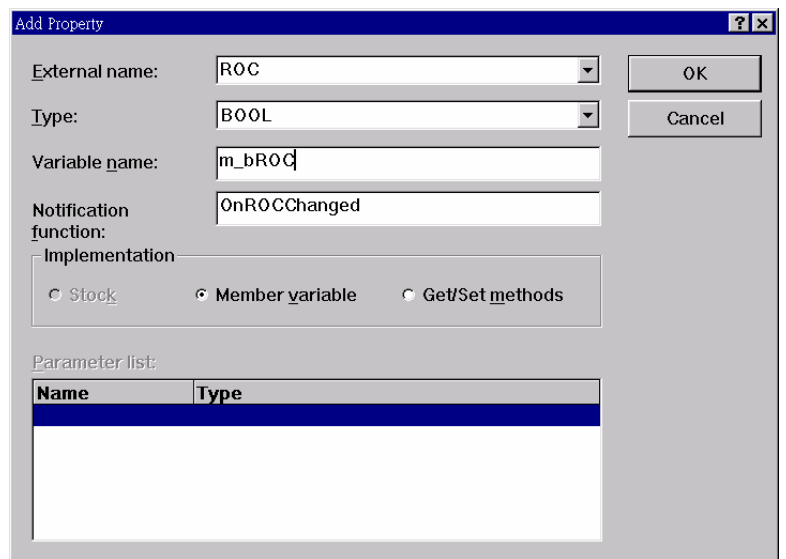
## 時間控制項

1. 建立新的專案? 並取名為 ROC
2. 接下來精靈所出現的畫面都採用預設值進行?



3. 展開 Class? 並啟動 ClassWizard? 為我們的控制項加入一個屬性? 這個屬性我們命名為 ROC? 成員變數的名稱我們命名為 m\_bROC? 型態為 BOOL?

4. 按下 OK 後? 在 CLASSWIZARD 中按下 EDIT CODE 以編輯程式碼? 我們在 OnRocChanged() 中加入 InvalidateControl() 以要求整個控制項重新繪製?



```
void CROCCtrl::OnROCChanged()
{
    // TODO: Add notification handler code
    InvalidateControl();
    SetModifiedFlag();
}
```



}

5. 接下來請在 CROCCtrl 中加入兩個資料成員? CString m\_strROC; 和 CString m\_strSTD;  
分別用來記錄出現中華民國年份還是一般的西洋年份?



6. 把 CROCCtrl::OnDraw 內容修正成下面?

```
// TODO: Replace the following code with your own drawing code.
CBrush brush(RGB(128,128,0));
pdc->FillRect(rcBounds, &brush);
int x = rcBounds.right / 3;
int y = rcBounds.bottom / 4;
if ( m_bROC) pdc->TextOut(x, y, m_strROC);
else pdc->TextOut(x, y, m_strSTD);
```

7. 在 CROCCtrl 中利用 ClassWizard 加入兩個 MESSAGE HANDLER? 一個是 WM\_CREATE 其對應的處理器為 OnCreate() 另外一個為 WM\_DESTROY 其對應的處理函數為 OnDestroy();

8. 把 OnCreate() 改寫成下面的樣子

```
int CROCCtrl::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    if (COleControl::OnCreate(lpCreateStruct) == -1)
        return -1;
    SetTimer(1, 1000, NULL);
    return 0;
}
```

9. 把 OnDestroy() 也改寫一下

```
void CROCCtrl::OnDestroy()
{
    COleControl::OnDestroy();
    KillTimer(1);
}
```

10. 由於我們已經設定好 TIMER? 接下來要在用 ClassWizard 加入一個訊息 WM\_TIMER? 用

來回應 TIMER 訊息?

```
void CROCCtrl::OnTimer(UINT nIDEvent)
{
    // TODO: Add your message handler code here and/or call default
    COleDateTime today = COleDateTime::GetCurrentTime();
    char year[5];
    char output[50];
    strcpy(output, "中華民國? ");
    itoa(today.GetYear() - 1911, year, 10);
```





```

strcat(output , year ) ;
strcat(output , "年" ) ;
m_strROC = output ;

strcpy(output , "西洋年份? " ) ;
itoa(today.GetYear() , year , 10 ) ;
strcat(output , year ) ;
strcat(output , "年" ) ;
m_strSTD = output ;

COleControl::OnTimer(nIDEvent);
}

```

11. 接下來我們要增加 Property Page 供使用者使用?

12. 請在 Resource Editor 建立一個 OlePropertyPage 的對話方塊?

13. 並為該對話視窗命名為 IDD\_PROPPAGE\_SET? 除此之外? 請在 String Table 中建立

IDS\_SET\_PPG 內容為[設定使用民國表示]? 以及 IDS\_SET\_PPG\_CAPTION 內容為[民國]?

14. 設定完成之後? 按照我前面提到的步驟? 利用 ClassWizard 建立一個新的 CLASS 叫做

CSetPropPage? 其父類別為 COlePropertyPage.以及在 UpdateRegistry()中更改註冊 ID ? 以及

在建構子改變父類別的第二個參數? 最後請修改 CROCCtrl 中的 Property Page Map?

15. 接下來請使用 ClassWizard 為表單上拉好的 IDC\_CHECK1 增加一個資料成員名為 m\_bROC;

其型態為 BOOL?

16. 最後在手動加入 DDP 巨集? 請打開 CSetPropPage::OnDataExchange()? 請加入一行?

```

void CSetPropPage::DoDataExchange(CDataExchange* pDX)
{
    // NOTE: ClassWizard will add DDP, DDX, and DDV calls here
    // DO NOT EDIT what you see in these blocks of generated code !
   //{{AFX_DATA_MAP(CSetPropPage)
    DDX_Check(pDX, IDC_CHECK1, m_bROC);
    DDP_Check(pDX, IDC_CHECK1 , m_bROC , "ROC");
    //}}AFX_DATA_MAP
    DDP_PostProcessing(pDX);
}

```



