



# Visual C++ 2012 影像處理

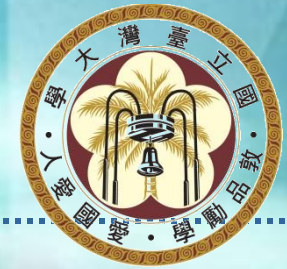
NTU CSIE 張傑帆



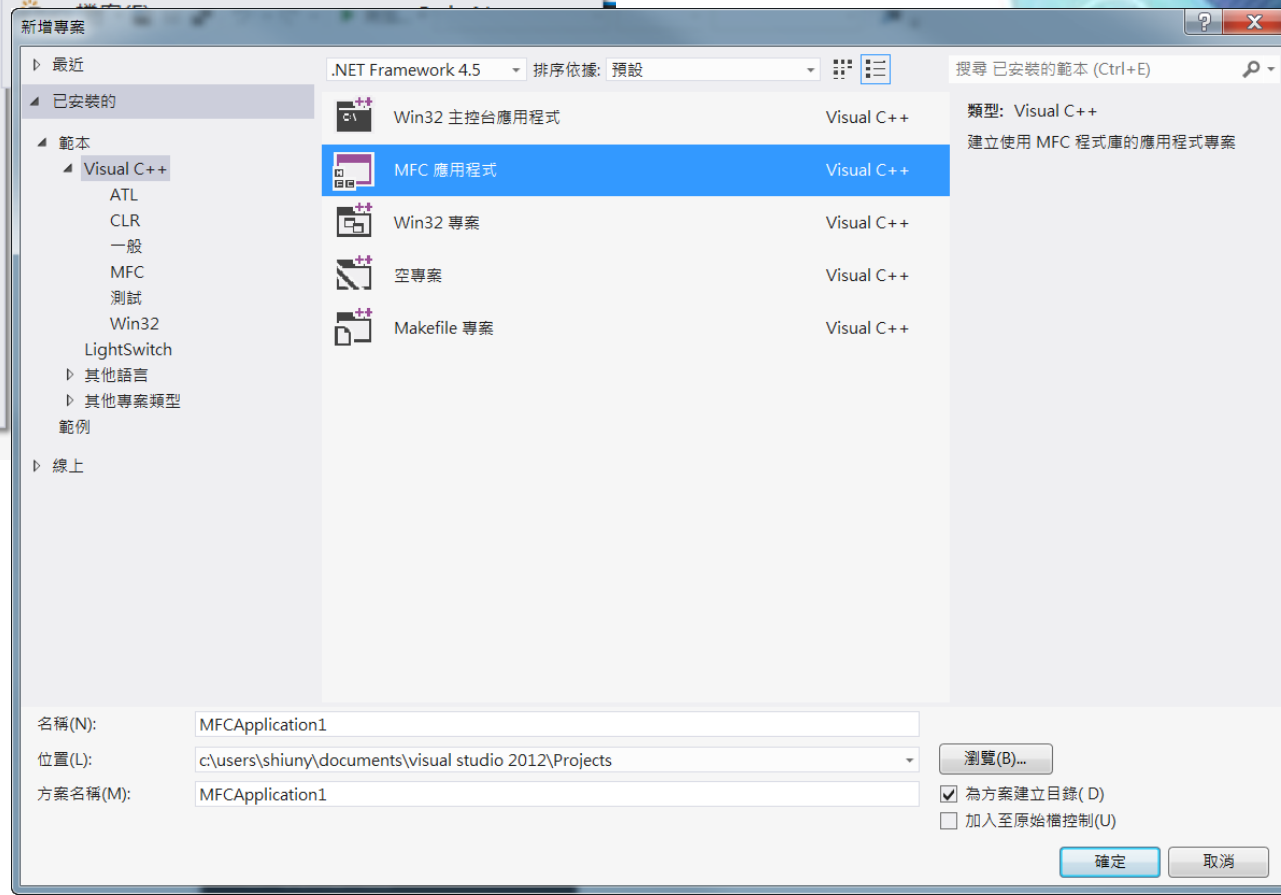
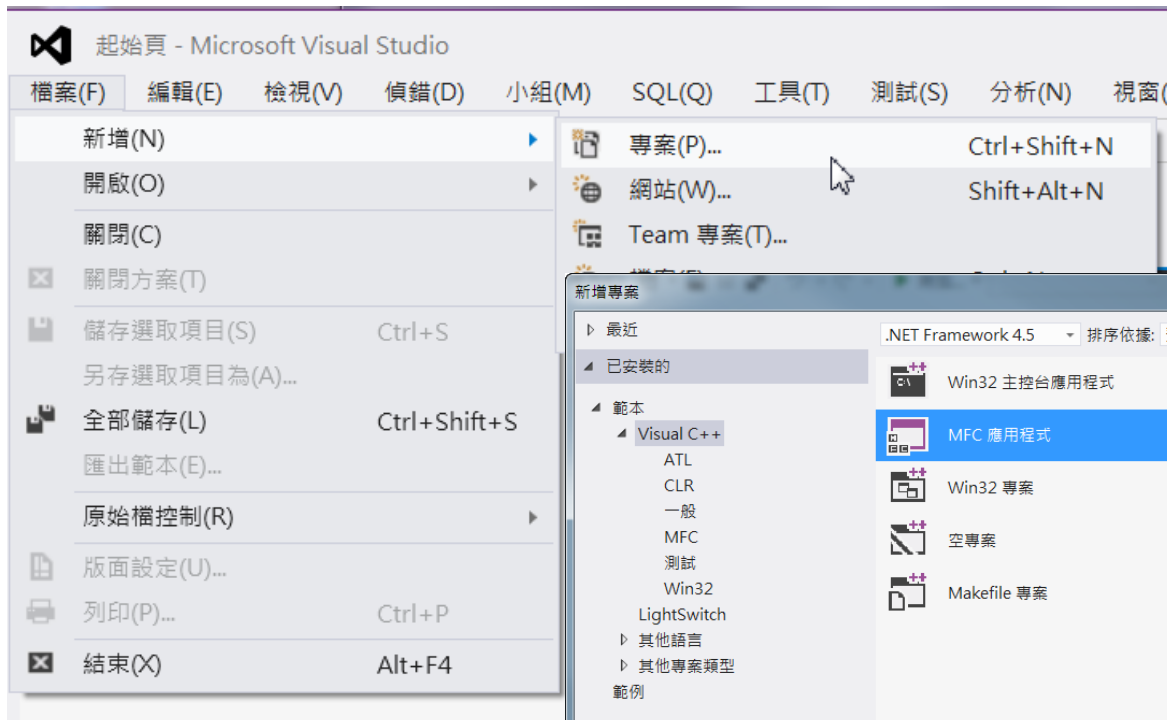
# Contents

---

- Visual C++ MFC 簡介
  - 新增專案
  - stdafx.h
- MFC應用於影像處理
  - 讀寫與顯示影像檔
  - 簡易影像處理範例
  - 常見問題及處理方式



# 新增專案





MFC 應用程式精靈 - MFCApplication1

 應用程式類型

概觀

應用程式類型

複合文件支援

文件範本屬性

資料庫支援

使用者介面功能

進階功能

產生的類別

應用程式類型:

☐ 單一文件(S)

☒ 多重文件(M)

☐ 索引標籤式文件(B)

☐ 對話方塊式(D)

☐ 使用 HTML 對話方塊(I)

☐ 無增強型 MFC 控制項(O)

☐ 多重最上層文件(T)

☒ 文件/檢視架構支援(V)

☒ 安全性開發週期 (SDL) 檢查(C)

資源語言(L):

中文 (繁體, 台灣)

☒ 使用 Unicode 程式庫(N)

專案樣式:

☒ MFC 標準(A)

☐ 檔案總管(X)

☐ Visual Studio(O)

☐ Office(E)

視覺化樣式和色彩(Y):

Windows 原生/預設

☐ 啟用視覺化樣式切換(C)

MFC 的使用:

☒ 使用 MFC 的共用 DLL(U)

☐ 使用 MFC 的靜態程式庫(E)

方案總管

搜尋 方案總管 (Ctrl+;)

方案 'MFCApplication1' (1 專案)

- MFCApplication1
  - 原始程式檔
    - ChildFrm.cpp
    - MainFrm.cpp
    - MFCApplication1.cpp
    - MFCApplication1Doc.cpp
    - MFCApplication1View.cpp
    - stdafx.cpp
  - 外部相依性
  - 標頭檔
    - ChildFrm.h
    - MainFrm.h
    - MFCApplication1.h
    - MFCApplication1Doc.h
    - MFCApplication1View.h
    - Resource.h
    - stdafx.h
    - targetver.h
  - 資源檔
    - MFCApplication1.ico
    - MFCApplication1.rc
    - MFCApplication1.rc2
    - MFCApplication1Doc.ico
    - Toolbar.bmp
    - ReadMe.txt

方案總管 類別檢視 屬性管... 資源檢視 Team...

類別檢視

<搜尋>

- MFCApplication1
  - 對應
    - 巨集和常數
    - 全域函式和變數
  - CAboutDlg
  - CChildFrame
  - CMainFrame
  - CMFCApplication1App
  - CMFCApplication1Doc
  - CMFCApplication1View

方案總管 類別檢視 屬性管... 資源檢視 Team...

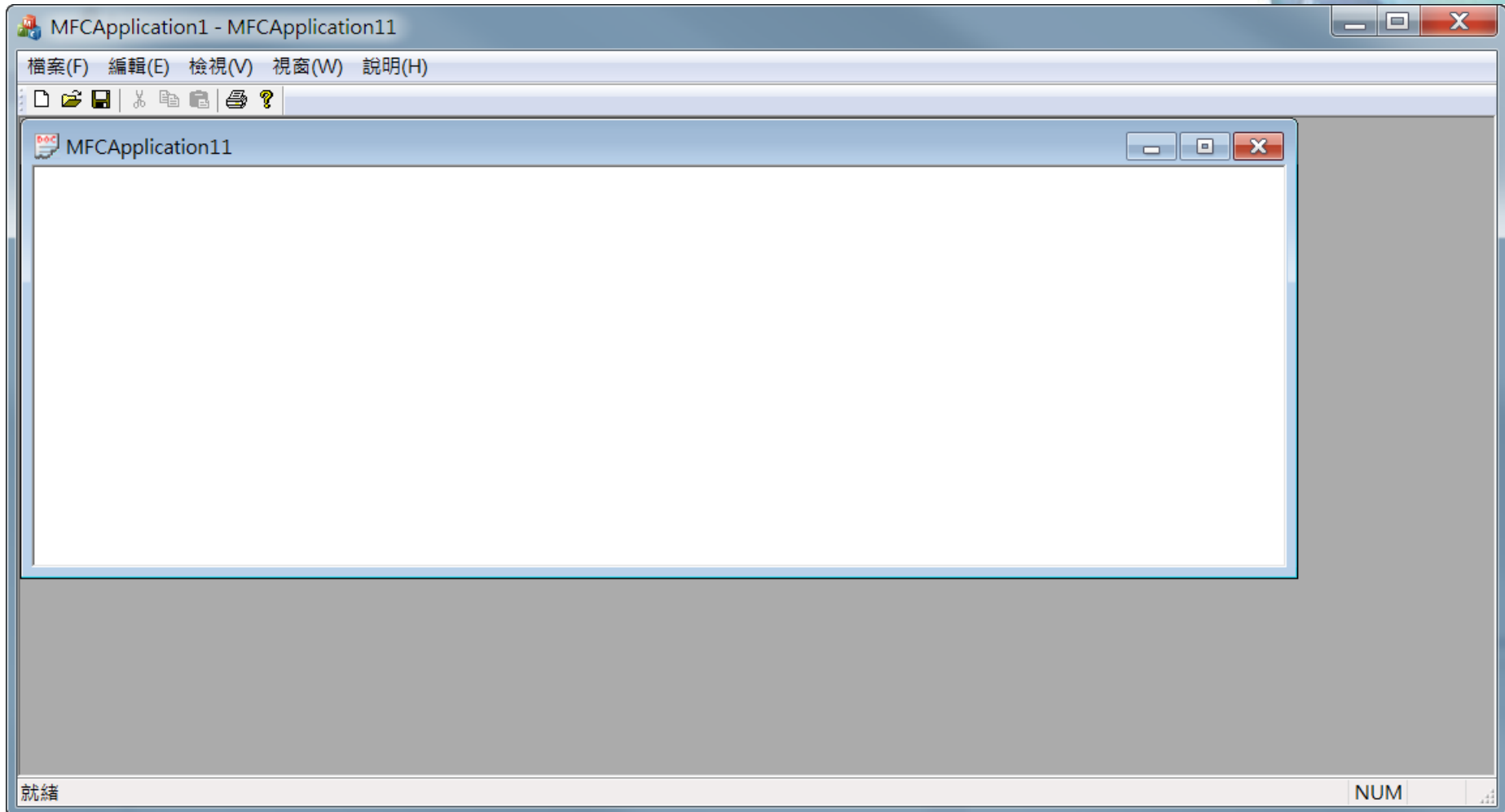
資源檢視 - MFCApplication1

- MFCApplication1
  - MFCApplication1.rc
    - Accelerator
      - IDR\_MAINFRAME
    - Dialog
      - IDD\_ABOUTBOX
    - Icon
      - IDR\_MAINFRAME
      - IDR\_MFCApplication1TYPE
    - Menu
      - IDR\_MAINFRAME
      - IDR\_MFCApplication1TYPE
    - String Table
      - String Table
    - Toolbar
      - IDR\_MAINFRAME
    - Version
      - VS\_VERSION\_INFO

方案總管 類別檢視 屬性管... 資源檢視 Team...



# MFC視窗





## stdafx.h

- 可將任一.h檔include至此
- 可讓其函式被任一函式或副程式使用
- 這樣便不用重覆的include

```
stdafx.h  ModuleView.cpp
(全域範圍)
#include <atlimage.h> //-----要加這個才能使用CImage
#include "Image.h"
#include <math.h>
#include <algorithm> //區段配置記憶體fill用的
```



# MFC應用於影像處理

---

- VC++ MFC影像檔簡介
- 讀檔
- 顯示影像檔
- 存檔
- 新增工具列按鈕
- 簡易影像處理範例
- 常見問題及處理方式





# VC++ MFC影像檔簡介

- 可使用CBitmap、CDibImage、Cimage等幾種點陣圖類型，此例為使用Cimage。
- CImage 本身為 ATL 之元件包成 MFC 的形式 讓我們方便使用
- CImage 支援大部分圖片的功能如圖檔的讀取及儲存
- 支援的檔案格式包括.jpg .bmp .gif .png
- 功能齊全不過這是VS.NET 以後才有的元件
- 使用前需Include CImage類的文件的atlimage.h後才能宣告CImage 類。



# VC++ MFC影像檔簡介

- **CImage**封裝了**DIB**(設備無關位圖)的功能，因而可以讓我們能夠處理每個位圖像素。它具有下列特性：
  1. **AlphaBlend**支持像素級的顏色混合，從而實現透明和半透明的效果。
  2. **PlgBlt**能使一個矩形區域的位圖映射到一個平行四邊形區域中，而且還可能使用位屏蔽操作。
  3. **TransparentBlt**在目標區域中產生透明圖像，**SetTransparentColor**用來設置某種顏色是透明色。
  4. **MaskBlt**在目標區域中產生源位圖與屏蔽位圖合成的效果。
- 5. 更詳細請參閱[MSDN](#)



# 建立額外.cpp與.h檔

方案總管

搜尋 方案總管 (Ctrl+;)

方案 'MFCApplication1' (1 專案)

- MFCApplication1
  - 原始程式檔
    - ChildF...
    - MainF...
    - MFCApp...
    - MFCApp...
    - MFCApp...
    - stdafx...
  - 外部相依性
  - 標頭檔
    - ChildF...
    - MainF...
    - MFCApp...
    - MFCApp...
    - MFCApp...
  - 資源檔
    - MFCApp...
    - MFCApp...
    - MFCApp...
    - MFCApplication1Doc.ico
    - Toolbar.bmp
    - ReadMe.txt

加入(D)

- 類別精靈(Z)... Ctrl+Shift+X
- 將範圍設定在此(S)
- 新增 方案總管 檢視(N)
- 剪下(T)
- 複製(Y)
- 貼上(P)
- 刪除(D)
- 重新命名(M)
- 屬性(R)

新增項目(W)... Ctrl+Shift+A

現有項目(G)... Shift+Alt+A

新的篩選條件(F)

類別(C)...

資源(R)...

偵錯

'MFCApplication1.exe' (Win32): 已載入 'C:\Windows\SysWow64\vd...

'MFCApplication1.exe' (Win32): 已卸載 'C:\Windows\SysWow64\vd...

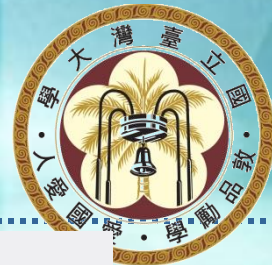
'MFCApplication1.exe' (Win32): 已載入 'C:\Windows\SysWow64\ole...

'MFCApplication1.exe' (Win32): 已卸載 'C:\Windows\SysWow64\ole...

0x76047A24 (user32.dll) (於 MFCApplication1.exe) 中第一個可能...

0x76047A24 (user32.dll) (於 MFCApplication1.exe) 中第一個可能...

'[146601 MFCApplication1.exe' 程式以返回碼 0 (0x0) 結束。



# 新增Image.h/.cpp

Image.h × Image.cpp MFCApplication1View.cpp

(全域範圍)

```
#ifndef IMAGE_H //防呆，防止重複引用
#define IMAGE_H

bool LoadImg(CImage *img, CString *title = NULL); //先預設函數就可以在呼叫時使用一個函數
bool SaveImg(CImage *img);

#endif
```

Image.h Image.cpp × MFCApplication1View.cpp

(全域範圍)

SaveImg(CImage \* img)

```
#include "stdafx.h"

bool LoadImg(CImage *img, CString *title)
{
    CString strFilter;
    CSimpleArray<GUID> aguidFileTypes;
    if (FAILED(img->GetExporterFilterString(strFilter, aguidFileTypes))) {
        return false;
    }
    CFileDialog dlg(TRUE, NULL, NULL, OFN_FILEMUSTEXIST, strFilter);

    if (dlg.DoModal() == IDOK) {
        img->Destroy();
        //if (FAILED(img->Load(dlg.GetFileName()))) { //2005 old version
```



# 於stdafx.h加入以下標頭檔

- #include <atlimage.h>
- #include "Image.h"
- #include <math.h>
- #include <algorithm>

```
stdafx.h  X Image.h Image.cpp MFCApplication1View.cpp
(全域範圍)
#endif
#ifndef _AFX_NO_AFXCMN_SUPPORT
#include <afxcmn.h> // MFC 支援的 Windows 通用控制項
#endif // _AFX_NO_AFXCMN_SUPPORT

#include <afxcontrolbars.h> // 功能區和控制列的 MFC 支援

#include <atlimage.h> //-----要加這個才能使用CImage
#include "Image.h"
#include <math.h>
#include <algorithm> //區段配置記憶體fill用的
```





## 於\*\*\*View.h中加入以下變數

- private:
- CImage OrgImg;
- CImage DispImg;
- CImage AnsImg;
- CImage Y;
- CImage U;
- CImage V;
- bool m\_bShowImage;//要不要顯示圖
- bool SetState;//要不要更新狀態
- CString CTitle;//Title 顯示的字

```
MFCApplication1View.h  x  stdafx.h  Image.h
CMFCApplication1View
virtual BOOL PreCreateWindow(CREATESTRUCT& cs)
protected:
virtual BOOL OnPreparePrinting(CPrintInfo* pInfo)
virtual void OnBeginPrinting(CDC* pDC, CPrintInfo* pInfo)
virtual void OnEndPrinting(CDC* pDC, CPrintInfo* pInfo)
private:
CImage OrgImg; //要 #include <atlimage.h>
CImage DispImg;
CImage AnsImg;
CImage Y;
CImage U;
CImage V;
bool m_bShowImage; //要不要顯示圖
bool SetState; //要不要更新狀態
CString CTitle; //Title 顯示的字
```



# 修改其建構式 加入初始值

- `m_bShowImage = false;`
- `SetState = false;`

```
MFCApplication1View.h  stdafx.h  Image.h  Image.cpp  MFCApplication1View.cpp*  X
→ CMFCApplication1View  CMFCApplication1View()
    ON_COMMAND(ID_FILE_OPEN, &CMFCApplication1View::OnFileOpen)
    END_MESSAGE_MAP()

// CMFCApplication1View 建構/解構

CMFCApplication1View::CMFCApplication1View()
{
    // TODO: 在此加入建構程式碼
    m_bShowImage = false;
    SetState = false;
}
```



# 修改OnDraw功能使其顯示

MFCApplication1View.h    stdafx.h    Image.h    Image.cpp    MFCApplication1View.cpp\*    OnDraw(CDC \* pDC)

// CMFCApplication1View 描繪

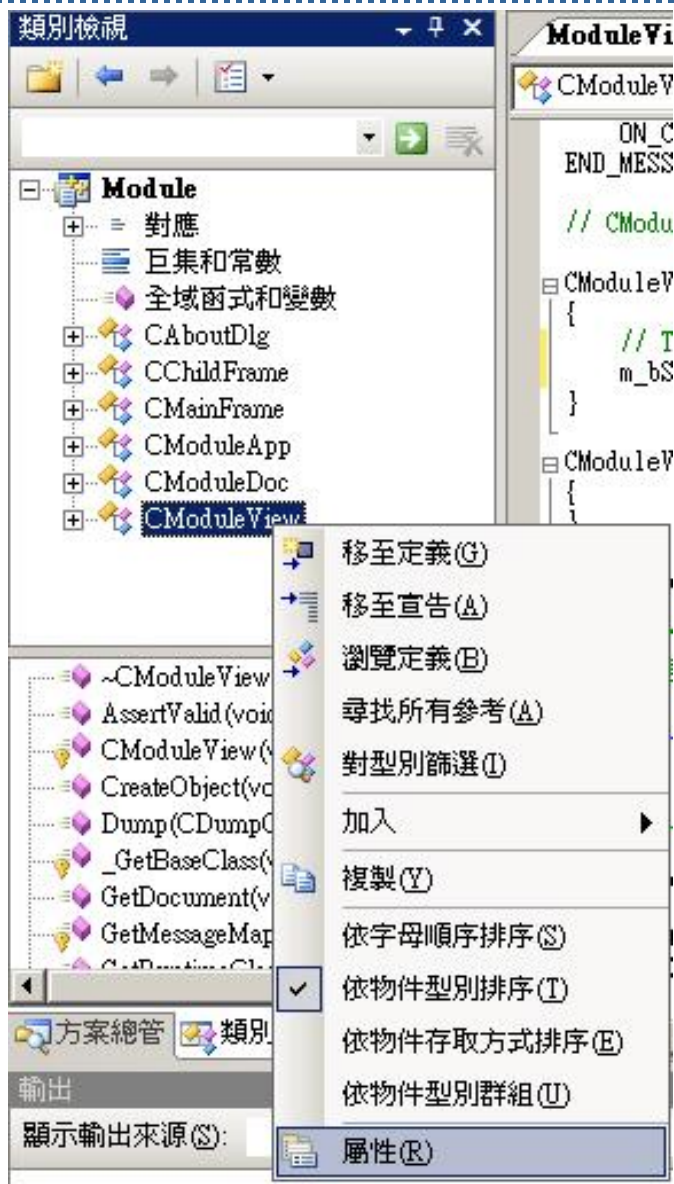
```
void CMFCApplication1View::OnDraw(CDC* pDC)
{
    CMFCApplication1Doc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if (!pDoc)
        return;
    // TODO: 在此加入原生資料的描繪程式碼
    if (m_bShowImage) {
        DispImg.Draw(*pDC, 0, 0);
    }
    if (SetState) {
        //改變視窗title
        CString CTitle("Title Test");
        LPCTSTR title = CTitle;
        pDoc->SetTitle(title);

        //改變子視窗大小
        CMDIChildWnd* childWnd = (CMDIChildWnd*)((CFrameWnd*)AfxGetApp()-> m_pMainWnd)-> GetActiveFrame();
        childWnd->SetWindowPos(&CWnd::wndTop, NULL, NULL, DispImg.GetWidth() + 22, DispImg.GetHeight() + 48, SWP_SHOWWINDOW);

        SetState = false;
    }
}
```



# 新增ON\_FILE\_OPEN COMMAND事件







# 讀檔

- 新增一 ON\_FILE\_OPEN COMMAND事件
- 加入以下程式碼

```
if ( LoadImg(&OrgImg, &CTitle) ) {  
    ImageCopy(&DispImg, &OrgImg);  
    m_bShowImage = true;  
    SetState = true;  
    Invalidate();  
} else  
    return;
```

MFCApplication1View.cpp\* X MFCApplication1View.h

(全域範圍)

```
void CMFCApplication1View::OnFileOpen()  
{  
    // TODO: 在此加入您的命令處理常式程式碼  
    if ( LoadImg(&OrgImg, &CTitle) ) {  
        ImageCopy(&DispImg, &OrgImg);  
        m_bShowImage = true;  
        SetState = true;  
        Invalidate();  
    } else  
        return;  
}
```



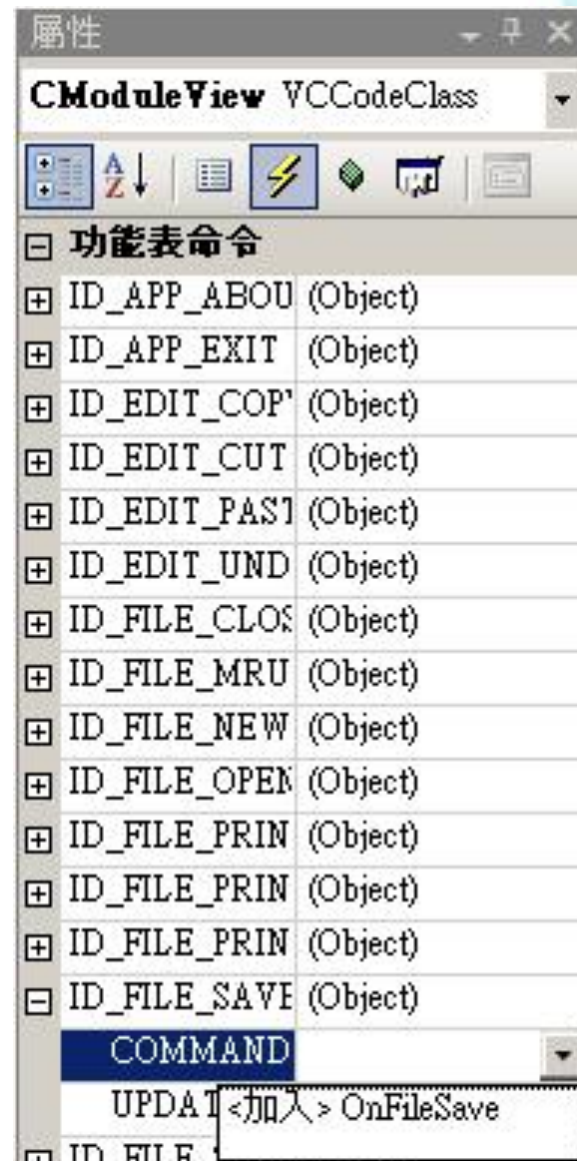
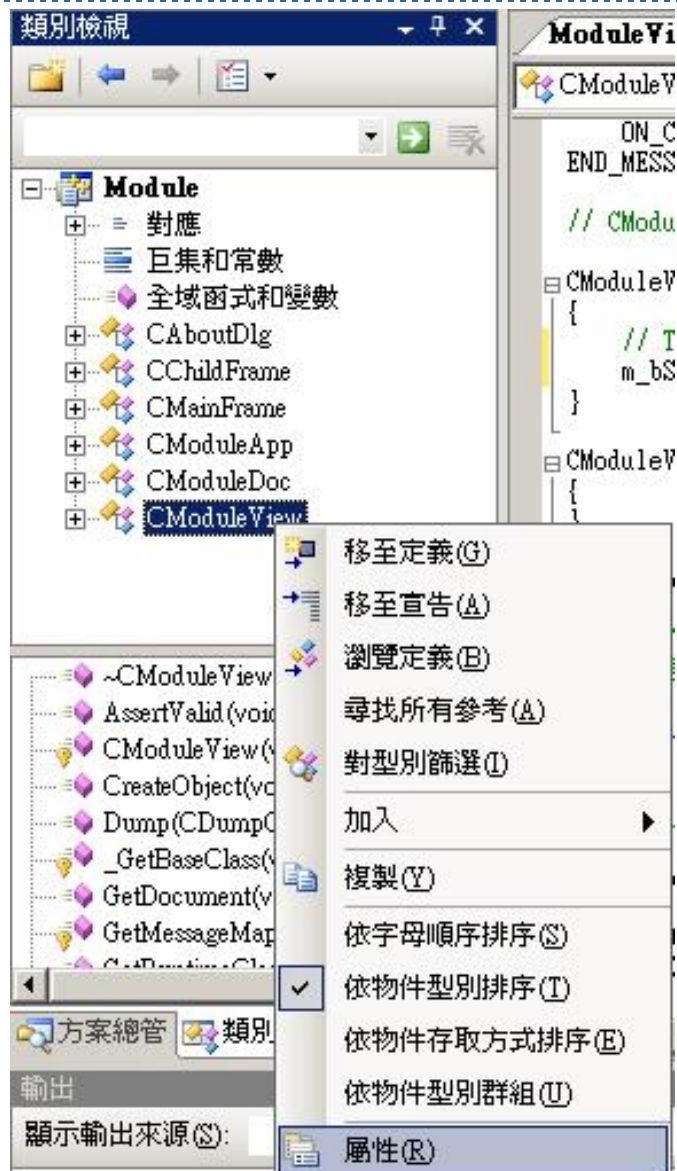


# 此時可以開啟圖檔了





# 新增ON\_FILE\_SAVE COMMAND事件





# 存檔

- 新增一 ON\_FILE\_SAVE COMMAND事件
- 加入以下程式碼

SaveImg(&DispImg);

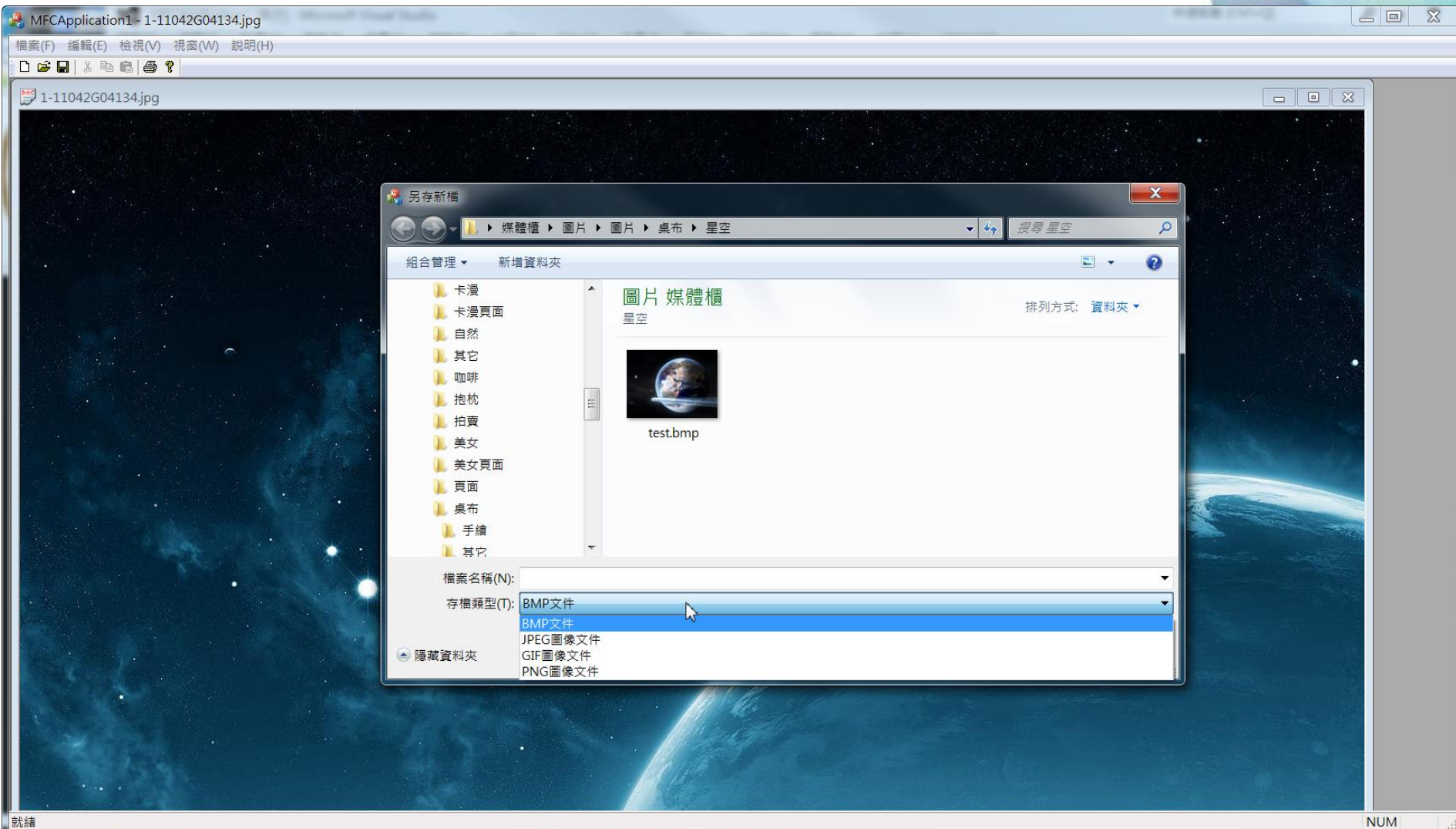
MFCApplication1View.cpp\*   MFCApplication1View.h\*

(全域範圍)

```
void CMFCApplication1View::OnFileSave()
{
    // TODO: 在此加入您的命令處理常式程式碼
    SaveImg(&DispImg);
}
```

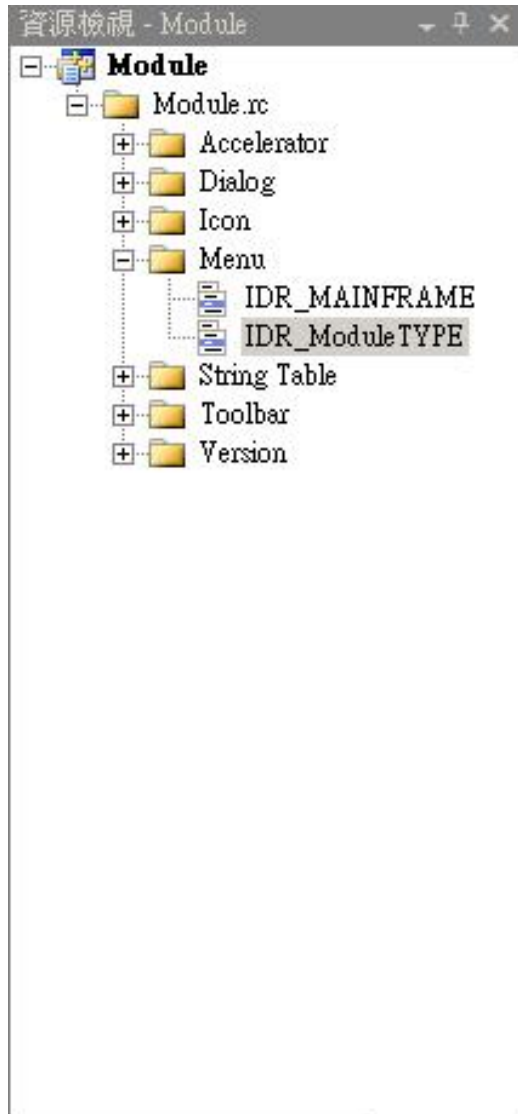


# 此時可以存檔了





# 新增工具列按鈕








# 新增工具列按鈕





# 新增工具列按鈕

事件處理常式精靈 - Module

 歡迎使用事件處理常式精靈

命令名稱:  
ID\_PROCESS\_IMAGEMEMORY

訊息類型(Y):  
COMMAND  
UPDATE\_COMMAND\_UI

函式處理常式名稱(N):  
OnProcessImagememory

處理常式描述:  
在選擇功能表項目或命令按鈕後呼叫

類別清單(L):  
CAboutDlg  
CChildFrame  
CMainFrame  
CModuleApp  
CModuleDoc  
CModuleView

```
void CModuleView::OnProcessImagememory()  
{  
    // TODO: 在此加入您的命令處理常式程式碼  
}
```

加入並編輯(A) 取消



# 簡易影像處理範例

```
int h = DispImg.GetHeight();
int w = DispImg.GetWidth();
COLORREF color;
BYTE r, g, b, gray;

for (int i = 0; i < h; i++) {
    for (int j = 0; j < w; j++) {
        color = DispImg.GetPixel(j, i);

        r=GetRValue(color);
        g=GetGValue(color);
        b=GetBValue(color);
        gray = r*0.299 + g*0.587 + b*0.114;

        //DispImg.SetPixel( j, i, RGB(r, g, b) );
        DispImg.SetPixel( j, i, RGB(gray, gray, gray) );
    }
}
Invalidate();
```



# 小練習

---

- 建置影像處理專案
- 做基本轉灰階練習



# 常見問題及處理方式

---

- 使用CImage內建GetPixel(x, y)與SetPixel(x, y)讀寫函式的速度過慢
- 多核心運算
- 陣列過大造成Stack Over Flow。
- 其它改善程式效率的方法
  - 動態陣列
  - 加速陣列初始化





# CImage內建讀寫函式的速度過慢

- 問題：CImage內建GetPixel(x, y)與SetPixel(x, y)讀寫函式的速度過慢
- 解決：使用以下函式，直接對記憶體存取

```
○ BYTE** Create2DList(CImage *pImg)
○ {
○     int i;
○     BYTE **list;

○     list=(BYTE**)malloc(pImg->GetHeight*sizeof(BYTE*));
○     for (i=0;i<pImg->GetHeight;i++)
○         list[i]=(BYTE*) pImg->GetPixelAddress(0,i);

○     return(list);
○ }
○ void Release2DList(BYTE** list)
○ {
○     free(list);
○ }
```



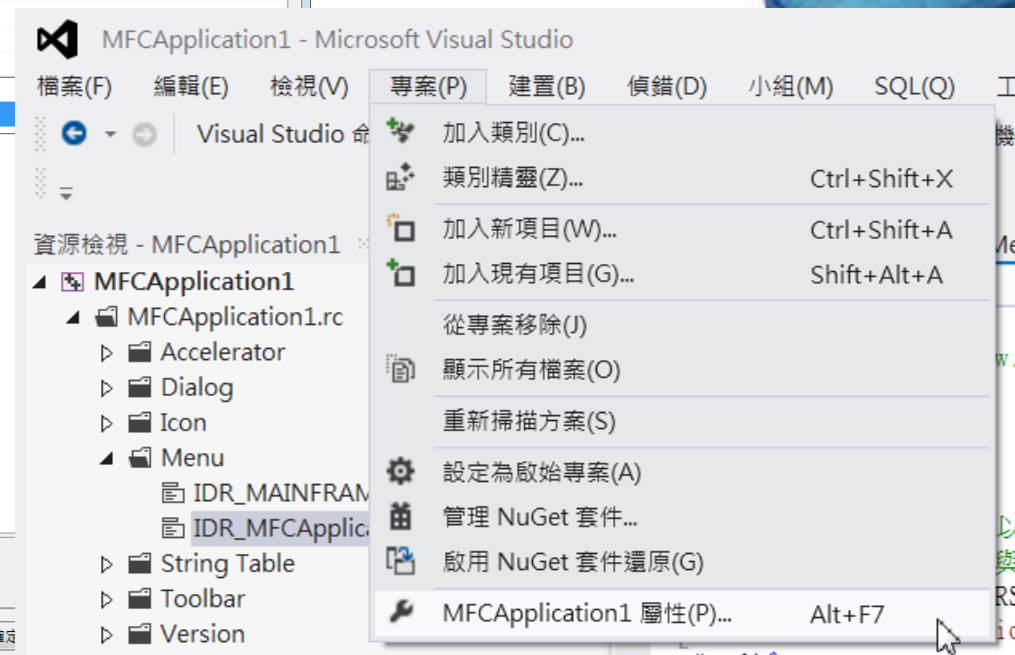
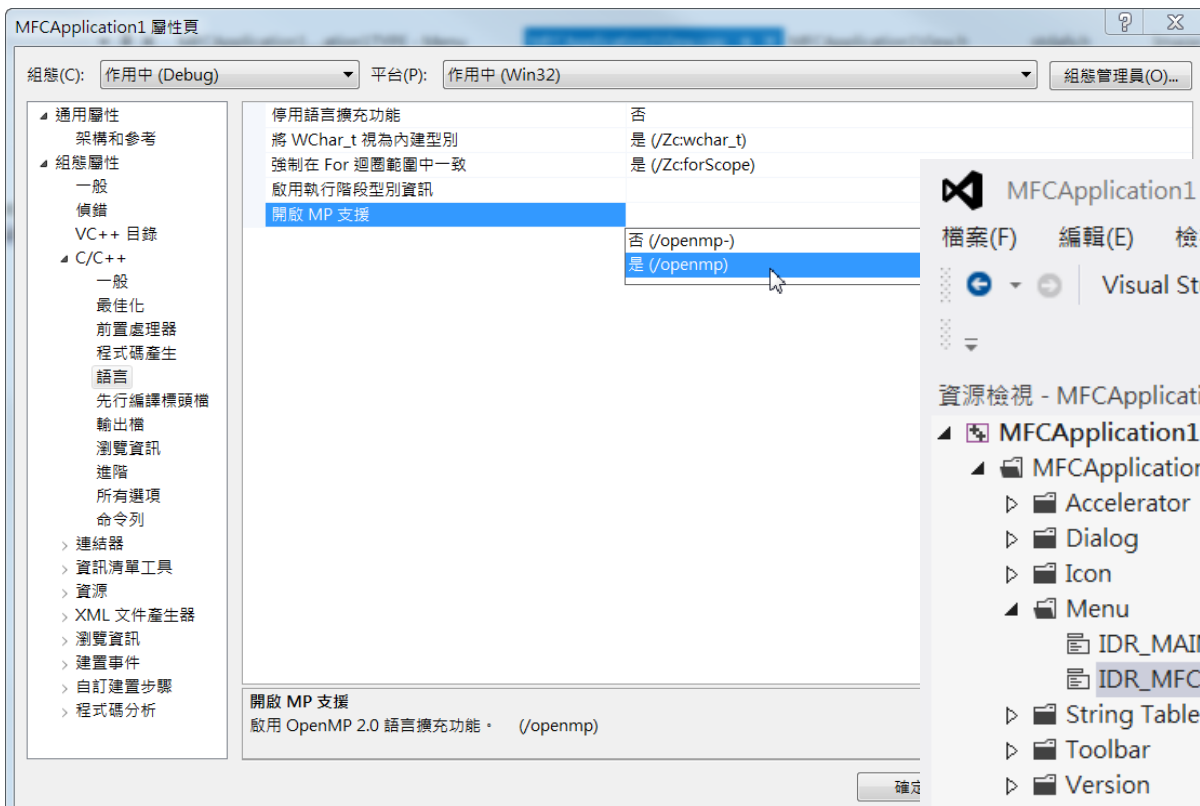
# CImage內建讀寫函式的速度過慢

- 使用方式
  - Cimage DispImg;
  - BYTE \*\*aout;
  - aout=(BYTE\*\*) Create2DList(&DispImg);
  - aout[x][y] = .....;
  - .....;//處理
  - Release2DList(aout);//用完記得要釋放
  - DispImg.Destory();//用完記得要釋放



# 開啟多核心運算

- 加入 `#include <omp.h>`
- 在 `for` 迴圈前加上一行 `#pragma omp parallel for`





# OpenMP不能平行處理之情形

- 變數有先後使用的串連
- OpenMP平行處理後會得到錯誤的結果

```
int FList [10];  
FList [0] = 0;  
FList [1] = 1;  
#pragma omp parallel for num_threads(4)  
for( a = 2; a < 10; ++ a )  
    FList [a] = FList [a-1] + FList [a-2];  
for( a = 0; a < 10; ++ a )  
    printf( "%d," , FList [a] );
```

- 原本的結果是 0,1,1,2,3,5,8,13,21,34
- 變成0,1,1,2,3,5,-1823049,-1878930,-4,-1717986912





# Stack Over Flow 的處理

- 也可將影像轉入陣列也可提升速度
- 但VC++的local很容易Over Flow而過不了Compiler
- Sol 1:把變數放到global裡，但不加const的全域變數是非常不被鼓勵的寫法。
- Sol 2:使用dynamic array
- `bool prime[20000000]; //這個會爆`  
`bool* prime = new bool[20000000]; //這個不會`



# 動態陣列-配置陣列格式:

– 變數型別 \*指標名 = new 變數型別[陣列長度]; //一維陣列

## ○ //多維陣列

– 變數型別 (\*指標名)[陣列長度2] = new 變數型別[陣列長度1][陣列長度2];

– 變數型別 (\*指標名)[陣列長度2][陣列長度3] = new 變數型別[陣列長度1][陣列長度2][陣列長度3];

## ○ 例:

– int \*p3 = new int[10];

– double (\*p4)[4] = new double[3][4];

– double (\*p5)[4][5] = new double[3][4][5];

delete ary; ary = NULL;

○ 手動分配出來的記憶體，請自己delete掉。

○ 把delete的pointer指回NULL

○ 這就是所謂的小三元



# 加速陣列初始化(舊版)

- 初始化陣列時用**for**迴圈是相當慢的方式
- C++有內建的函式可快速的初始化
- `#include <algorithm>`
- `char *p = new char[100];`
- `std::fill(p, p+100, '\0');` // zero all characters



# 動態陣列應用

```
int *his = new int[w]; memset(his , 0, w*sizeof(int) );
```

```
double **TDB_num = new double*[h];  
for(int i=0; i<h; i++) {  
    TDB_num[i] = new double[w];  
    memset( TDB_num[i], 0, w*sizeof(double) );  
}
```

```
double ***Mean1 = new double**[d1];  
for(int j=0; j<d1; j++) {  
    Mean1[j] = new double*[mh];  
    for(int i=0; i<mh; i++) {  
        Mean1[j][i] = new double[mw];  
        memset( Mean1[j][i], 0, mw*sizeof(double) );  
    }  
}  
delete []his;
```





# 小練習

---

- 將影像放進動態陣列
- 練習使用直接對記憶體存取法





Module - Microsoft Visual Studio (系統管理員)

檔案(F) 編輯(E) 檢視(V) 專案(P) 建置(B) 偵錯(D) 小組(M) 資料(A) 格式(O) 工具(T) 測試(S) 視窗(W) 說明(H)

Debug Win32 load

資源檢視 - Module

- Module
  - Module.rc\*
    - Accelerator
      - Dialog
        - IDD\_ABOUTBOX
        - IDD\_DIALOG1
        - IDD\_DIALOG2
      - Icon
      - Menu
        - IDR\_MAINFRAME
        - IDR\_ModuleTYPE
      - String Table
      - Toolbar
        - IDR\_MAINFRAME
      - Version

Module.rc - IDD\_DIALOG2 - Dialog\* x ModuleView.cpp Dialog1.h Module.rc - IDD\_DIALOG1 - Dialog\* Image.c

對話方塊編輯器

- 指標
- Button
- Check Box
- ab Edit Control
- Combo Box
- List Box
- Group Box
- Radio Button
- Static Text
- Picture Control
- Horizontal Scroll Bar
- Vertical Scroll Bar
- Slider Control
- Spin Control
- Progress Control
- Hot Key
- List Control
- Tree Control
- Tab Control
- Animation Control
- Rich Edit 2.0 Control
- Date Time Picker
- Month Calendar Control
- IP Address Control
- Extended Combo Box
- Custom Control
- SysLink Control
- Split Button Control
- Network Address Control
- Command Button Control
- ab MFC Button Control

確定 取消

原型影像

透明度: 50% 位移 X: 0 Y: 0

輸出

顯示輸出來源(S): 偵錯

Detected memory leaks!  
Dumping objects ->  
c:\users\shiuny\documents\visual studio 2010\projects\digital image process hw\hw1\module 01\_2010\module\moduleview.cpp  
Data: < 0 R > 00 00 00 00 0C 00 00 00 30 00 00 00 52 00 00 00  
c:\users\shiuny\documents\visual studio 2010\projects\digital image process hw\hw1\module 01\_2010\module\moduleview.cpp  
Data: < i a > 00 00 00 00 84 00 00 00 69 01 00 00 61 01 00 00  
Object dump complete.  
'[B440] Module.exe: 原生' 程式以返回碼 0 (0x0) 結束。

方 琴 風 資 T. 程式碼定義視窗 輸出

就緒



Module - Microsoft Visual Studio (系統管理員)

檔案(F) 編輯(E) 檢視(V) 專案(P) 建置(B) 偵錯(D) 小組(M) 資料(A) 格式(O) 工具(T) 測試(S) 視窗(W) 說明(H)

Debug Win32 load

資源檢視 - Module

- Module
  - Module.rc\*
  - Accelerator
  - Dialog
    - IDD\_ABOUTBOX
    - IDD\_DIALOG1
    - IDD\_DIALOG2
  - Icon
  - Menu
    - IDR\_MAINFRAME
    - IDR\_ModuleTYPE
  - String Table
  - Toolbar
    - IDR\_MAINFRAME
  - Version

Module.rc - IDD\_DIALOG2 - Dialog\*

Dialog

Static

範例編輯

確定

MFC 加入類別精靈 - MFCApplication1

歡迎使用 MFC 加入類別精靈

名稱: Dialog1

類別名稱(L): Dialog1

基底類別(B): CDialogEx

對話方塊 ID(D): IDD\_DIALOG2

.h 檔(I): Dialog1.h

.cpp 檔(P): Dialog1.cpp

Active Accessibility(Y)

DHTML 資源 ID(S): IDR\_HTML\_DIALOG1

.HTM 檔(M): Dialog1.htm

Automation:

- ☒ 無(N)
- ☐ Automation(A)
- ☐ 可由型別 ID 建立(E)

型別 ID(T): MFCApplication1.Dialog1

☐ 產生 DocTemplate 資源(G)

輸出

顯示輸出來源(S): 偵錯

Detected memory leaks!

Dumping objects ->

c:\users\shiny\documents\visual studio 2010\projects\digital image process hw\hw1\module 01\_2010\module\moduleview.cpp(342): {687

Data: < 0 R > 00 00 00 00 00 00 00 30 00 00 00 52

c:\users\shiny\documents\visual studio 2010\projects\digital image process hw\hw1\module 01\_2010\module\moduleview.cpp(342): {687

Data: < i a > 00 00 00 00 84 00 00 00 69 01 00 00 61 01 00 00

Object dump complete.

'[8440] Module.exe: 原生' 程式以返回碼 0 (0x0) 結束。

ID	IDD_DIALOG2
Local Edit	False
Menu	

ID

指定對話方塊資源的識別項。

0,0 202 x 162





Module - Microsoft Visual Studio (系統管理員)

檔案(F) 編輯(E) 檢視(V) 專案(P) 建置(B) 偵錯(D) 小組(M) 資料(A) 格式(O) 工具(T) 測試(S) 視窗(W) 說明(H)

Debug Win32 load

資源檢視 - Module

- Module
  - Module.rc\*
  - Accelerator
    - Dialog
      - IDD\_ABOUTBOX
      - IDD\_DIALOG1
      - IDD\_DIALOG2
    - Icon
    - Menu
      - IDR\_MAINFRAME
      - IDR\_ModuleTYPE
    - String Table
    - Toolbar
      - IDR\_MAINFRAME
    - Version

Module.rc - IDD\_DIALOG2 - Dialog\* x ModuleView.cpp Dialog1.h Module.rc - IDD\_DIALOG1 - Dialog\* Image.c

對話方塊編輯器

指標

- Button
- Check Box
- Edit Control
- Combo Box
- List Box
- Group Box
- Radio Button
- Static Text
- Picture Control
- Horizontal Scroll Bar
- Vertical Scroll Bar
- Slider Control
- Spin Control
- Progress Control
- Hot Key
- List Control
- Tree Control
- Tab Control
- Animation Control
- Rich Edit 2.0 Control
- Date Time Picker
- Month Calendar Control
- IP Address Control
- Extended Combo Box
- Custom Control
- SysLink Control
- Split Button Control
- Network Address Control
- Command Button Control
- MFC Button Control

原型影像

透明度: 50% 位移 X: 0 Y: 0

輸出

顯示輸出來源(S): 偵錯

Detected memory leaks!  
Dumping objects ->  
c:\users\shiuny\documents\visual studio 2010\projects\digital image process hw\hw1\module 01\_2010\module\moduleview.cpp  
Data: < 0 R > 00 00 00 00 0C 00 00 00 30 00 00 00 52 00 00 00  
c:\users\shiuny\documents\visual studio 2010\projects\digital image process hw\hw1\module 01\_2010\module\moduleview.cpp  
Data: < i a > 00 00 00 00 84 00 00 00 69 01 00 00 61 01 00 00  
Object dump complete.  
'[8440] Module.exe: 原生' 程式以返回碼 0 (0x0) 結束。

就緒



Module - Microsoft Visual Studio (系統管理員)

檔案(F) 編輯(E) 檢視(V) 專案(P) 建置(B) 偵錯(D) 小組(M) 資料(A) 格式(O) 工具(T) 測試(S) 視窗(W) 說明(H)

Debug Win32 load

資源檢視 - Module

- Module
  - Module.rc\*
    - Accelerator
    - Dialog
      - IDD\_ABOUTBOX
      - IDD\_DIALOG1
      - IDD\_DIALOG2
    - Icon
    - Menu
      - IDR\_MAINFRAME
      - IDR\_ModuleTYPE
    - String Table
    - Toolbar
      - IDR\_MAINFRAME
    - Version

Module.rc - IDD\_DIALOG2 - Dialog\*

Module\view.cpp Dialog1.h Module.rc - IDD\_DIALOG1 - Dialog\* Image.cpp

屬性

IDC\_STATIC1 (Text Control) IDC\_STATIC1

外觀

Align Text	Left
Border	False
Caption	Static
Center Image	False
Client Edge	False
End Ellipsis	False
Modal Frame	False
No Prefix	False
No Wrap	False
Notify	False
Path Ellipsis	False
Real Size Co	False
Right Align T	False
Right To Left	False
Simple	False
Static Edge	False
Sunken	False
Transparent	False
Word Ellipsis	False

行為

Accept Files	False
Disabled	False
Help ID	False
Visible	True

其他

(Name)	IDC_STATIC1 (T
Group	True
ID	IDC_STATIC

(Name)

輸出

顯示輸出來源(S): 偵錯

Detected memory leaks!  
Dumping objects ->  
c:\users\shiuny\documents\visual studio 2010\projects\digital image process hw\hw1\module 01\_2010\module\moduleview.cpp(346) : {68E  
Data: < 0 R > 00 00 00 00 0C 00 00 00 30 00 00 00 52 00 00 00  
c:\users\shiuny\documents\visual studio 2010\projects\digital image process hw\hw1\module 01\_2010\module\moduleview.cpp(342) : {687  
Data: < i a > 00 00 00 00 84 00 00 00 69 01 00 00 61 01 00 00  
Object dump complete.  
'[8440] Module.exe: 原生' 程式以返回碼 0 (0x0) 結束。

請按 ALT 鍵停用控制項位置上的輔助線和界限。

7,7 126x87



Module - Microsoft Visual Studio (系統管理員)

檔案(F) 編輯(E) 檢視(V) 專案(P) 建置(B) 偵錯(D) 小組(M) 資料(A) 格式(O) 工具(T) 測試(S) 視窗(W) 說明(H)

Debug Win32 load

資源檢視 - Module

- Module
  - Module.rc\*
    - Accelerator
    - Dialog
      - IDC\_ABOUTBOX
      - IDC\_DIALOG1
      - IDC\_DIALOG2
    - Icon
    - Menu
      - IDR\_MAINFRAME
      - IDR\_ModuleTYPE
    - String Table
    - Toolbar
      - IDR\_MAINFRAME
    - Version

Module.rc - IDD\_DIALOG2 - Dialog\* | ModuleView.cpp | Dialog1.h | Module.rc - IDD\_DIALOG1 - Dialog\* | Image.cpp

Static

範例編輯 確定 取消

屬性 IDC\_STATIC1 (Text Control) Iste

Border	False
Caption	Static
Center Image	False
Client Edge	False
End Ellipsis	False
Model Fram	False
No Prefix	False
No Wrap	False
Notify	False
Path Ellipsis	False
Real Size Co	False
Right Align T	False
Right To Left	False
Simple	False
Static Edge	False
Sunken	False
Transparent	False
Word Ellipsis	False

行為

Accept Files	False
Disabled	False
Help ID	False
Visible	True

其他

(Name)	IDC_STATIC1 (T
Group	True
ID	IDC_STATIC
Tabstop	False

ID 指定控制項的識別項。

輸出

顯示輸出來源: 偵錯

```
Detected memory leaks!
Dumping objects ->
c:\users\shiuny\documents\visual studio 2010\projects\digital image process hw\hw1\module 01_2010\module\moduleview.cpp(346) : {68E
Data: < 0 R > 00 00 00 00 0C 00 00 00 30 00 00 00 52 00 00 00
c:\users\shiuny\documents\visual studio 2010\projects\digital image process hw\hw1\module 01_2010\module\moduleview.cpp(342) : {687
Data: < i a > 00 00 00 00 84 00 00 00 69 01 00 00 61 01 00 00
Object dump complete.
'[8440] Module.exe: 原生' 程式以返回碼 0 (0x0) 結束。
```

就緒



Module - Microsoft Visual Studio (系統管理員)

檔案(F) 編輯(E) 檢視(V) 專案(P) 建置(B) 偵錯(D) 小組(M) 資料(A) 格式(O) 工具(T) 測試(S) 視窗(W) 說明(H)

Debug Win32 load

資源檢視 - Module

- Module
- Module.rc\*
- Accelerator
- Dialog
  - IDD\_ABOUTBOX
  - IDD\_DIALOG1
  - IDD\_DIALOG2
- Icon
- Menu
  - IDR\_MAINFRAME
  - IDR\_ModuleTYPE
- String Table
- Toolbar
  - IDR\_MAINFRAME
- Version

Module.rc - IDD\_DIALOG2 - Dialog\*

Dialog

Static

範例編輯器 確定

加入變數(B)

輸出

顯示輸出來源(S): 偵錯

Detected memory leaks!

Dumping objects ->

c:\users\shiumy\documents\visual studio 2010\projects\digital image process hw\hw1\module 01\_2010\module\moduleview.cpp(346) : {68E...}

Data: < 0 R > 00 00 00 00 0C 00 00 00 30 00 00 00 52 00 00 00

c:\users\shiumy\documents\visual studio 2010\projects\digital image process hw\hw1\module 01\_2010\module\moduleview.cpp(342) : {687...}

Data: < i a > 00 00 00 00 84 00 00 00 69 01 00 00 61 01 00 00

Object dump complete.

'[8440] Module.exe: 原生' 程式以返回碼 0 (0x0) 結束。

屬性

IDC\_STATIC1 (Text Control) ISt...

Border	False
Caption	Static
Center Image	False
Client Edge	False
End Ellipsis	False
Modal Frame	False
No Prefix	False
No Wrap	False
Notify	False
Path Ellipsis	False
Real Size Co	False
Right Align T	False
Right To Left	False
Simple	False
Static Edge	False
Sunken	False
Transparent	False
Word Ellipsis	False

行為

Accept Files	False
Disabled	False
Help ID	False
Visible	True

其他

(Name)	IDC_STATIC1 (T
Group	True
ID	IDC_STATIC
Tabstop	False

ID

指定控制項的識別項。





加入成員變數精靈 - Module

歡迎使用加入成員變數精靈

加入成員變數精靈 - MFCApplication1

歡迎使用加入成員變數精靈

存取(A): public

變數型別(V): CString

變數名稱(N): Text1

註解 (不需要輸入 //)

存取(A): public

變數型別(V): double

變數名稱(N): input

註解 (不需要輸入 //)(M):

☒ 控制項變數(O)

控制項 ID(I): IDC\_EDIT1

控制項型別(Y): EDIT

最小值(U):

.h 檔(F):

類別(T): Value

數(C):

最大值(E):

.cpp 檔(P):

完成 取消



# 使用dialog

- 加入#include "Dialog的名稱.h"
  - Ex: #include "Dialog1.h"
- 若出現error C2065: 'IDD\_DIALOG1': 未宣告的識別項請在Dialog1.h中加入#include "resource.h"

- 使用：

```
Dialog1 dlg1;
```

```
dlg1.input = ZoomT;
```

```
dlg1.Text1 = "Please input Zooming times for Height";
```

```
if (dlg1.DoModal() == IDOK) {
```

```
    ZoomT = dlg1.input;
```

```
    zh = h*ZoomT;
```

```
}
```



# 小練習

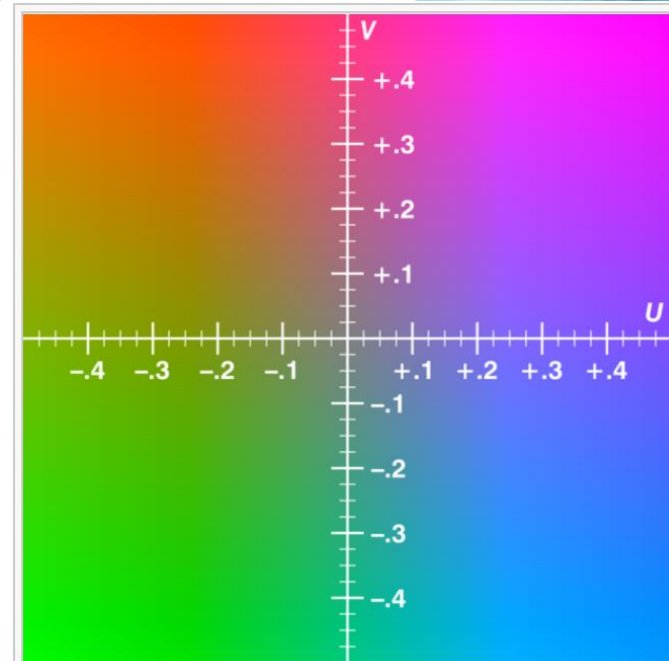
---

- 新增一輸入方塊
- 使用縮放範例



# RGB & YUV

- YUV是編譯true-color顏色空間（color space）的種類
- 「Y」表示明亮度（Luminance、Luma）
- 「U」是色度（Chrominance）
- 「V」則是濃度（Chroma）



U-V color plane範例 · Y value = 0.5 · 代表RGB色域 (color gamut)



# RGB & YUV

- YUV與RGB的轉換公式：

$$Y = 0.299 * R + 0.587 * G + 0.114 * B$$

$$U = 0.436 * (B - Y) / (1 - 0.114) + 128$$

$$V = 0.615 * (R - Y) / (1 - 0.299) + 128$$

- 在逆轉關係上，從YUV到RGB，可得

$$R = Y + 1.13983 * (V - 128)$$

$$G = Y - 0.39465 * (U - 128) - 0.58060 * (V - 128)$$

$$B = Y + 2.03211 * (U - 128)$$





# RGB & YUV

```
#define CLIP(X) ( (X) > 255 ? 255 : (X) < 0 ? 0 : X)
```

```
// RGB -> YUV
```

```
#define RGB2Y(R, G, B) CLIP(( ( 66 * (R) + 129 * (G) + 25 * (B) + 128) >> 8) + 16)
```

```
#define RGB2U(R, G, B) CLIP(( ( -38 * (R) - 74 * (G) + 112 * (B) + 128) >> 8) + 128)
```

```
#define RGB2V(R, G, B) CLIP(( ( 112 * (R) - 94 * (G) - 18 * (B) + 128) >> 8) + 128)
```

```
// YUV -> RGB
```

```
#define C(Y) ( (Y) - 16 )
```

```
#define D(U) ( (U) - 128 )
```

```
#define E(V) ( (V) - 128 )
```

```
#define YUV2R(Y, U, V) CLIP(( 298 * C(Y) + 409 * E(V) + 128) >> 8)
```

```
#define YUV2G(Y, U, V) CLIP(( 298 * C(Y) - 100 * D(U) - 208 * E(V) + 128) >> 8)
```

```
#define YUV2B(Y, U, V) CLIP(( 298 * C(Y) + 516 * D(U) + 128) >> 8)
```



# MessageBox

---

```
CString fmt;  
fmt = "SNR of image = " + ;  
::AfxMessageBox(fmt);
```

```
CString error("These image are not same type");::AfxMessageBox(error);  
  
::AfxMessageBox(_T("The result threshold T") );
```

```
CString fmt;  
fmt.Format(_T("Heigh = %d \nWidth = %d \nBytesPerPixel = %d "), P.nHeight,  
P.nWidth, P.nBytesPerPixel);  
::AfxMessageBox(fmt);
```



**THANKS FOR YOUR ATTENTION!**