Tìm hiều cơ bản về MFC

# HWND

## Window là gì?

* Trong ứng dụng giao diện đồ họa trên hệ điều hành Windows thì window là vùng hình chữ nhật của màn hình. Window cho phép ứng dụng hiển thị output đến người dùng và nhận input từ người dùng.
* Window bao gồm : cửa sổ, dialog, button,…

## Handle là gì?

Là một số nguyên 32 bit không dấu do HĐH tạo ra dùng làm định danh cho các đối tượng (window, file, vùng nhớ, menu,…).

## HWND là gì?

* Là một handle của một window
* Mọi thao tác chúng ta muốn thực hiện với window (move, show, destroy, chỉnh sửa style, thêm control,…) đều thực hiện thông qua HWND của cửa sổ đó.

# MFC Hello World Program

**Em bắt đầu tìm hiểu MFC xung quanh ứng dụng này**

**Hello.h**

class CMyApp : public CWinApp

{

public:

virtual BOOL InitInstance ();

};

class CMainWindow : public CFrameWnd

{

public:

CMainWindow ();

protected:

afx\_msg void OnPaint ();

DECLARE\_MESSAGE\_MAP ()

};

**Hello.cpp**

#include <afxwin.h>

#include "Hello.h"

CMyApp myApp;

/////////////////////////////////////////////////////////////////////////

// CMyApp member functions

BOOL CMyApp::InitInstance ()

{

m\_pMainWnd = new CMainWindow;

m\_pMainWnd->ShowWindow (m\_nCmdShow);

m\_pMainWnd->UpdateWindow ();

return TRUE;

}

/////////////////////////////////////////////////////////////////////////

// CMainWindow message map and member functions

BEGIN\_MESSAGE\_MAP (CMainWindow, CFrameWnd)

ON\_WM\_PAINT ()

END\_MESSAGE\_MAP ()

CMainWindow::CMainWindow ()

{

Create (NULL, \_T ("The Hello Application"));

}

void CMainWindow::OnPaint ()

{

CPaintDC dc (this);

CRect rect;

GetClientRect (&rect);

dc.DrawText (\_T ("Hello, MFC"), -1, &rect,

DT\_SINGLELINE | DT\_CENTER | DT\_VCENTER);

}

# CWinApp

* Trái tim của một ứng dụng MFC đó chính là application object được kế thừa từ lớp CWinApp. CWinApp cung cấp một message loop có nhiệm vụ nhận thông điệp và gửi thông điệp đó đến cửa sổ của ứng dụng.
* Một ứng dụng chỉ có thể có một và chỉ applicaton object và đối tượng này phải được khai báo là biến toàn cục để nó có thể được khởi tạo ngay khi chương trình được bắt đầu.
* Bất kì ứng dụng nào cùng phải override lại phương thức InitInstance của lớp CWinAPP. Phương thức InitInstance này được gọi ngay khi chương trình được khởi động vào trước khi cửa sổ được khởi tạo. Trong hàm InitInstance ta phải chỉ định cửa sổ được tạo. Nếu phương thức InitInstance thì ứng dụng sẽ bị hủy.
* Một số phương thức có thể override của lớp CWinAPP
* *ExitInstance*
* Cho phép dọn dẹp hệ thống trước khi ứng dụng bị hủy. Nếu trong InitInstance có thực hiện cấp vùng nhớ cho tài nguyên nào đó thì ExitInstance là cơ hội tốt nhât để giải phóng vùng nhớ đó
* Mặc định trong ExitInstance thực hiện dọn dẹp một số tài nguyên đc yêu cầu bởi framework. Do đó khi override phương thức này thì phải đảm bảo phương thức ExitInstance của lớp chae cũng được gọi. Giá trị trả vể của ExitInstance là mã thoát của WinMain
* *OnIdleOnIdle*
* Được gọi khi ứng dụng ở trạng tháng idel
* Hay được sử dụng để thực hiện các tác vụ chạy nền ví dụ như thu dọn rác.
* *Run*

Override phương thức này để thay đổi message loop mặc định bằng message loop do người lập trình tự định nghĩa

* *PreTranslateMessage*

Được sử dụng khi có nhu cầu thực hiện tiền xử lý với một số loại message cụ thể nào đó trước khi nó được gửi đến cửa sổ.

# CWnd

Là một class trong MFC mà bên trong có chứa thuộc tính là HWND. CWnd cung cấp các interface cho phép tạo cửa số, nhận và xử lý các thông điệp.

Các đối tượng cửa sổ đều phải kế thừa từ lớp này. Nhưng không kế thừa trực tiếp mà kế thừa thông qua lớp con của nó. Ứng với mỗi loại cửa sổ chuyên biệt thì CWnd có các lớp con như sau : Frame Window, MDI Frame Window, MDI Child Window, View, Dialog. Riêng Button thì kế thừa trực tiếp từ lớp CWnd

## **CFrameWindow**

* Frame window là cửa sổ chính của ứng dụng, được hiển thị đầu tiên khi ứng dụng được chạy
* Tạo cửa sổ chính của ứng dụng

1. Khai báo một class kế thừa từ lớp CFrameWindow.
2. Trong phương thức khởi tạo của class này thực hiện chỉ định các thuộc tính cần thiết cho cửa sổ
3. Khai báo các phương thức xử lý thông điệp và message map để cho biết thông điệp nào sẽ được phương thức nào xử lý.

* Không dùng toán tử delete để hủy đối tượng frame window. Thay vào đó phải dùng phương thức CFrameWindow::DestroyWindow

## **WM\_PAINT**

Tượng tự như trong Win32 mỗi khi cần vẽ lại cửa sổ thì thông điệp WM\_PAINT sẽ được gửi đến cửa sổ. Trong Win32 muốn thực hiện vẽ thì ta phải lấy handle đến device context của màn hình. Trong MFC HDC này đã được đóng gọi trong đối tượng thuộc lớp CDC. CPaintDC là lớp con của CDC, chỉ sử dụng để vẽ khi bắt thông điệp WM\_PAINT.

Khi xử lý thông điệp WM\_PAINT trong Win32, trước khi vẽ thì ta phải gọi BeginPaint vã sau khi vẽ xong thì phải gọi EndPaint. Trong MFC BeginPaint được gọi trong phương thức khởi tạo của CPaintDC và EndPaint được gọi trong phương thức hủy.

# The Message Map

* Trong chương trình Hello ở trên, làm sao MFC biết khi có thông điệp WM\_PAINT gửi đến thì phương thức nào sẽ được gọi để xử lý thông điệp WM\_PAINT?
* Câu trả lời là dựa vào message map. Message map là một table. Mỗi dòng trong table này sẽ cho biết thông điệp và phương thức xử lý tương ứng với thông điệp đó.
* Cú pháp cài đặt một message map như sau

BEGIN\_MESSAGE\_MAP (CMainWindow, CFrameWnd)

ON\_WM\_PAINT ()

END\_MESSAGE\_MAP ()

***CMainWindow*** : là lớp cửa sổ chính của ứng dụng.

***CFrameWnd*** : là lớp cha của lớp CMainWindow

Cần phải truyền luôn lớp cha của lớp cửa sổ chính là vì : Khi có một message gửi đến thì MFC sẽ tra trong message map xem có phương thức xử lý thông điệp tương ứng không. Nếu có thì MFC sẽ gọi phương thức đó, nếu không thì MFC sẽ gửi thông điệp này lên cho lớp cửa sổ cha xử lý. Lớp cửa sổ cha lại xử lý tương tự, cũng tra message map, nếu không thấy phương thức xử lý tương ứng thì tiếp tục lại forward lên lớp cha nữa.

Macro ON\_WM\_PAINT () dùng để chỉ định phương thức nào sẽ xử lý thông điệp WM\_PAINT. Nhìn vào macro không thấy một tham số nào. Vậy làm sao MFC biết người lập trình muốn phương thức nào xử lý thông điệp WM\_PAINT?

Macro ON\_WM\_PAINT() được định nghĩa là :

#define ON\_WM\_PAINT() \

{ WM\_PAINT, 0, 0, 0, AfxSig\_vv, \

(AFX\_PMSG)(AFX\_PMSGW)(void(AFX\_MSG\_CALL CWnd::\*)(void))&OnPaint },

Trong file AFXMSG\_.H đã định nghĩa cứng phương thức xử lý thông điệp WM\_PAINT là OnPaint. Do đó khi sử dụng macro ON\_WM\_PAINT() thì trong lớp CMainWindow phải định nghĩa phương thức OnPaint.

Chúng ta không thể định nghĩa các phương thử xử lý thông điệp một cách tùy tiện. Mỗi thông điệp thì prototype của phương thức xử lý là không giống nhau. Cần phải tra tài liệu MFC để biết tham số và giá trị trả về của các phương thức này.

Vậy nếu khi ta có một thông điệp do ngưởi dùng định nghĩ, không có sẵn macro thì ta làm sao để chỉ định phương thức xử lý thông điệp này trong message map?

MFC cung cấp cho chúng ta macro ON\_MESSAGE thực hiện điều này

ON\_MESSAGE (WM\_MYMESSAGE, OnMyMessage)

***WM\_MYMESSAGE*** : là ID của thông điệp

***OnMyMessage*** : phương thức xử lý thông điệp

# CDialog

* Dialog box cũng tương tự như các cửa sổ khác, cũng là một đối tượng GUI cho phép ứng dụng tương tác với người dùng.
* Dialog box có 2 loại mà Modal và Modaless
* Modal Dialog : là dạng dialog phổ biến nhất. Khi mở modal dialog thì user không được chuyển focus sang cửa sổ khác cùng ứng dụng.
* Modaless Dialog : User vẫn có thể chuyển focus qua lại giữa modaless dialog và các cửa sổ khác.
* Dialog Template : Là một đoạn resource sript định nghĩa kích thước, style, vị trí, ID...và các control trên dialog box. Đoạn resource script này ta có thể tạo ra bằng cách kéo thả nhờ vào cửa sổ Property và Toolbox của Visual Studio
* Trong MFC một đối tượng Dialog là sự kết hợp giữa một Dialog Template và một lớp (do người lập trình định nghĩa) kế thừa từ lớp CDialog của MFC.

*Giả sử sau khi tạo được một đoạn resource script như sau*

**IDD\_PREPARATION\_DLG** DIALOGEX 0, 0, 340, 229

STYLE DS\_SETFONT | DS\_FIXEDSYS | WS\_CAPTION | WS\_SYSMENU

CAPTION "Simple Dialog"

FONT 8, "MS Shell Dlg", 400, 0, 0x0

BEGIN

LTEXT "TODO: layout dialog bar",IDC\_STATIC,126,4,77,8

PUSHBUTTON "Button1",IDC\_BUTTON1,150,105,50,14

END

*Dialog box trên có ID là* : **IDD\_PREPARATION\_DLG**

*Ta sẽ thực hiện khai báo class CMyDialog như sau*

class CMyDialog : public CDialog

{

public:

CMyDialog(CWnd\* pParent = NULL); // standard constructor

virtual ~CMyDialog();

enum { **IDD = IDD\_PREPARATION\_DLG** };

};

*Cũng tương tự như CFrameWnd, CDialog cũng cho chúng ta khai báo message map để định nghĩa các sự kiện nào sẽ được xử lý*