

KỸ THUẬT LẬP TRÌNH VISUAL C++

(MFC – Microsoft Foundation Classes)

- ❖ Giới thiệu về MFC, Visual C++ 6.0
- ❖ Các loại ứng dụng trong Visual C++ 6.0
- ❖ Xử lý chuột, bàn phím
- ❖ MessageBox

NỘI DUNG

- ❖ Giới thiệu về MFC
- ❖ Microsoft Visual C++ 6.0
- ❖ Các loại ứng dụng trong windows
- ❖ Chương trình MFC đầu tiên
- ❖ Xử lý chuột và bàn phím
- ❖ Message Box

NỘI DUNG

- ❖ Giới thiệu về MFC
- ❖ Microsoft Visual C++ 6.0
- ❖ Các loại ứng dụng trong windows
- ❖ Chương trình MFC đầu tiên
- ❖ Xử lý chuột và bàn phím
- ❖ Message Box

Giới thiệu về MFC (1/2)

- ❖ **MFC** – Microsoft Foundation Class
- ❖ **MFC** là một thư viện các lớp (class, OOP) trong ngôn ngữ Visual C++, được sử dụng để xây dựng các ứng dụng trên Windows
- ❖ **MFC** được xây dựng trên cơ sở các hàm thư viện API của Windows
- ❖ Trong một ứng dụng MFC, ta sẽ không gọi trực tiếp các hàm **Windows API** mà sẽ tạo các object từ những lớp của MFC, sau đó gọi các hàm của object này
- ❖ Đa số các hàm của lớp MFC có cùng tên với những hàm Windows API

Giới thiệu về MFC (2/2)

- ❖ **MFC** tạo ra một Application Framework giúp cho lập trình viên thiết kế kiến trúc của ứng dụng nhất quán và khoa học hơn
- ❖ Thông qua các class trong MFC, người lập trình có thể xây dựng ứng dụng nhanh và ít tốn công sức hơn so với việc sử dụng đơn thuần các hàm thư viện API của Windows
- ❖ Trong MFC ta vẫn có thể gọi các hàm Windows API
- ❖ MFC có nhiều version 1, 2, 3,..6. Version 6.0 có khoảng 200 class

Một số tính năng của MFC (1/4)

❖ Version 1: (Visual C++ 1.0)

- Các lớp List, Array, String, Time, Date, File access,...
- Các lớp giao diện cơ bản
- MDI, OLE 1.0

❖ Version 2: (Visual C++ 1.0)

- File open, save
- Print preview, printing
- Scrolling window, Splitter window
- Tool bar, status bar
- Access to Visual Basic controls
- Context-sensitive help
- An improved interface to OLE version 1.0
- DLL

Một số tính năng của MFC (2/4)

❖ Version 2.5: (Visual C++ 1.5)

- Hỗ trợ kết nối ODBC (Open Database Connectivity) cho phép truy xuất đến các CSDL Access, FoxPro, SQL Server
- OLE 2.01

❖ Version 3: (Visual C++ 1.51)

- Hỗ trợ Tab dialog (property sheet)
- Docking tool bar

❖ Version 3.1: (Visual C++ 2.1)

- Hỗ trợ các control chuẩn của Windows 95
- ODBC Level 2
- Các lớp Winsock phục vụ lập trình TCP/IP

Một số tính năng của MFC (3/4)

❖ Version 4.0: (Visual C++ 4.0)

- Hỗ trợ kết nối DAO (Data Access Object)
- Windows 95 docking control bar
- Bổ sung thêm Treeview và RichEdit
- Các lớp đồng bộ hóa tiểu trình
- OLE control container

❖ Version 4.2: (Visual C++ 4.2)

- Các lớp WinInet
- Các lớp ActiveX document server
- Các tính năng mở rộng của ActiveX control
- Cải tiến một số tính năng của ODBC

Một số tính năng của MFC (4/4)

❖ Version 4.21: (Visual C++ 5.0)

- Sửa một số lỗi trong version 4.2
- Redesigned IDE, Developer Studio 97
- Hỗ trợ lập trình COM (Component Object Model)
- ...

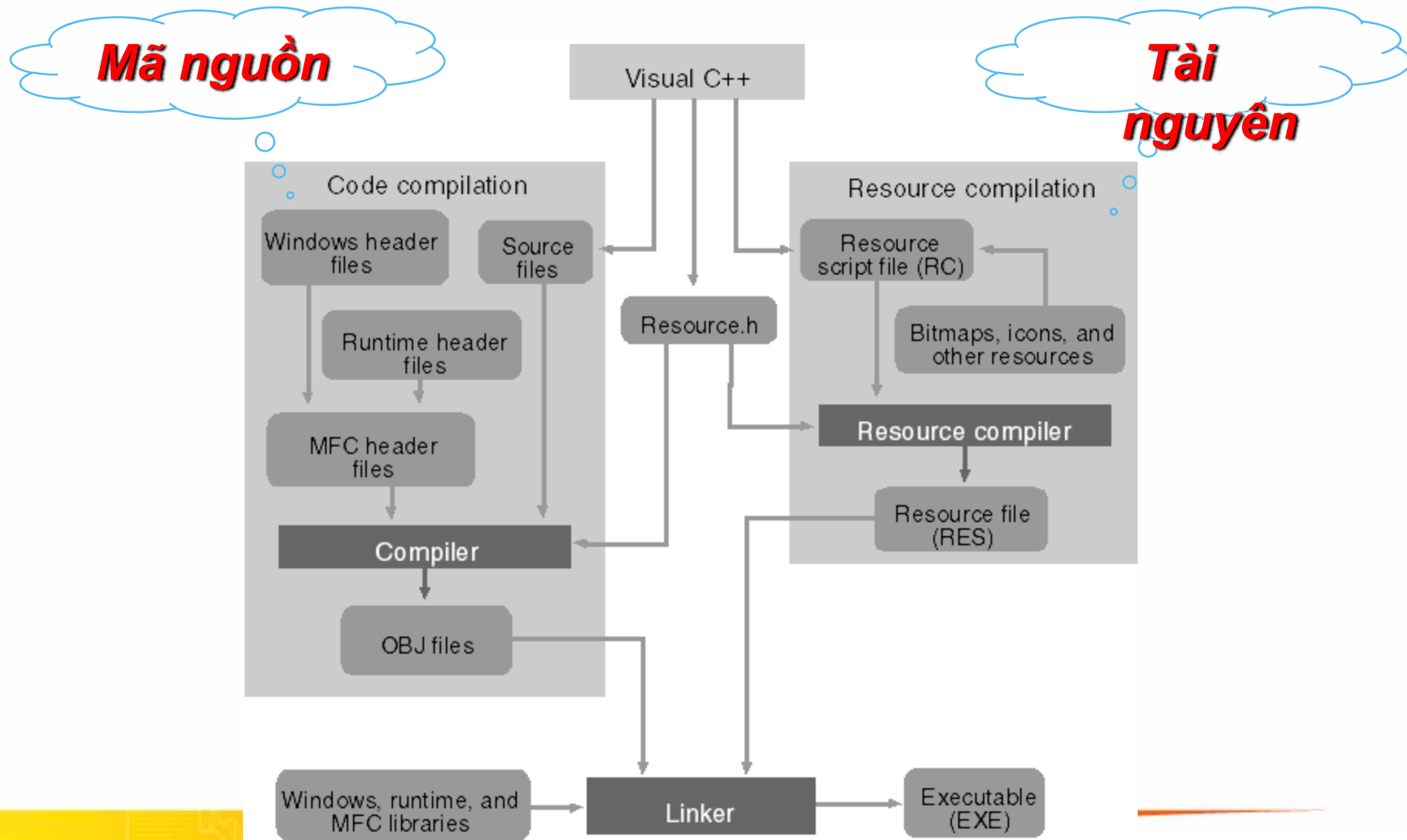
❖ Version 6.0: (Visual C++ 6.0)

- Hỗ trợ Các lớp cho những control chuẩn trong ID 4.0
- Hỗ trợ Dynamic HTML, cho phép tạo lập động các trang HTML
- Active Document Containment, cho phép ứng dụng MFC có thể chứa các Active document
- OLE DB và ADO

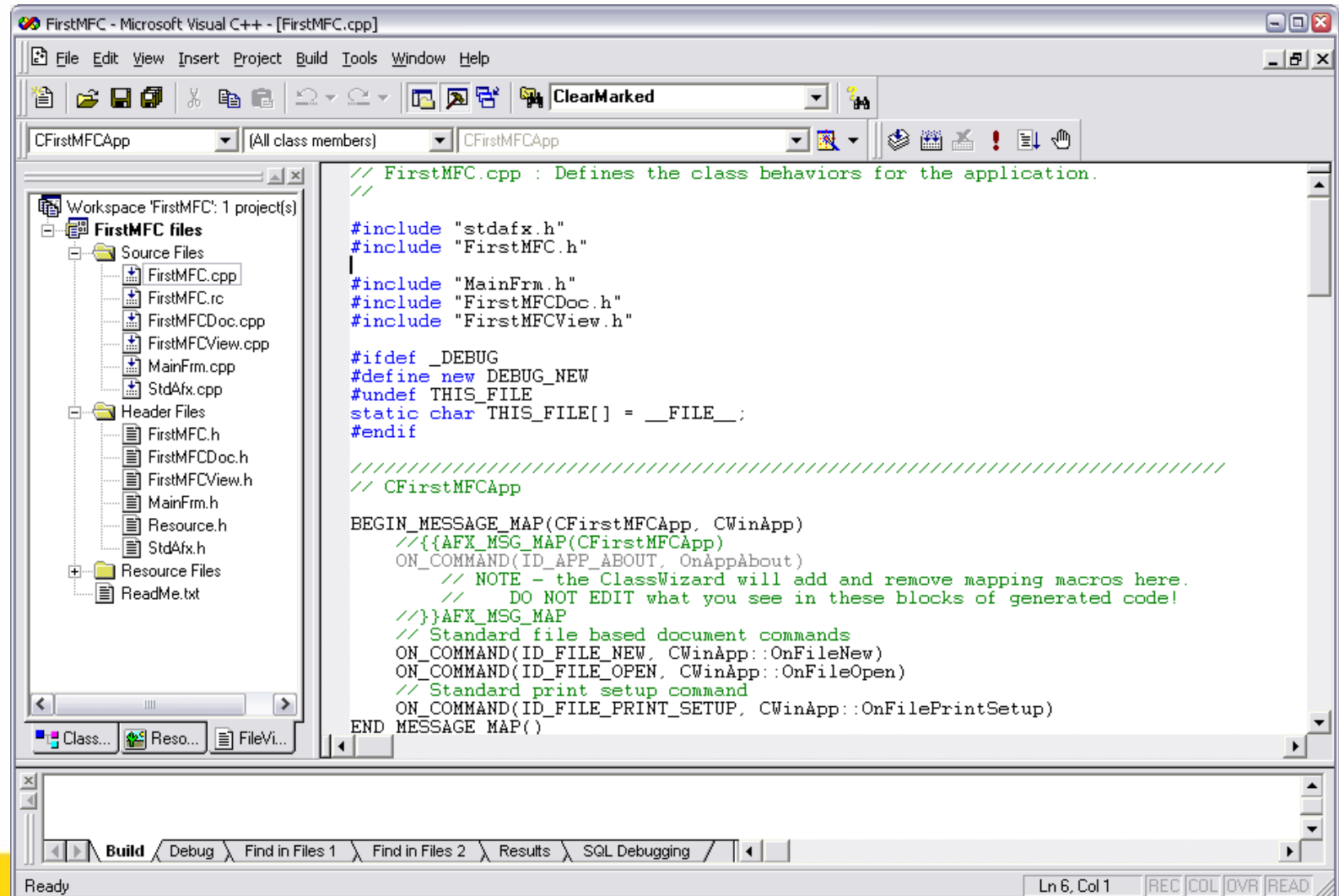
NỘI DUNG

- ❖ Giới thiệu về MFC
- ❖ Microsoft Visual C++ 6.0
- ❖ Các loại ứng dụng trong windows
- ❖ Chương trình MFC đầu tiên
- ❖ Xử lý chuột và bàn phím
- ❖ Message Box

Các thành phần trong ứng dụng Visual C++



Cửa sổ chính của MS Visual C++ 6.0



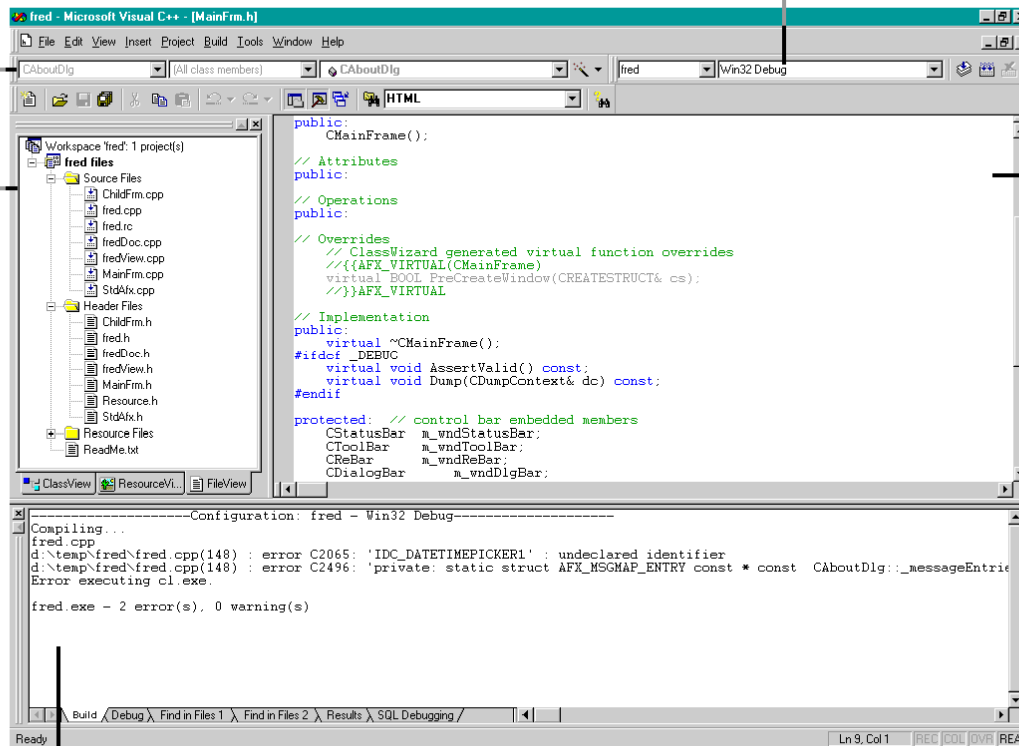
Cửa sổ chính của MS Visual C++ 6.0

Workspace window

WizardBar

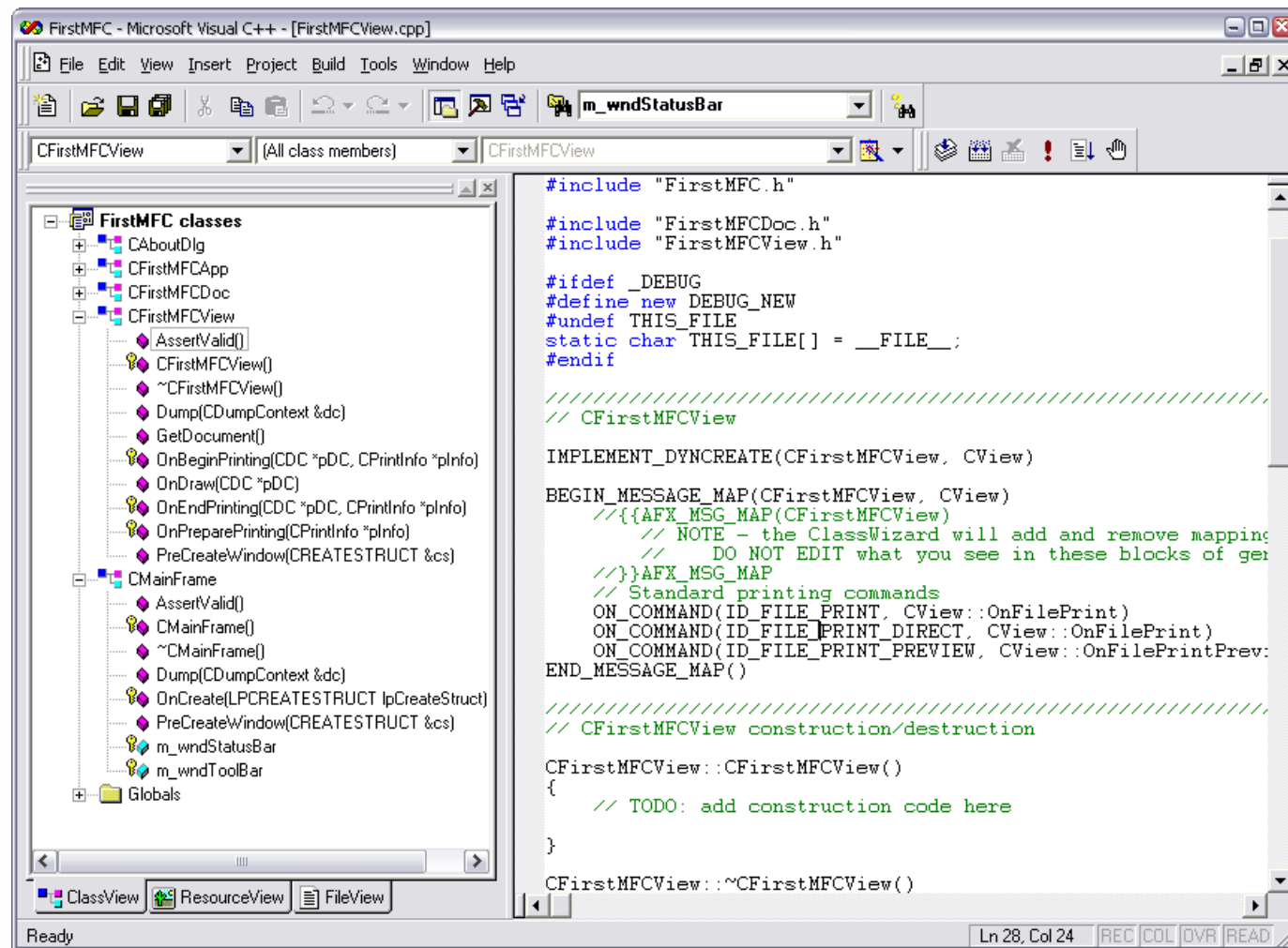
Source code

Build target

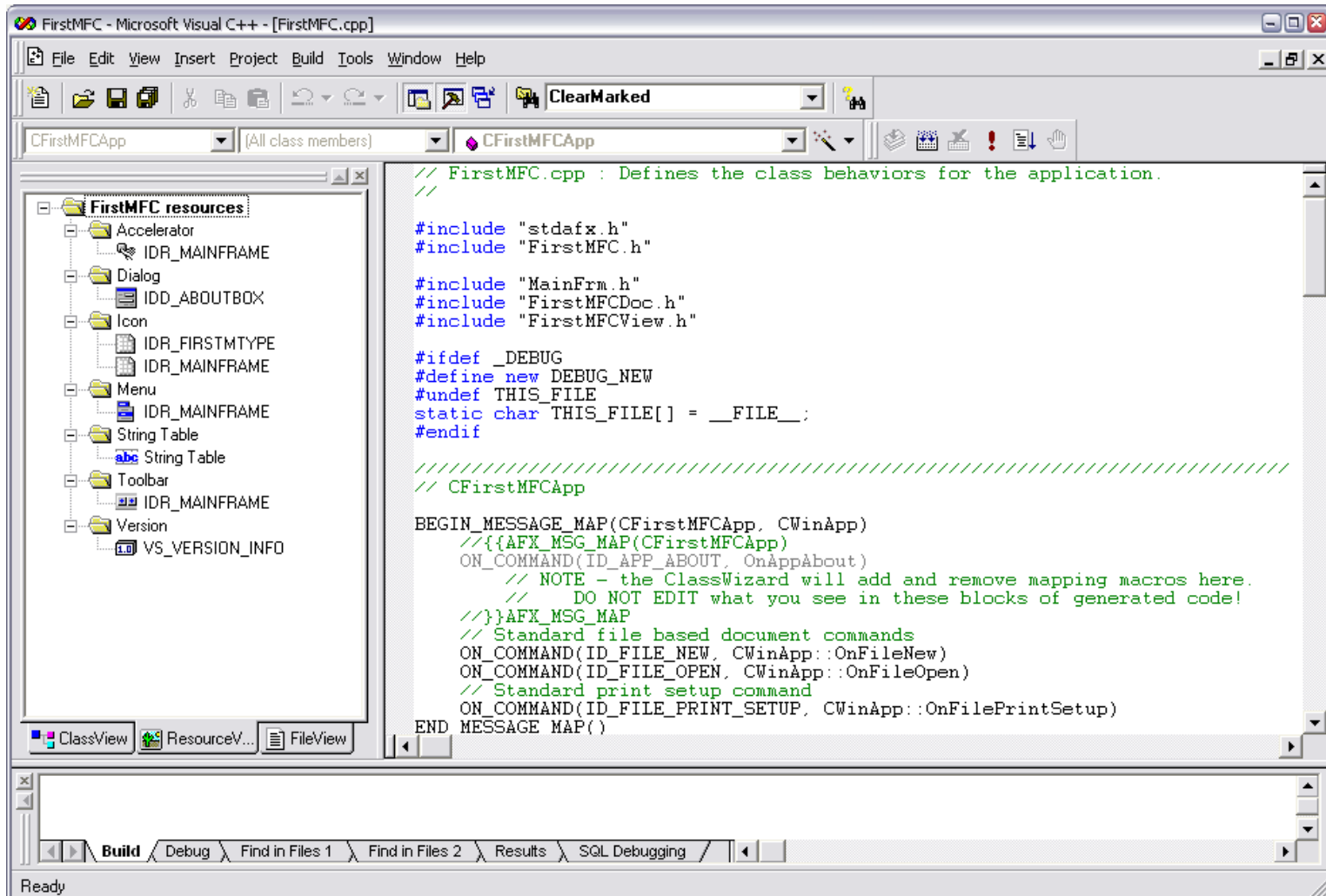


Compiler and linker output messages

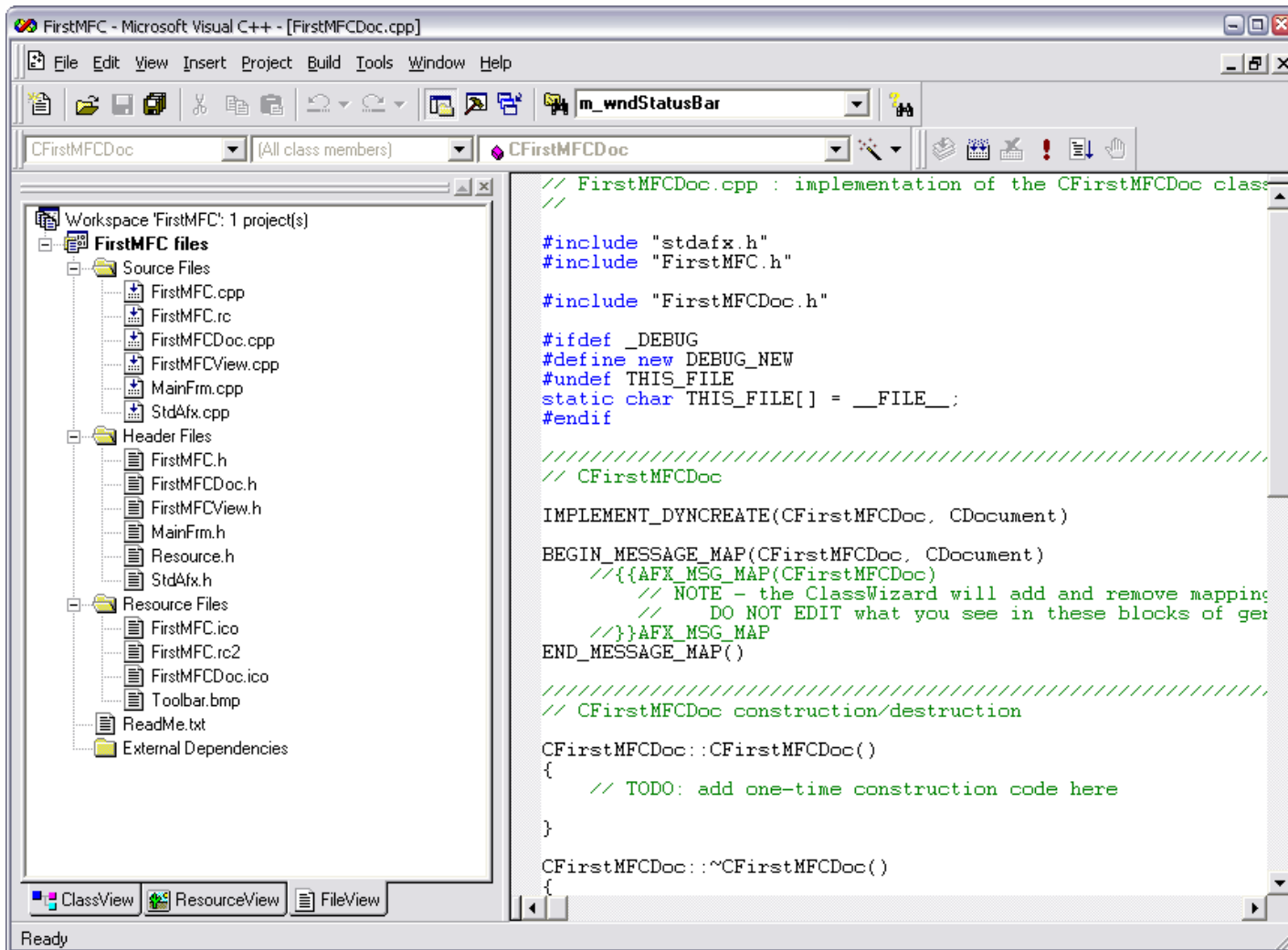
MS Visual C++ 6.0 – Class View



MS Visual C++ 6.0 – Resource View



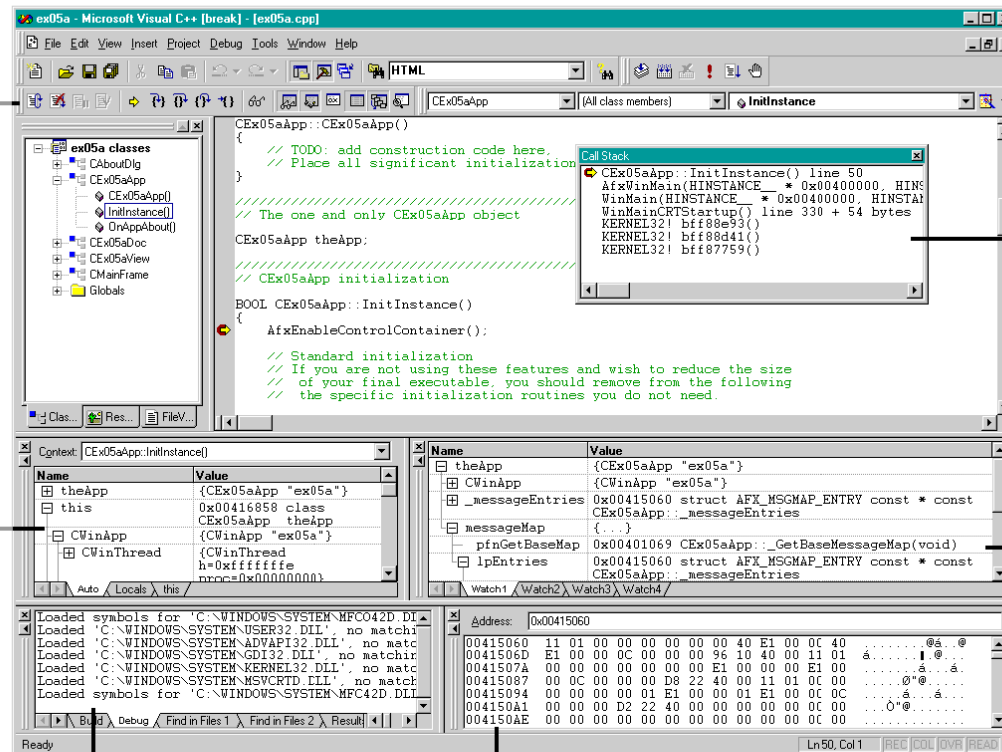
MS Visual C++ 6.0 – File View



Cửa sổ Debugger trong MS Visual C++ 6.0

Debug toolbar

Call stack



Debug output window

Memory contents

Designated watch variables

Variables from current and previous statements

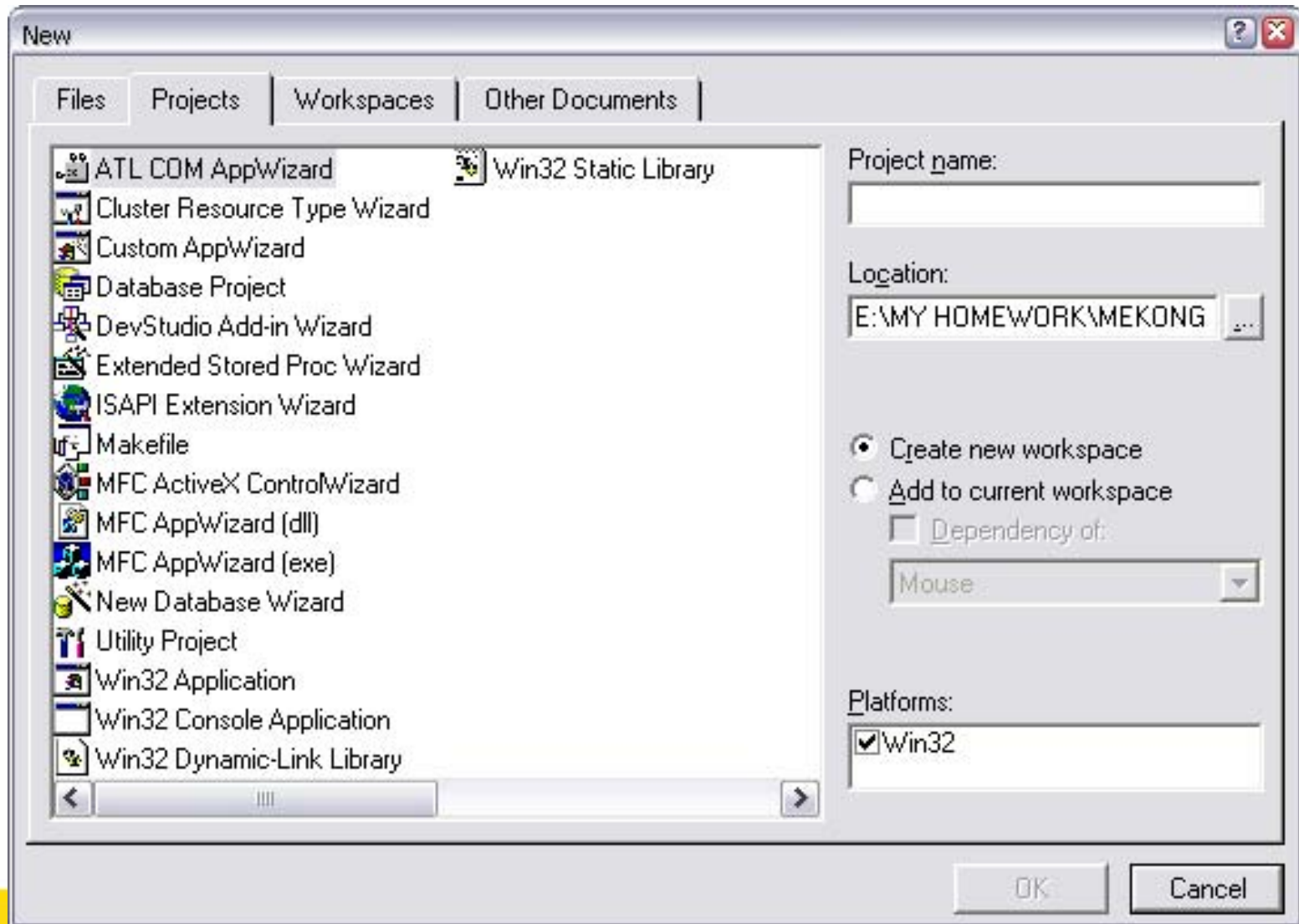
NỘI DUNG

- ❖ Giới thiệu về MFC
- ❖ Microsoft Visual C++ 6.0
- ❖ Các loại ứng dụng trong windows
- ❖ Chương trình MFC đầu tiên
- ❖ Xử lý chuột và bàn phím
- ❖ Message Box

Các loại ứng dụng (1/2)

- ❖ **Win32 Console:** ứng dụng 32 bits, với giao diện dạng DOS command line
- ❖ **Win32 (SDK):** ứng dụng 32 bits, chỉ sử dụng thư viện SDK
- ❖ **Win32 DLL:** ứng dụng 32 bits, dạng thư viện liên kết động (Dynamic – Linked Library) sử dụng SDK
- ❖ **Win32 LIB:** ứng dụng 32 bits, dạng thư viện liên kết tĩnh (Static – Linked Library) sử dụng SDK
- ❖ **MFC EXE:** ứng dụng 32 bits, sử dụng thư viện MFC
- ❖ **MFC DLL:** ứng dụng 32 bits, dạng thư viện liên kết động (Dynamic – Linked Library) sử dụng MFC

Các loại ứng dụng (2/2)



NỘI DUNG

- ❖ Giới thiệu về MFC
- ❖ Microsoft Visual C++ 6.0
- ❖ Các loại ứng dụng trong windows
- ❖ Chương trình MFC đầu tiên
- ❖ Xử lý chuột và bàn phím
- ❖ Message Box

Ứng dụng MFC đầu tiên – Hello.h

```
class CMyApp : public CWinApp
{
    public:
        virtual BOOL InitInstance ();
};
class CMainWindow : public CFrameWnd
{
    public:
        CMainWindow ();
    protected:
        afx_msg void OnPaint ();
        DECLARE_MESSAGE_MAP ()
};
```

Ứng dụng MFC đầu tiên – Hello.cpp

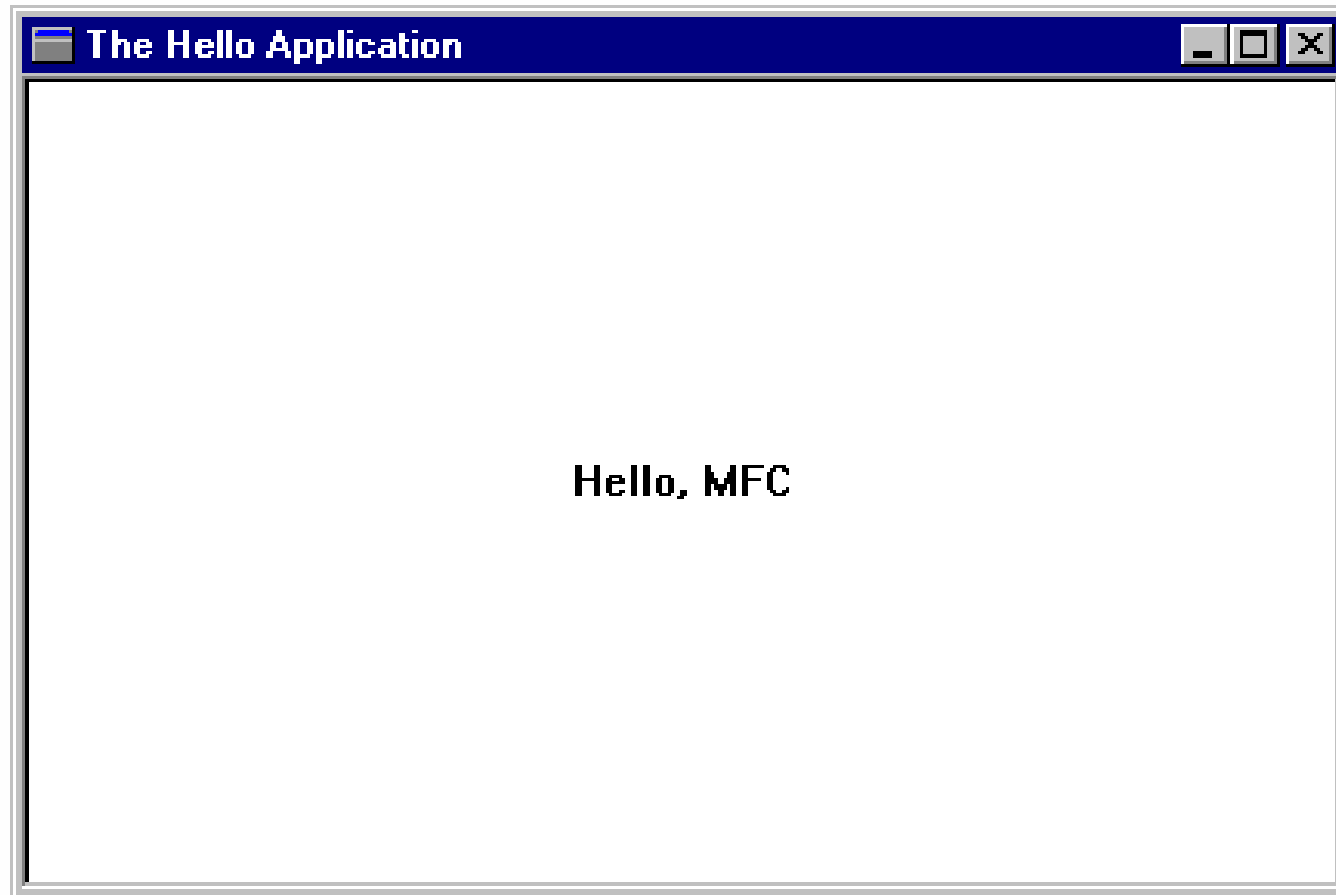
```
#include <afxwin.h>
#include "Hello.h"
CMyApp myApp;
////////////////////////////////////
// CMyApp member functions
BOOL CMyApp::InitInstance ()
{
    m_pMainWnd = new CMainWindow;
    m_pMainWnd->ShowWindow (m_nCmdShow);
    m_pMainWnd->UpdateWindow ();
    return TRUE;
} //////////////////////////////////////
```

Ứng dụng MFC đầu tiên – Hello.cpp (tt)

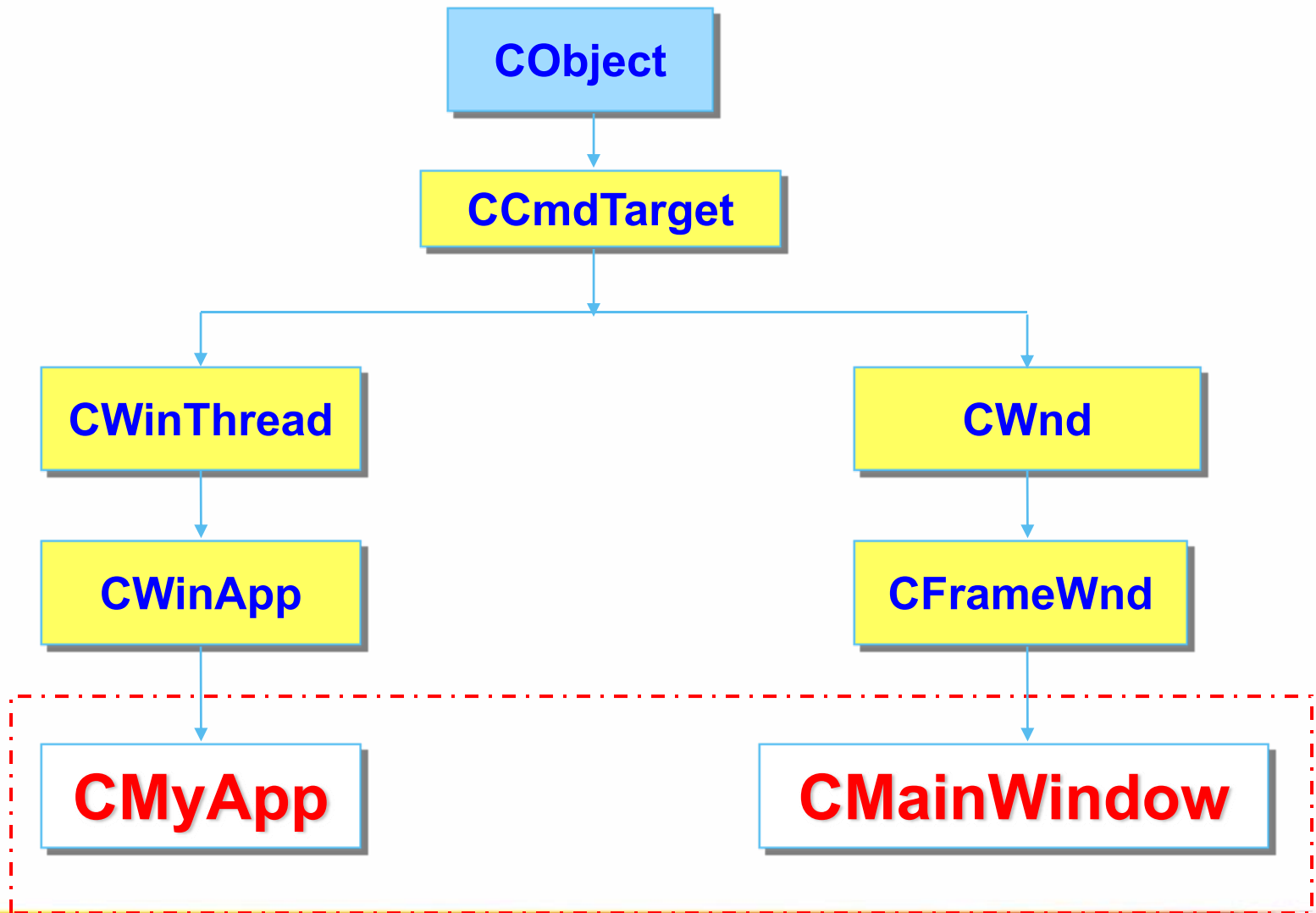
```
// CMainWindow message map and member functions
BEGIN_MESSAGE_MAP (CMainWindow, CFrameWnd)
ON_WM_PAINT ()
END_MESSAGE_MAP ()
```

```
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);
}
```


Kết quả chương trình Hello



Các class của ứng dụng Hello



Các bước thực thi ứng dụng

```
int APIENTRY WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine,
    int nCmdShow)
{
    // TODO: Place code here.
    MSG msg;

    MyRegisterClass(hInstance);
    // Perform application initialization:
    if (!InitInstance (hInstance, nCmdShow))
    {
        return FALSE;
    }
    // Main message loop:
    while (GetMessage(&msg, NULL, 0, 0))
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    return msg.wParam;
}
```

Đăng ký lớp cửa sổ

WinMain

Đăng ký lớp cửa sổ

Tạo lập cửa sổ

Hiển thị cửa sổ

Message Loop

```
ATOM MyRegisterClass(HINSTANCE hInstance)
{
    WNDCLASSEX wcex;

    wcex.cbSize = sizeof(WNDCLASSEX);

    wcex.style          = CS_HREDRAW | CS_VREDRAW;
    wcex.lpfnWndProc    = (WNDPROC)WndProc;
    wcex.cbClsExtra     = 0;
    wcex.cbWndExtra     = 0;
    wcex.hInstance     = hInstance;
    wcex.hIcon          = LoadIcon(hInstance, (LPCTSTR)IDI_BAI02);
    wcex.hCursor        = LoadCursor(NULL, IDC_ARROW);
    wcex.hbrBackground  = (HBRUSH)(COLOR_WINDOW+1);
    wcex.lpszMenuName    = (LPCSTR)IDC_BAI02;
    wcex.lpszClassName  = szWindowClass;
    wcex.hIconSm        = LoadIcon(wcex.hInstance, (LPCTSTR)IDI_SMALL);

    return RegisterClassEx(&wcex);
}
```

CMyApp

```
BOOL CMyApp::InitInstance ()
{
    m_pMainWnd = new CMainWindow;
    m_pMainWnd->ShowWindow
        (m_nCmdShow);
    m_pMainWnd->UpdateWindow ();
    return TRUE;
}
```

Tạo lập cửa sổ

WinMain

Đăng ký lớp cửa sổ

Tạo lập cửa sổ

Hiển thị cửa sổ

Message Loop

CMainWindow

```
CMainWindow::CMainWindow ()  
{  
    Create (NULL, _T ("The Hello Application"));  
}
```

```
BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)  
{  
    HWND hWnd;  
  
    hInst = hInstance; // Store instance handle in our global variable  
  
    hWnd = CreateWindow(szWindowClass, szTitle, WS_OVERLAPPEDWINDOW,  
        CW_USEDEFAULT, 0, CW_USEDEFAULT, 0, NULL, NULL, hInstance, NULL);  
  
    if (!hWnd)  
    {  
        return FALSE;  
    }  
  
    ShowWindow(hWnd, nCmdShow);  
    UpdateWindow(hWnd);  
  
    return TRUE;  
}
```

Hàm tạo lập cửa sổ

CMainWindow

```
CMainWindow::CMainWindow ()  
{  
    Create (NULL, _T ("The Hello Application"));  
}
```

```
BOOL Create (LPCTSTR lpszClassName,  
             LPCTSTR lpszWindowName,  
             DWORD dwStyle = WS_OVERLAPPEDWINDOW,  
             const RECT& rect = rectDefault,  
             CWnd* pParentWnd = NULL,  
             LPCTSTR lpszMenuName = NULL,  
             DWORD dwExStyle = 0,  
             CCreateContext* pContext = NULL)
```

Các tham số trong hàm tạo lập cửa sổ

Thuộc tính	Ý nghĩa
lpszClassName	Tên lớp cửa sổ
lpszWindowName	Tiêu đề cửa sổ
<u>dwStyle</u>	Kiểu dáng cửa sổ
rect	Kích thước và vị trí ban đầu của cửa sổ
pParentWnd	Cửa sổ cha, NULL cửa sổ không có cửa sổ cha
lpszMenuName	Thực đơn liên quan đến cửa sổ, NULL nghĩa là cửa sổ không có Menu
dwExStyle	Kiểu mở rộng của cửa sổ
pContext	Sử dụng trong ứng dụng Documents/Views

Các kiểu dáng chuẩn của cửa sổ

Kiểu	Ý nghĩa
WS_BORDER	Tạo cửa sổ với khung mỏng
WS_CAPTION	Tạo cửa sổ với thanh tiêu đề
WS_CHILD	Tạo cửa sổ con, sử dụng để tạo điều khiển như phím và các cửa sổ con MDI
WS_OVERLAPPED	Lập cửa sổ maximize và không có cửa sổ cha
WS_OVERLAPPEDWINDOW	Lập cửa sổ với các kiểu: WS_OVERLAPPED, WS_CAPTION, WS_SYSMENU, WS_MAXIMIZEBOX, WS_MINIMIZEBOX,
WS_DISABLED	Cửa sổ sẽ bị vô hiệu lực
WS_DLGMODALFRAME	Cửa sổ có viền như dialog
WS_MAXIMIZE	Cửa sổ có Maximize
WS_MAXIMIZEBOX	Cửa sổ có nút maximize trên thanh tiêu đề
WS_HSCROLL	Cửa sổ có thanh trượt ngang
WS_VSCROLL	Cửa sổ có thanh trượt dọc

Hiển thị cửa sổ

WinMain

Đăng ký lớp cửa sổ

Tạo lập cửa sổ

Hiển thị cửa sổ

Message Loop

```
BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)
{
    HWND hWnd;

    hInst = hInstance; // Store instance handle in our global variable

    hWnd = CreateWindow(szWindowClass, szTitle, WS_OVERLAPPEDWINDOW,
        CW_USEDEFAULT, 0, CW_USEDEFAULT, 0, NULL, NULL, hInstance, NULL);

    if (!hWnd)
    {
        return FALSE;
    }

    ShowWindow(hWnd, nCmdShow);
    UpdateWindow(hWnd);

    return TRUE;
}
```

CMainWindow

```
BOOL CMyApp::InitInstance ()
{
    m_pMainWnd = new CMainWindow;
    m_pMainWnd->ShowWindow
        (m_nCmdShow);
    m_pMainWnd->UpdateWindow ();
    return TRUE;
}
```

Vòng lặp và xử lý thông điệp

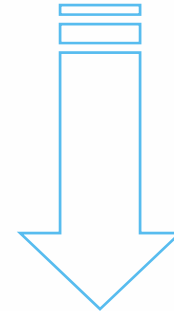
```
while (GetMessage (&msg, NULL, 0, 0))
{
    TranslateMessage (&msg);
    DispatchMessage (&msg);
}

LRESULT CALLBACK WndProc (HWND hwnd, UINT message, WPARAM wParam,
LPARAM lParam)
{
    PAINTSTRUCT ps;
    HDC hdc;
    switch (message) {
        case WM_PAINT:
            hdc = BeginPaint (hwnd, &ps);
            Ellipse (hdc, 0, 0, 200, 100);
            EndPaint (hwnd, &ps);
            return 0;

        case WM_DESTROY:
            PostQuitMessage (0);
            return 0;
    }
    return DefWindowProc (hwnd, message, wParam, lParam);
}
```

CMainWindow

**Tiếp nhận và xử lý
thông điệp như
thế nào ?**



Message Map

Message Map – Ánh xạ thông điệp

- ❖ Ánh xạ thông điệp là bảng tương quan giữa thông điệp và các hàm thành phần quản lý các thông điệp đó
- ❖ Bất kỳ lớp nào suy diễn từ CCmdTarget đều có thể chứa ánh xạ thông điệp
- ❖ Ánh xạ thông điệp được khai báo như các thành phần lớp khác trong khai báo lớp

```
class CMainWindow : public CFrameWnd
{
    public:
        CMainWindow ();
    protected:
        afx_msg void OnPaint ();
        DECLARE_MESSAGE_MAP ()
};
```

Message Map (tt)

- ❖ Mỗi message sẽ được xử lý bởi 1 hàm thành phần tương ứng
- ❖ **Những thông điệp với tiền tố WM_XYZ sẽ được cài đặt tương ứng với hàm theo qui tắc**
 - WM_ được chuyển thành On
 - XYZ được chuyển thành chữ thường trừ ký tự đầu
- ❖ **Ví dụ:**
 - WM_PAINT sẽ được chuyển thành OnPaint
 - WM_MOUSEMOVE sẽ được chuyển thành OnMouseMove
 - WM_LBUTTONDOWN sẽ được chuyển thành OnLButtonDown

Message Map (tt)

❖ Khai báo Message Map

```
class CMainWindow : public CFrameWnd
{
    public:
        CMainWindow ();
    protected:
        afx_msg void OnPaint ();
        afx_msg void OnLButtonDown(UINT, CPoint);
        DECLARE_MESSAGE_MAP ()
};
```

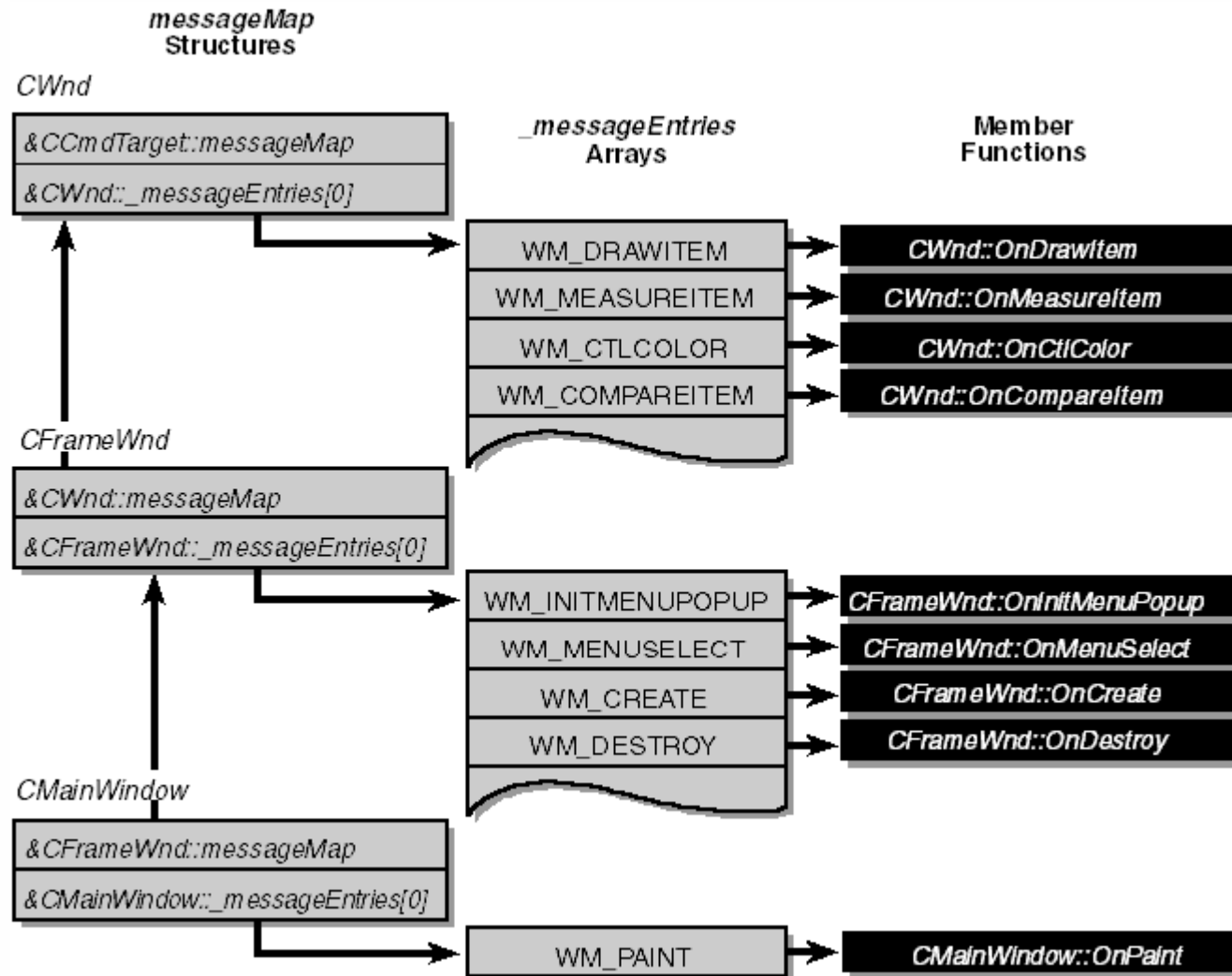
❖ Cài đặt xử lý Message

```
BEGIN_MESSAGE_MAP (CMainWindow, CFrameWnd)
    ON_WM_PAINT ()
    ON_WM_LBUTTONDOWN()
END_MESSAGE_MAP ()
Void CMainWindow::OnPaint()
{
}
Void CMainWindow::OnLButtonDown(UINT nFlags, CPoint point)
{
}
```

Thông điệp WM_XXX và hàm MFC tương ứng

Message	Afx function
WM_CHAR	afx_msg void OnChar(UINT,UINT,UINT)
WM_CREATE	afx_msg int OnCreate(LPCREATESTRUCT)
WM_DESTROY	afx_msg void OnDestroy()
WM_LBUTTONDOWN	afx_msg void OnLButtonDown(UINT, CPoint)
WM_LBUTTONUP	afx_msg void OnLButtonUp(UINT, CPoint)
WM_MOUSEMOVE	afx_msg void OnMouseMove(UINT, CPoint)
WM_RBUTTONDOWN	afx_msg void OnRButtonDown(UINT, CPoint)
WM_RBUTTONUP	afx_msg void OnRButtonUp(UINT, CPoint)
WM_SIZE	afx_msg void OnSize()
WM_PAINT	afx_msg void OnPaint()

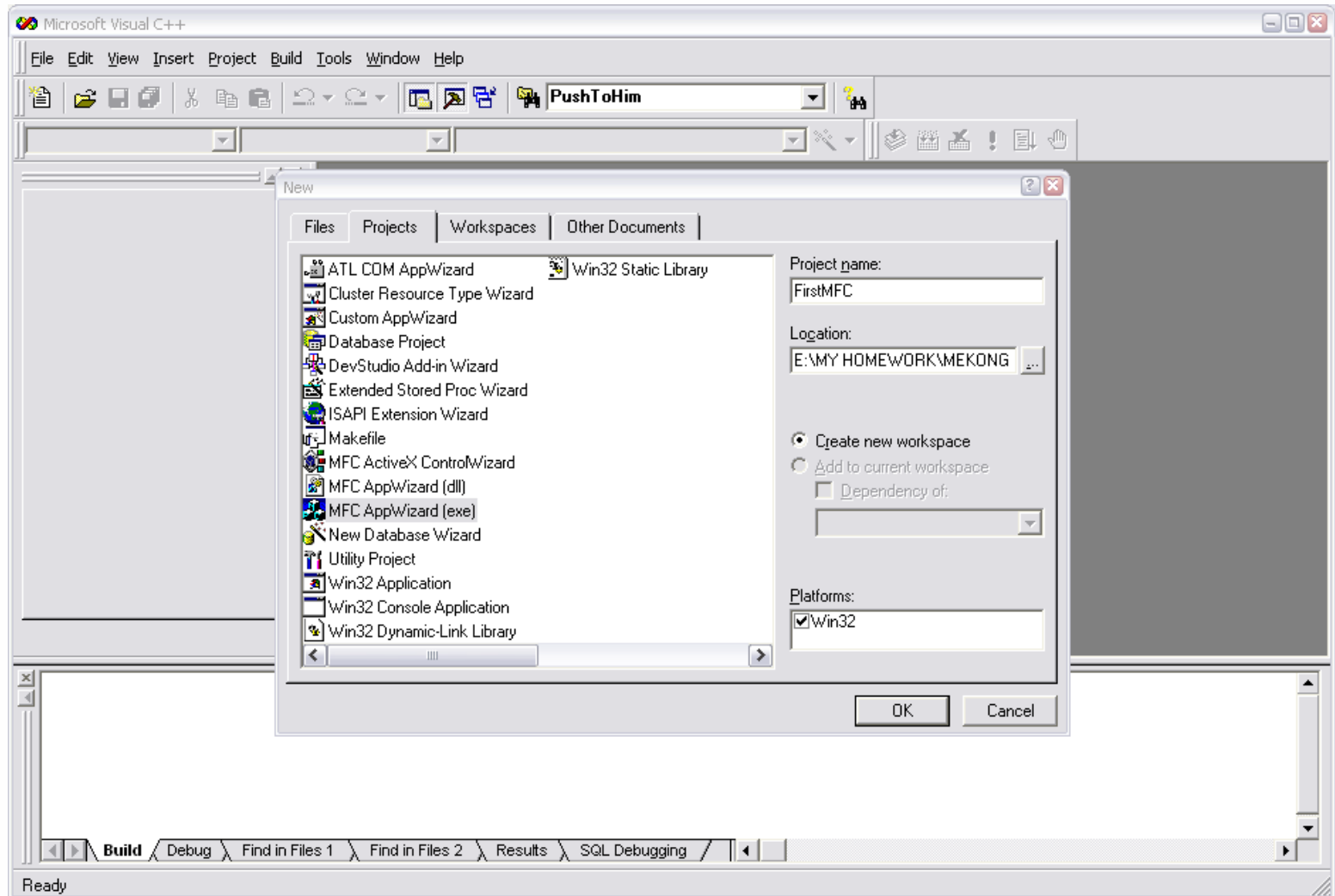
Xử lý thông điệp cửa sổ - Message Map



Tạo ứng dụng MFC bằng AppWizard

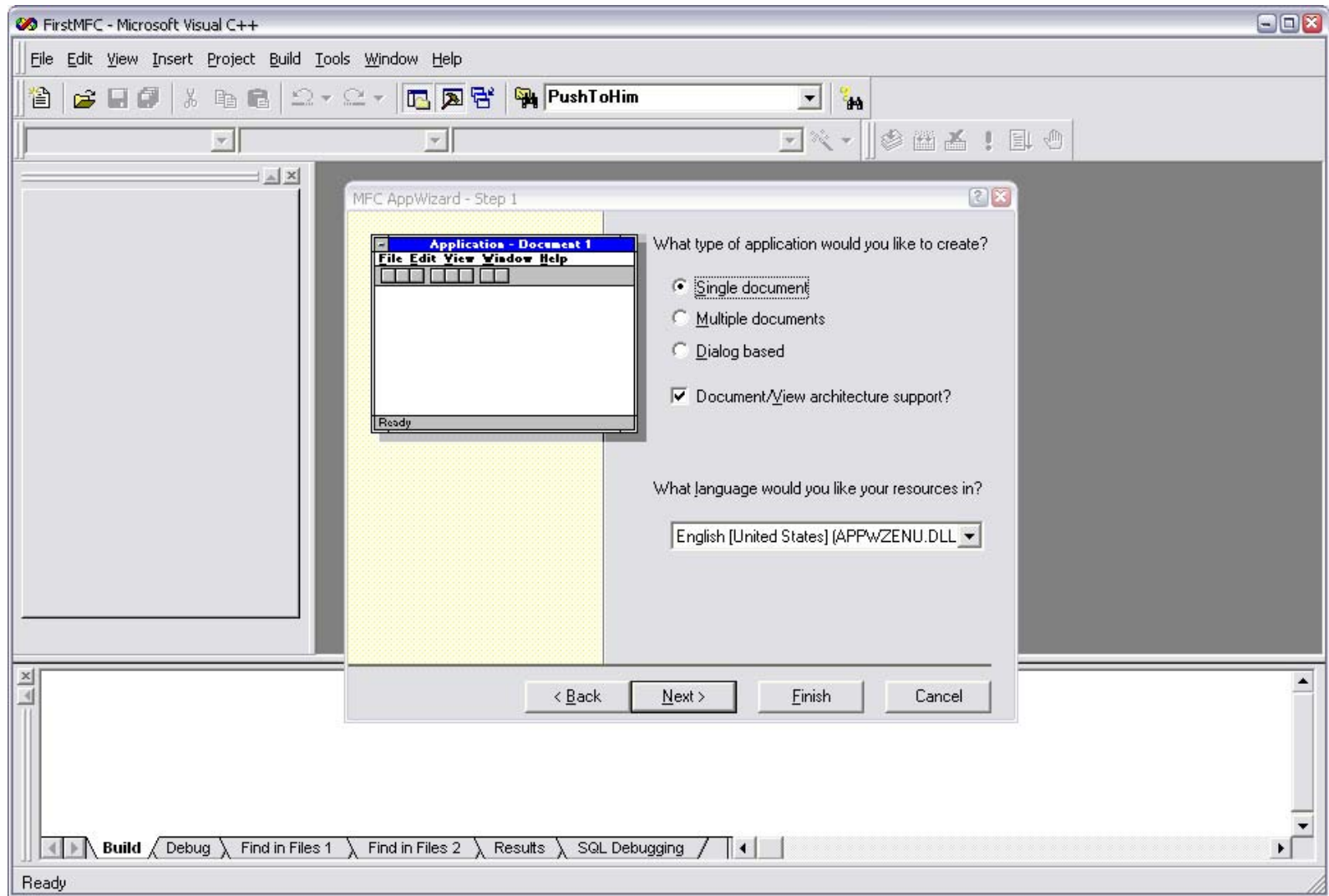
- ❖ Chọn menu **File → New**
- ❖ Chọn Tab **Projects**
- ❖ Chọn loại project **"MFC AppWizard (exe)"**
- ❖ Đặt tên project trong ô **"Project name"**
- ❖ Xác định đường dẫn thư mục trong **"Location"**
- ❖ Step 1: Chọn loại ứng dụng **"Single Document"**, bỏ option **"Document/View architecture support"**
- ❖ Nhấn **"Next"** đến **"Step 4 of 6"**

Tạo ứng dụng MFC bằng AppWizard



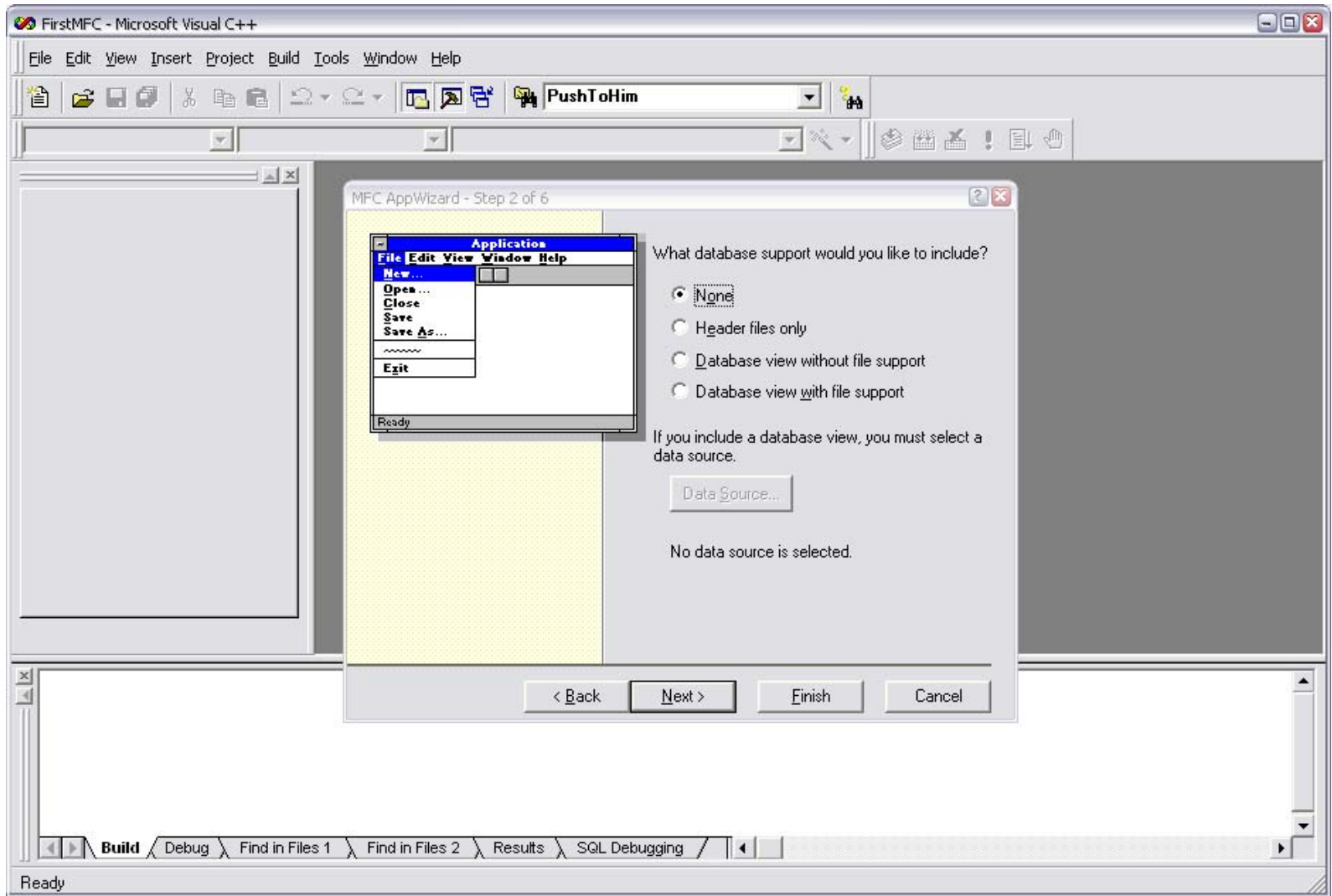
Chọn loại project “MFC AppWizard (EXE)”

Tạo ứng dụng MFC bằng AppWizard



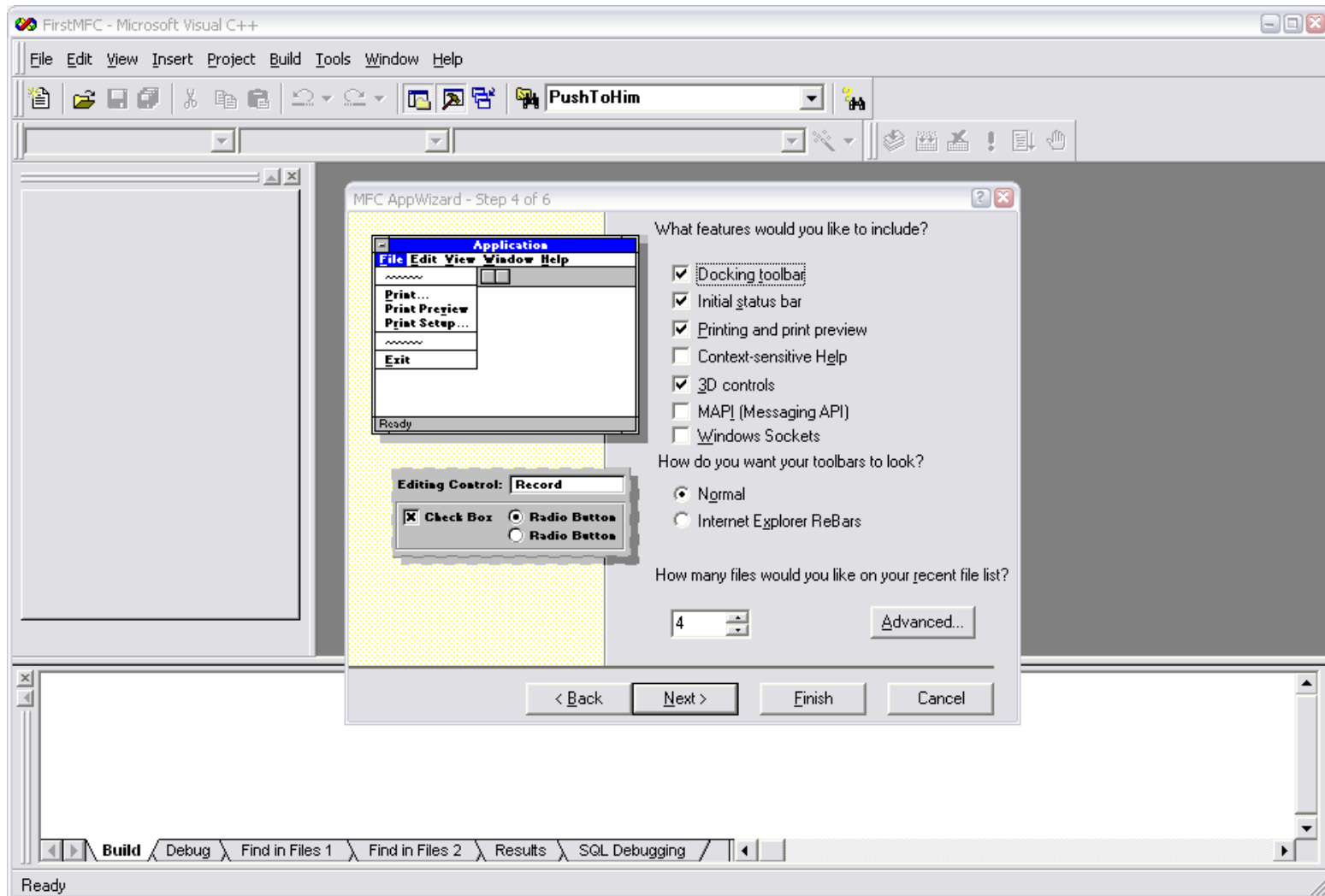
Chọn option “Single document”

Tạo ứng dụng MFC bằng AppWizard

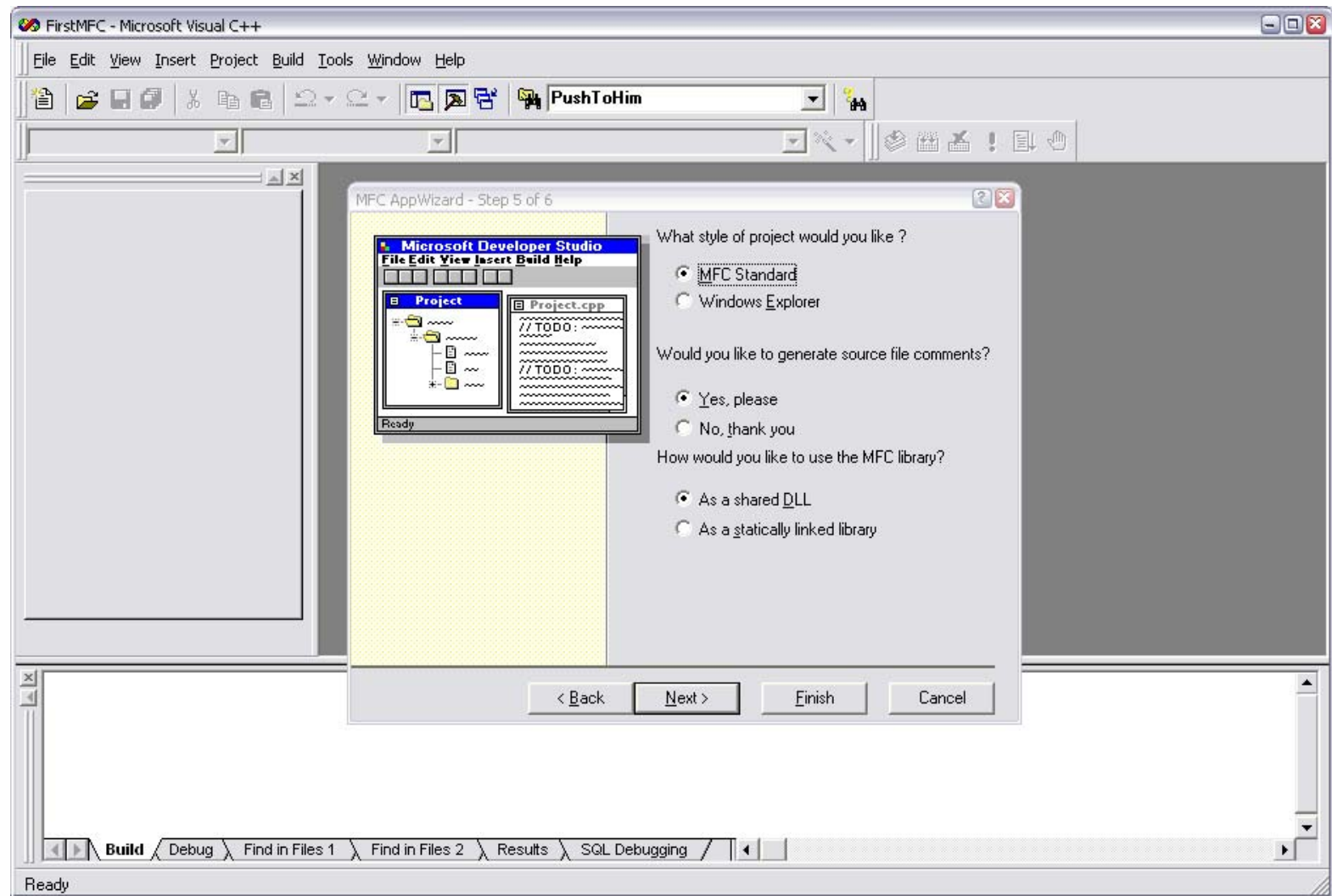


Chọn option "None"

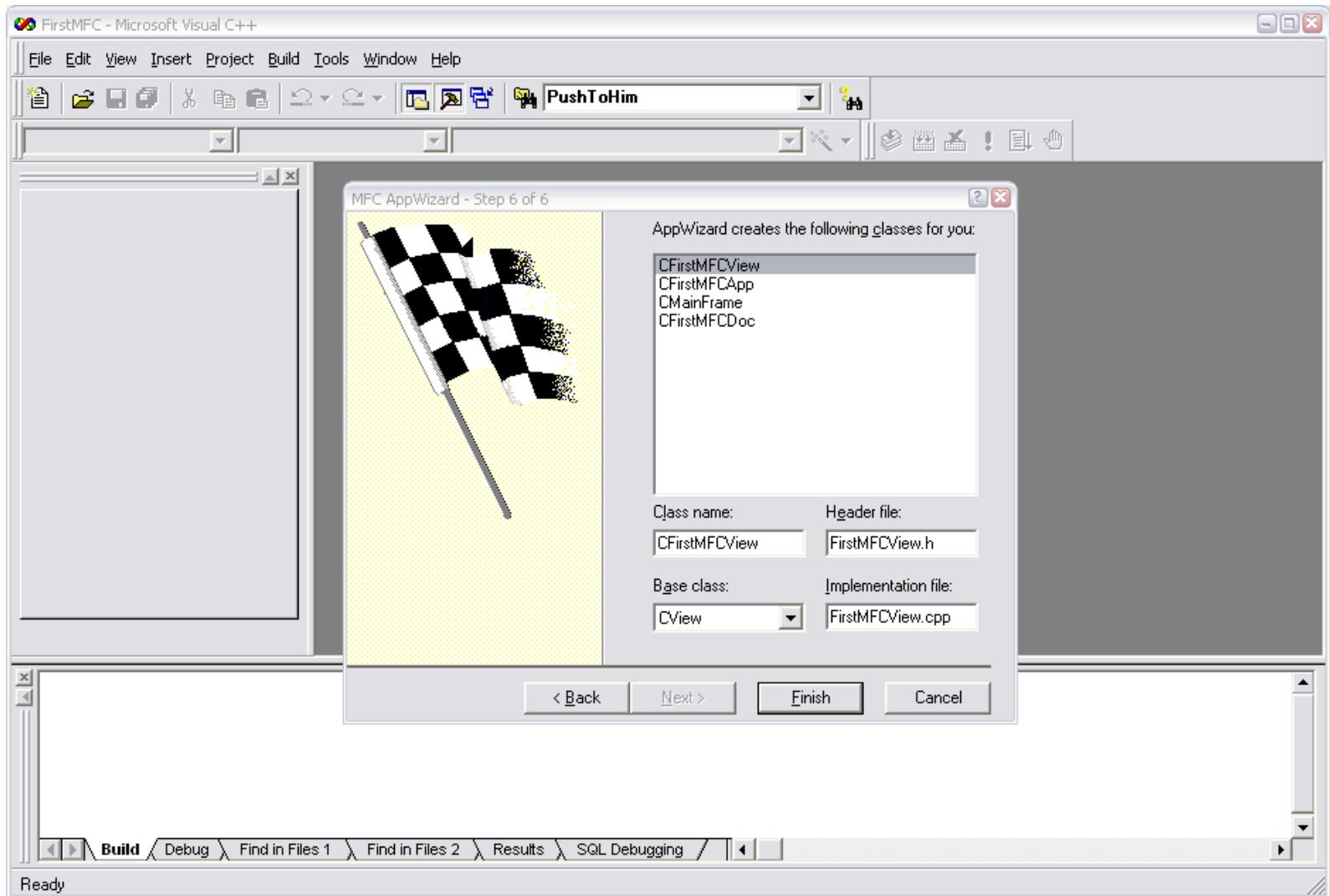
Tạo ứng dụng MFC bằng AppWizard



Tạo ứng dụng MFC bằng AppWizard



Tạo ứng dụng MFC bằng AppWizard



Tạo ứng dụng MFC bằng AppWizard

Application type of FirstMFC:

Single Document Interface Application targeting:
Win32

Classes to be created:

Application: CFirstMFCApp in FirstMFC.h and FirstMFC.cpp

Frame: CMainFrame in MainFrm.h and MainFrm.cpp

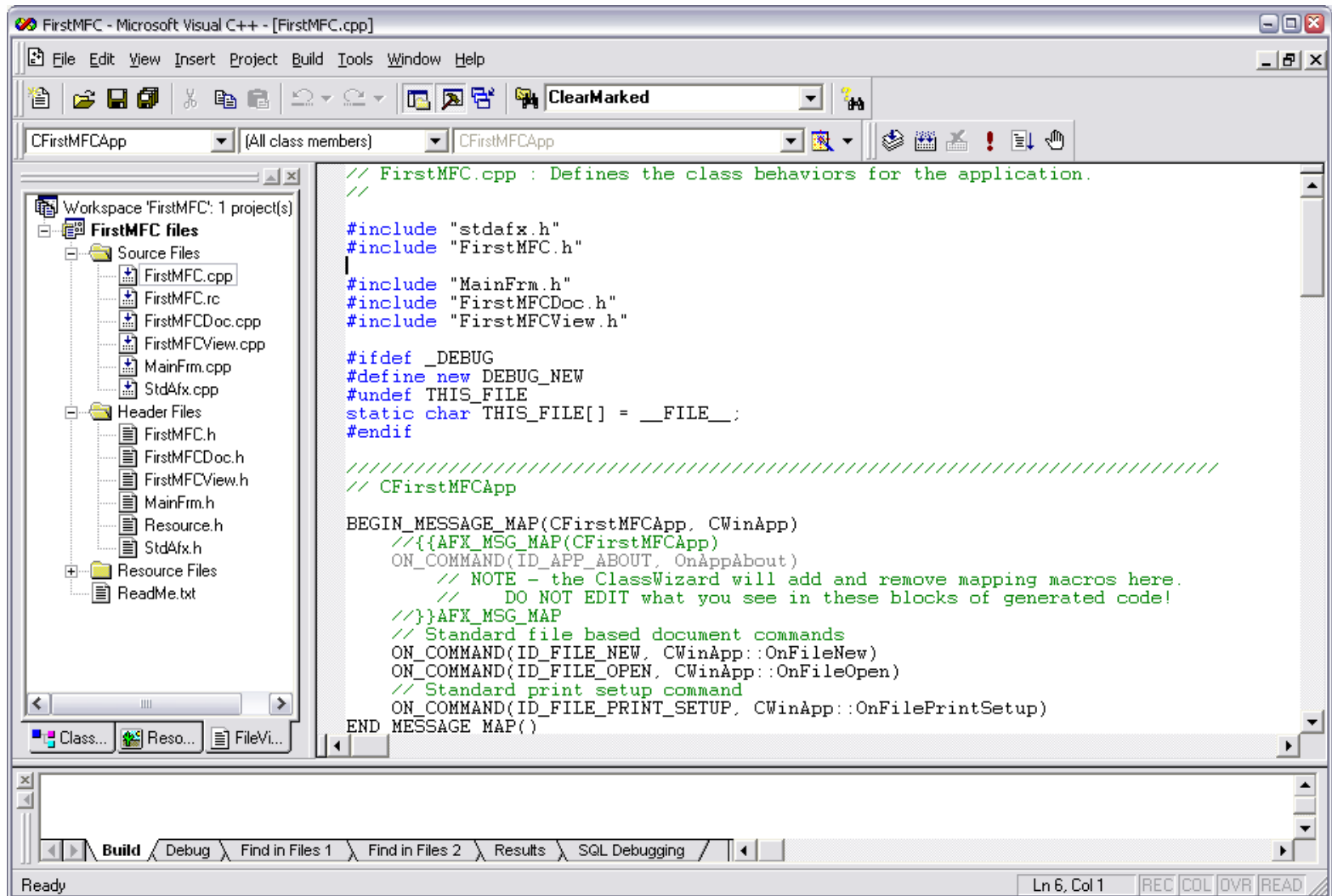
Document: CFirstMFCDoc in FirstMFCDoc.h and FirstMFCDoc.cpp

View: CFirstMFCView in FirstMFCView.h and FirstMFCView.cpp

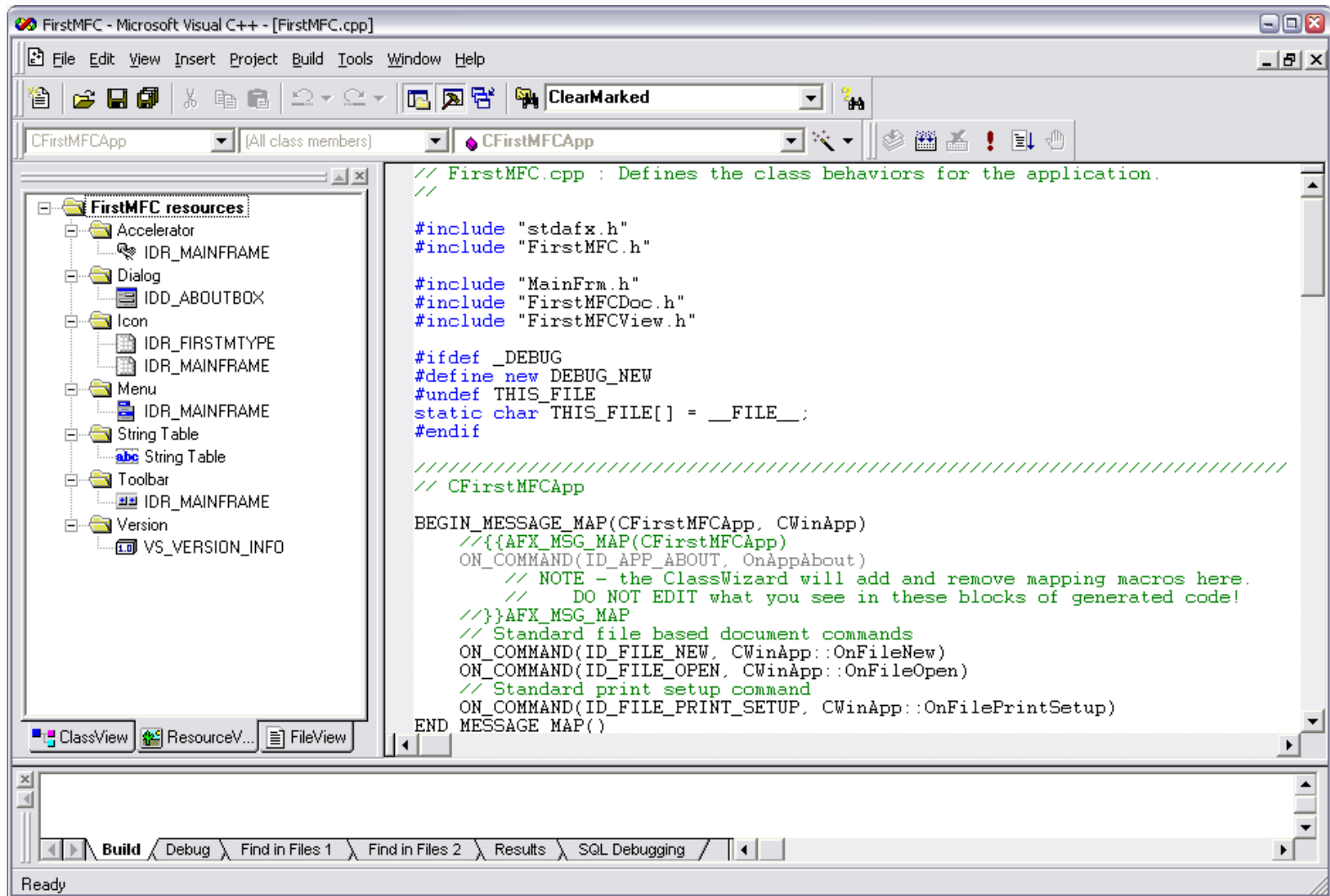
Features:

- + Initial toolbar in main frame
- + Initial status bar in main frame
- + Printing and Print Preview support in view
- + 3D Controls
- + Uses shared DLL implementation (MFC42.DLL)
- + ActiveX Controls support enabled
- + Localizable text in:
 - English [United States]

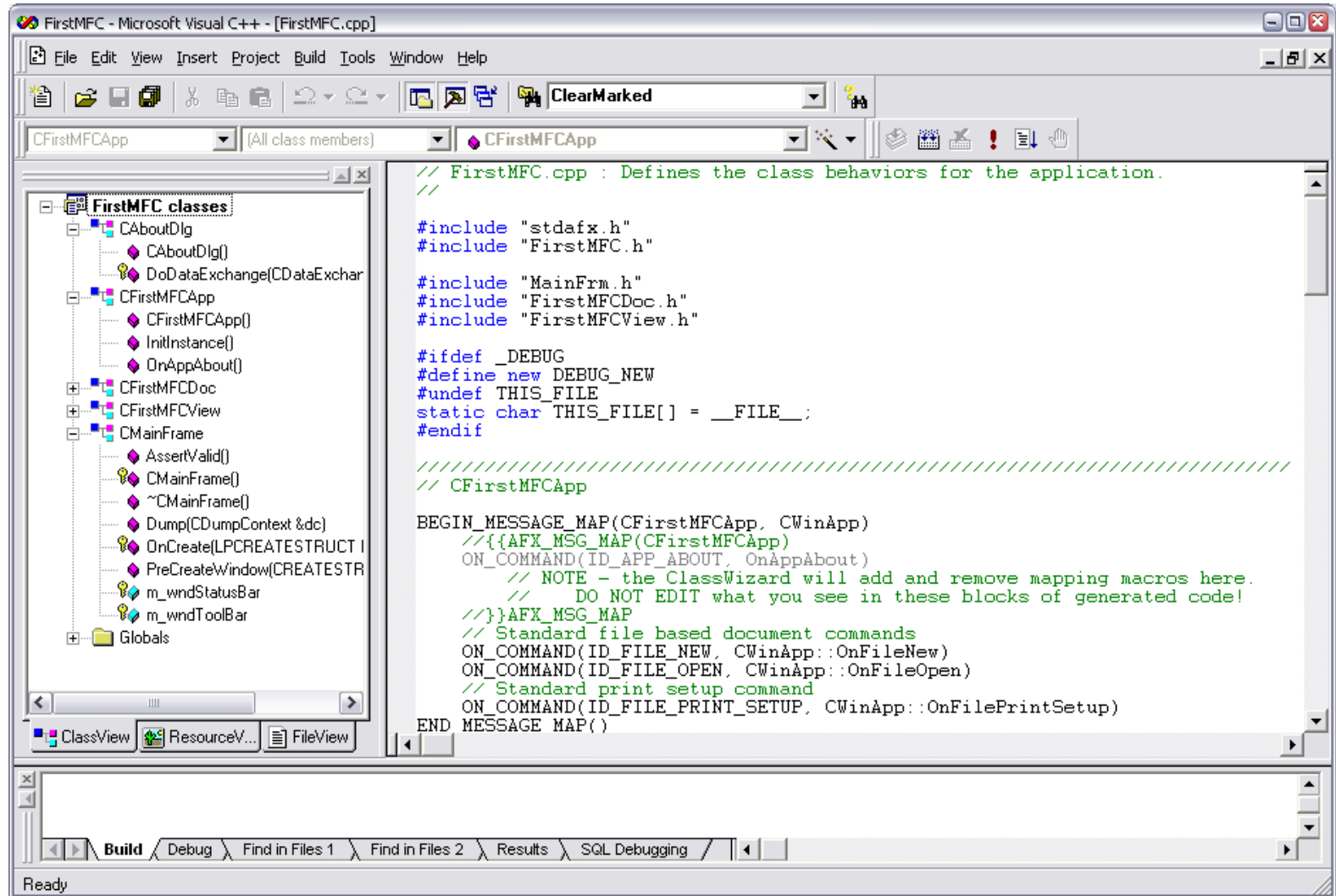
Ứng dụng MFC đầu tiên – File view



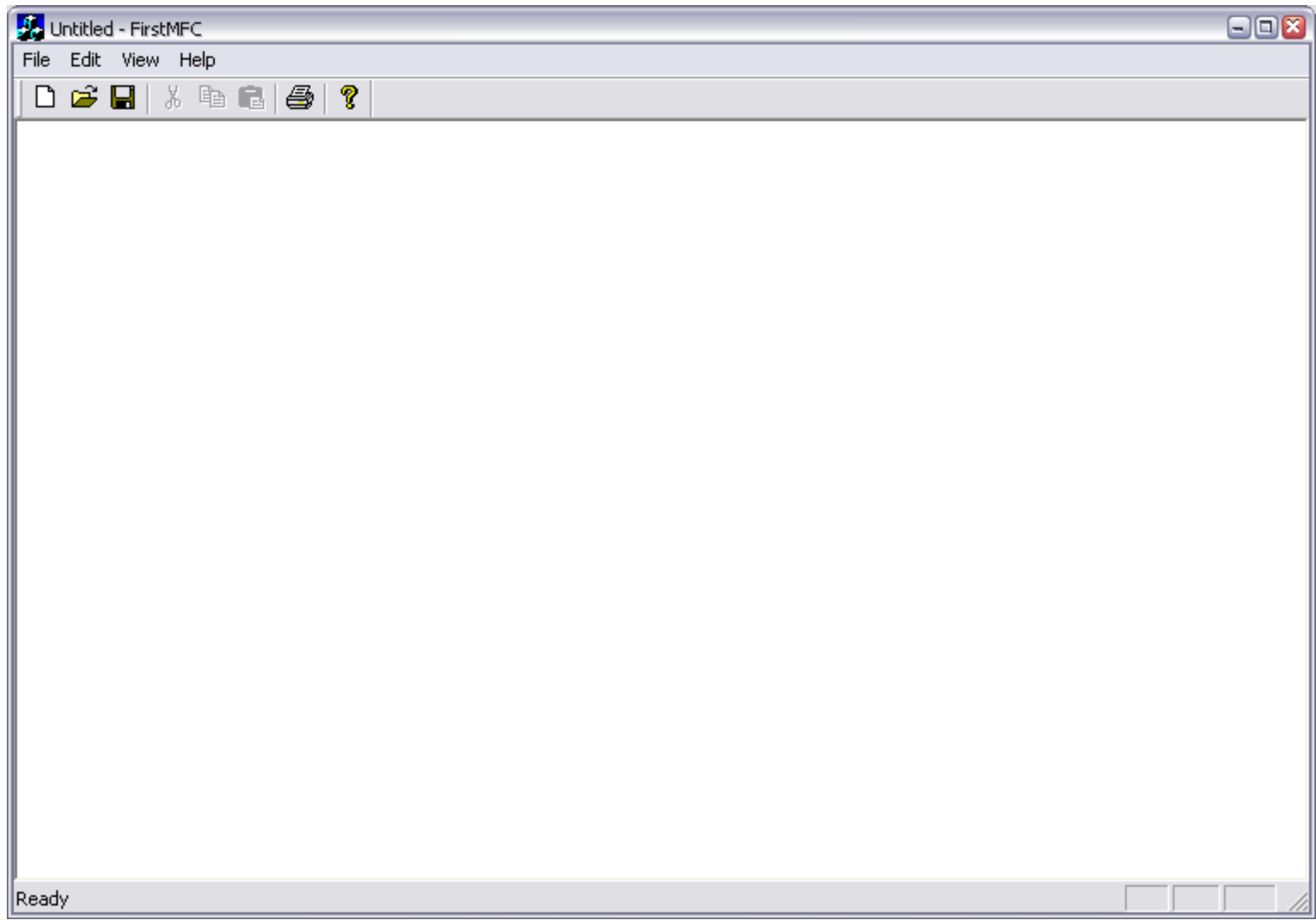
Ứng dụng MFC đầu tiên – Resource view



Ứng dụng MFC đầu tiên – Class view



Ứng dụng MFC đầu tiên – kết quả chương trình



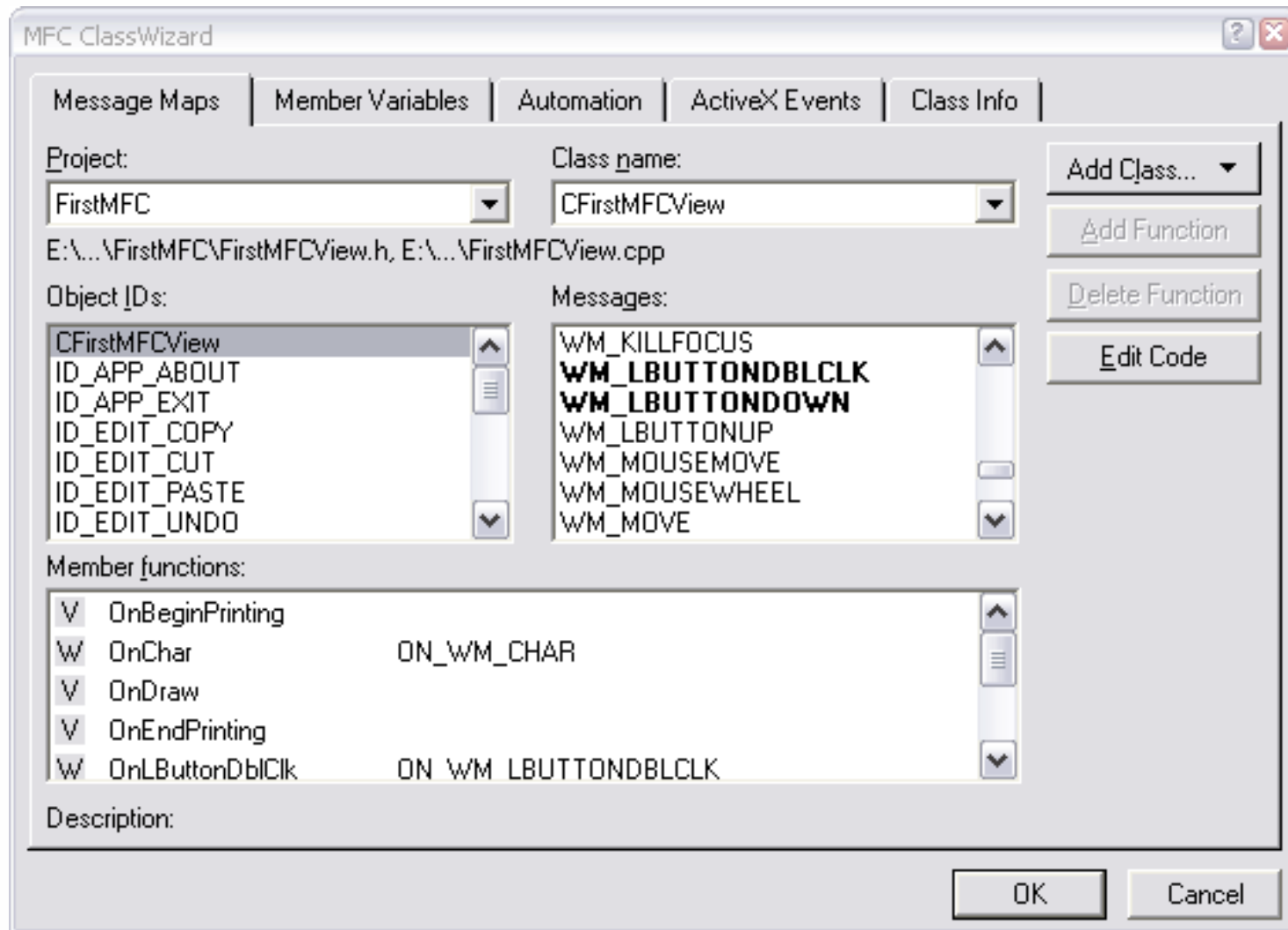
NỘI DUNG

- ❖ Giới thiệu về MFC
- ❖ Microsoft Visual C++ 6.0
- ❖ Các loại ứng dụng trong windows
- ❖ Chương trình MFC đầu tiên
- ❖ Xử lý chuột và bàn phím
- ❖ Message Box

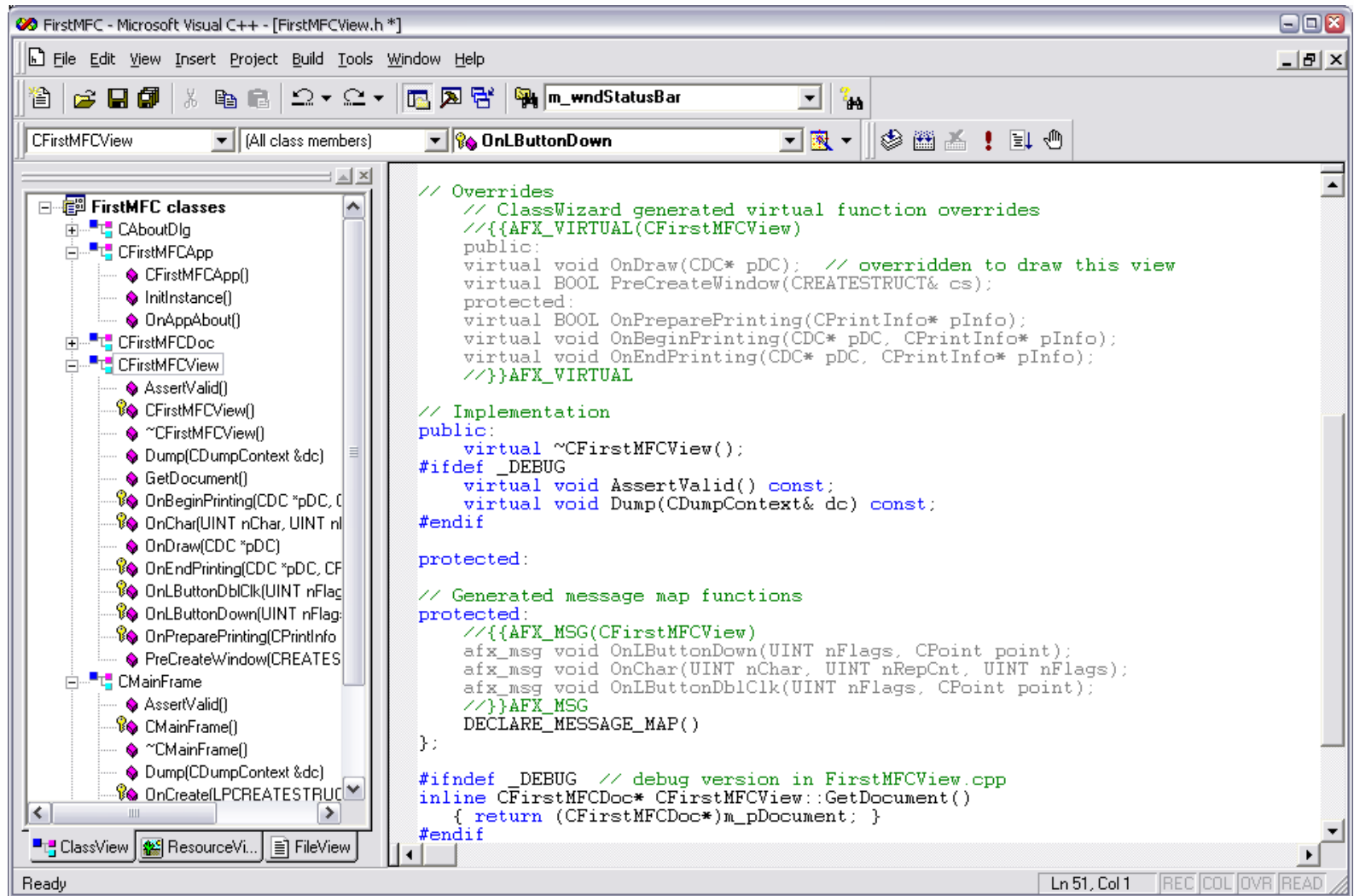
Xử lý thông điệp chuột

Message	Afx function
WM_LBUTTONDOWN	<code>afx_msg void OnLButtonDown(UINT, CPoint)</code>
WM_LBUTTONUP	<code>afx_msg void OnLButtonUp(UINT, CPoint)</code>
WM_LBUTTONBLCLK	<code>afx_msg void OnLButtonDblClk(UINT, CPoint)</code>
WM_RBUTTONDOWN	<code>afx_msg void OnRButtonDown(UINT, CPoint)</code>
WM_RBUTTONUP	<code>afx_msg void OnRButtonUp(UINT, CPoint)</code>
WM_RBUTTONBLCLK	<code>afx_msg void OnRButtonDblClk(UINT, CPoint)</code>
WM_MOUSEMOVE	<code>afx_msg void OnMouseMove(UINT, CPoint)</code>

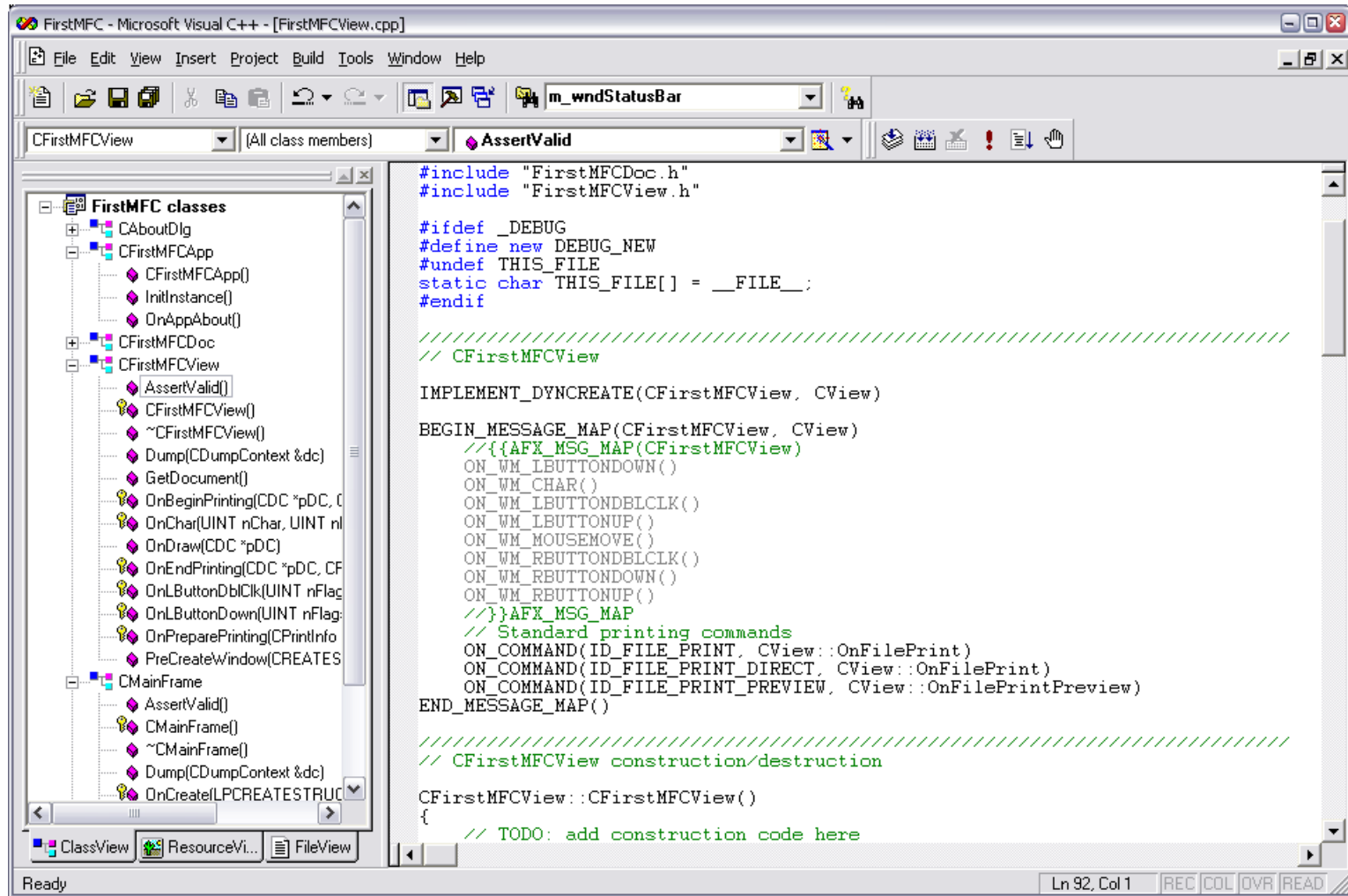
Message Map và ClassWizard



Hàm xử lý thông điệp chuột tương ứng



Hàm xử lý thông điệp chuột tương ứng



```
#include "FirstMFCDoc.h"
#include "FirstMFCView.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////
// CFirstMFCView

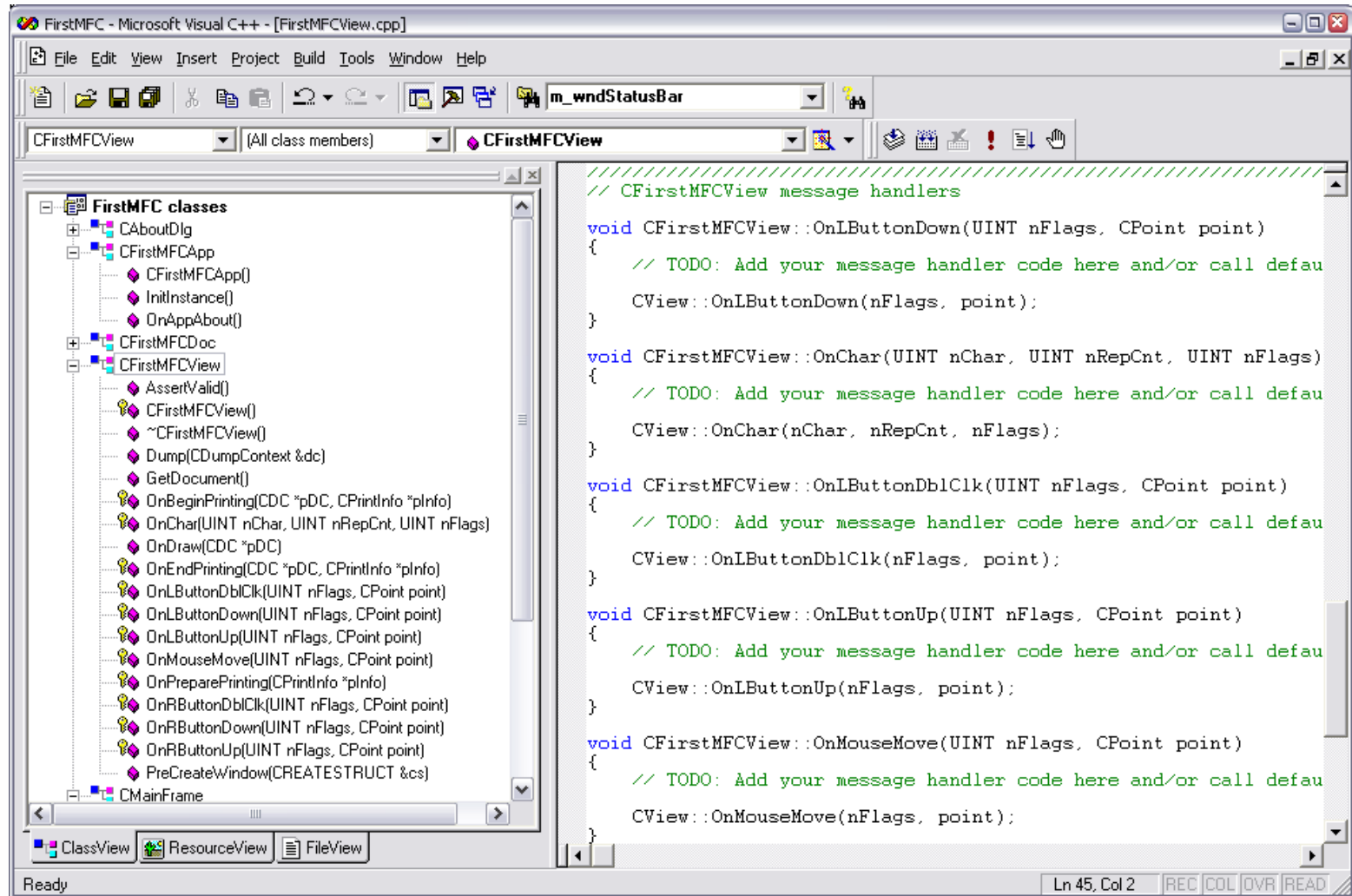
IMPLEMENT_DYNCREATE(CFirstMFCView, CView)

BEGIN_MESSAGE_MAP(CFirstMFCView, CView)
//{{AFX_MSG_MAP(CFirstMFCView)
ON_WM_LBUTTONDOWN()
ON_WM_CHAR()
ON_WM_LBUTTONDBLCLK()
ON_WM_LBUTTONUP()
ON_WM_MOUSEMOVE()
ON_WM_RBUTTONDBLCLK()
ON_WM_RBUTTONDOWN()
ON_WM_RBUTTONUP()
//}}AFX_MSG_MAP
// Standard printing commands
ON_COMMAND(ID_FILE_PRINT, CView::OnFilePrint)
ON_COMMAND(ID_FILE_PRINT_DIRECT, CView::OnFilePrint)
ON_COMMAND(ID_FILE_PRINT_PREVIEW, CView::OnFilePrintPreview)
END_MESSAGE_MAP()

////////////////////////////////////
// CFirstMFCView construction/destruction

CFirstMFCView::CFirstMFCView()
{
    // TODO: add construction code here
}
```


Hàm xử lý thông điệp chuột tương ứng



Hàm xử lý thông điệp chuột tương ứng

```
void CMapView::OnLButtonDown(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default

    CView::OnLButtonDown(nFlags, point);
}
```

```
void CMapView::OnLButtonUp(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default

    CView::OnLButtonUp(nFlags, point);
}
```

```
void CMapView::OnMouseMove(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default

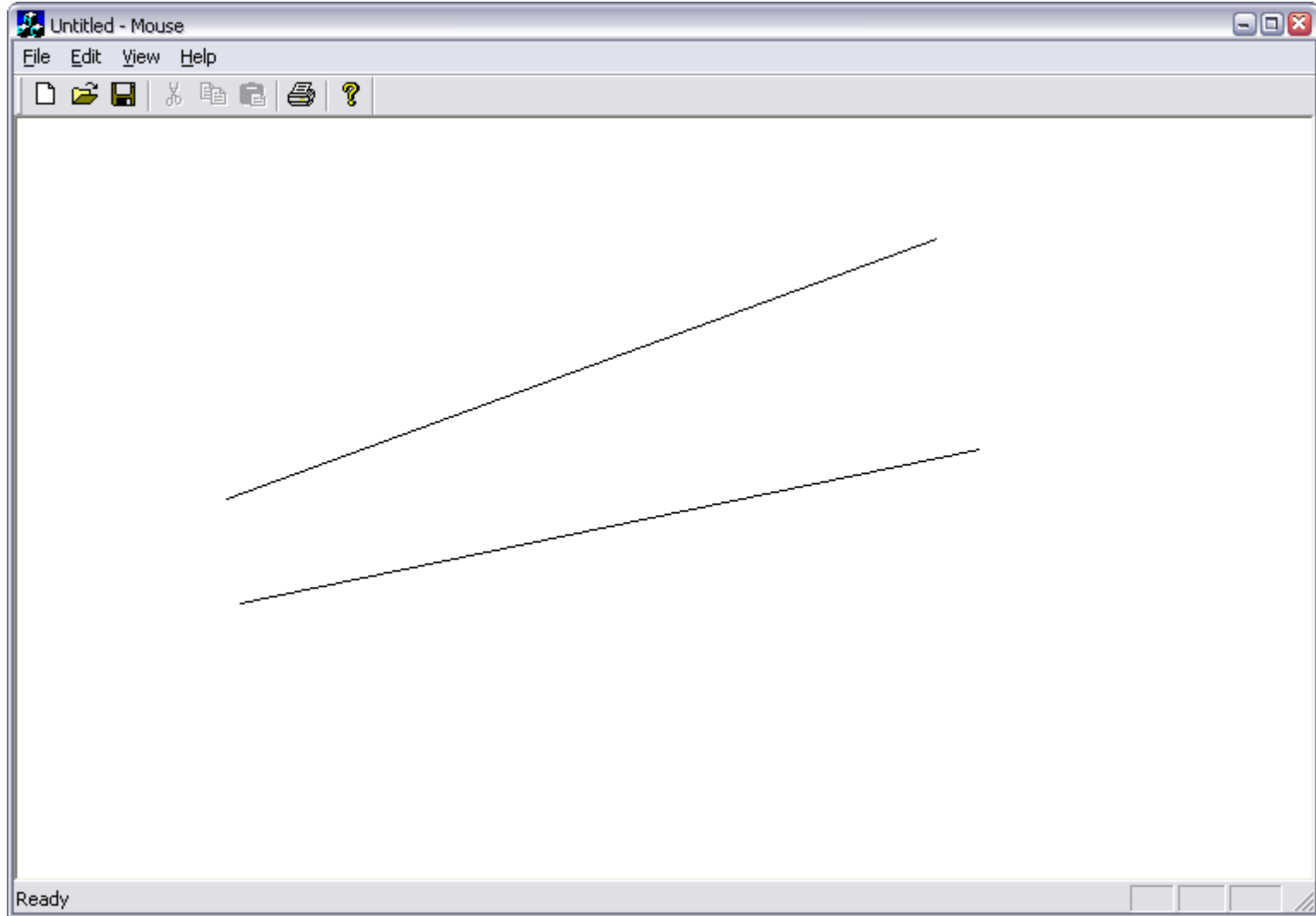
    CView::OnMouseMove(nFlags, point);
}
```

Hàm xử lý thông điệp chuột tương ứng

```
void CMouseView::OnLButtonDown(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    m_nStartX = point.x;
    m_nStartY = point.y;
    CView::OnLButtonDown(nFlags, point);
}
```

```
void CMouseView::OnLButtonUp(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    CClientDC dc(this);
    dc.MoveTo(m_nStartX,m_nStartY);
    dc.LineTo(point.x,point.y);
    CView::OnLButtonUp(nFlags, point);
}
```

Kết quả xử lý thông điệp chuột

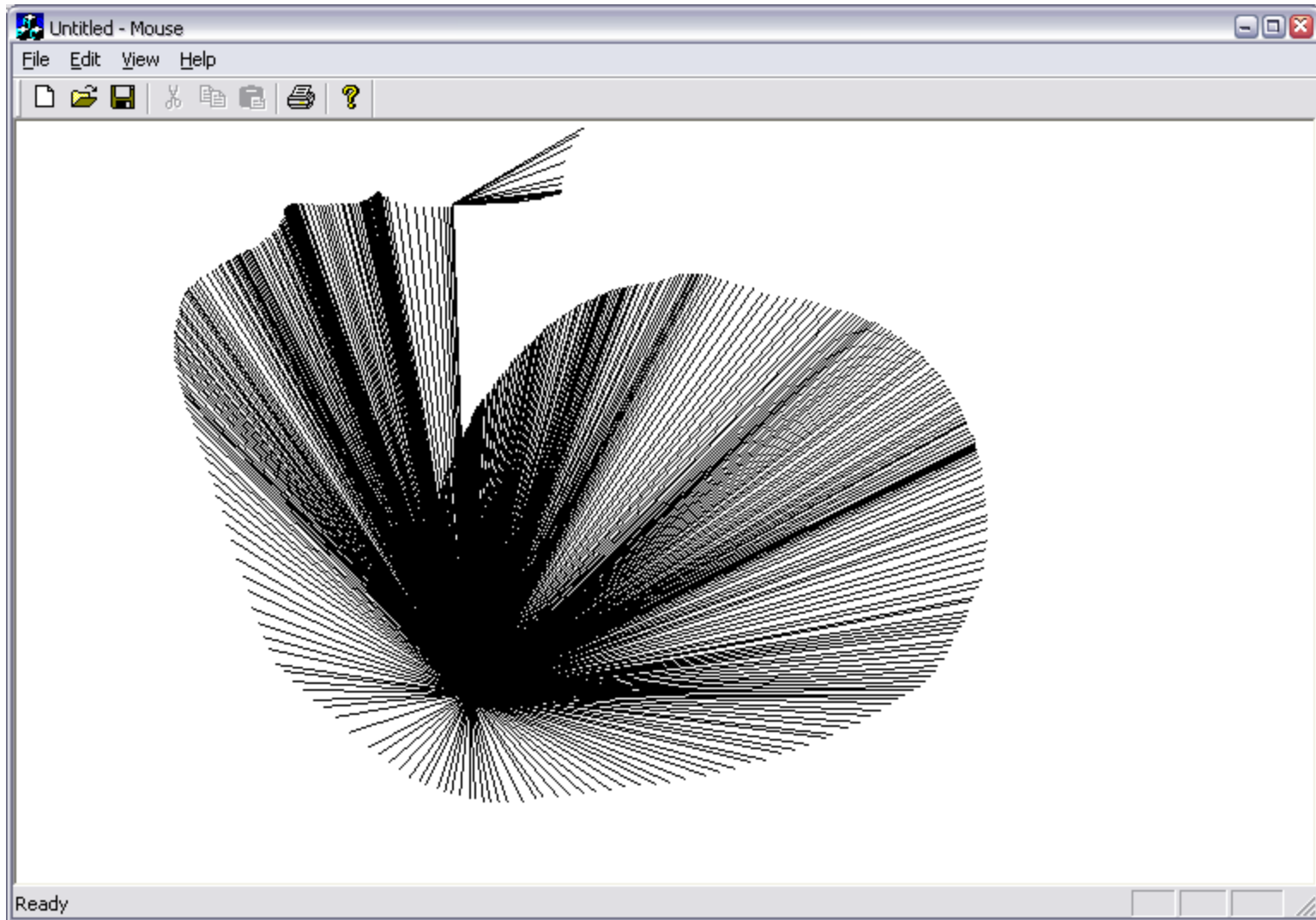


Hàm xử lý thông điệp chuột tương ứng

```
void CMouseView::OnLButtonDown(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    m_nStartX = point.x;
    m_nStartY = point.y;
    CView::OnLButtonDown(nFlags, point);
}
```

```
void CMouseView::OnMouseMove(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    CClientDC dc(this);
    dc.MoveTo(m_nStartX,m_nStartY);
    dc.LineTo(point.x,point.y);
    CView::OnLButtonUp(nFlags, point);
}
```

Kết quả xử lý thông điệp chuột

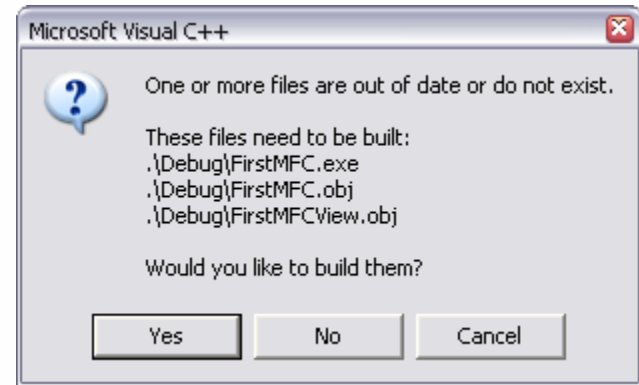
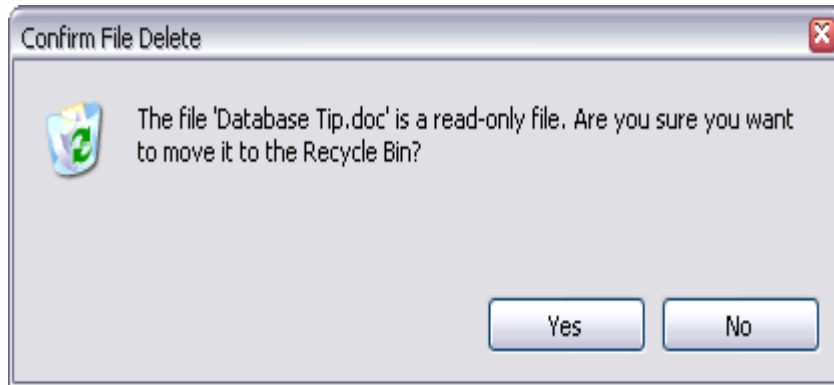


NỘI DUNG

- ❖ Giới thiệu về MFC
- ❖ Microsoft Visual C++ 6.0
- ❖ Các loại ứng dụng trong windows
- ❖ Chương trình MFC đầu tiên
- ❖ Xử lý chuột và bàn phím
- ❖ Message Box

Message Box - Hộp thông báo

- ❖ **Hộp thông báo được sử dụng để thông báo, trao đổi giữa ứng dụng với người dùng**



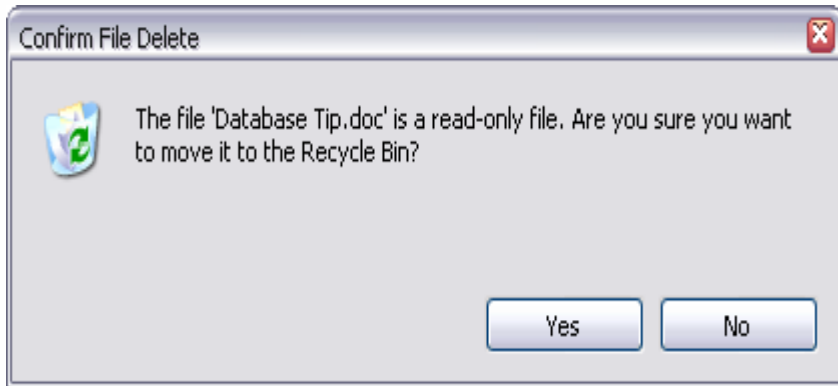
- ❖ **Để tạo lập hộp thông báo, ta sử dụng hàm MessageBox**

```
Cwnd::int MessageBox (LPCTSTR lpszText,  
                      LPCTSTR lpszCaption = NULL,  
                      UINT nType = MB_OK)
```


Các kiểu hộp thông báo và giá trị trả về tương ứng

Type	Buttons	Possible Return Codes
MB_ABORTRETRYIGNORE	Abort, Retry, Ignore	IDABORT, IDRETRY, IDIGNORE
MB_OK	OK	IDOK
MB_OKCANCEL	OK, Cancel	IDOK, IDCANCEL
MB_RETRYCANCEL	Retry, Cancel	IDRETRY, IDCANCEL
MB_YESNO	Yes, No	IDYES, IDNO
MB_YESNOCANCEL	Yes, No, Cancel	IDYES, IDNO, IDCANCEL

Message Box - Hộp thông báo

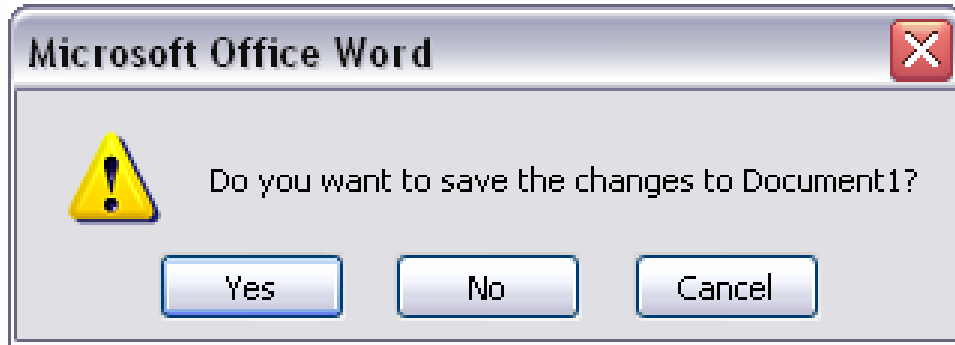


- ❖ Để tạo lập hộp thông báo trên ta sử dụng hàm **MessageBox** như sau:

```
int nRes = MessageBox ("The file 'Database Tip.doc' is a read-only  
file. Are you sure you want to move it to the Recycle Bin ?",  
"Confirm File Delete", MB_YESNO)
```

- ❖ Chọn "Yes" → **nRes = IDYES**
- ❖ Chọn "No" → **nRes = IDNO**

Message Box - Hộp thông báo



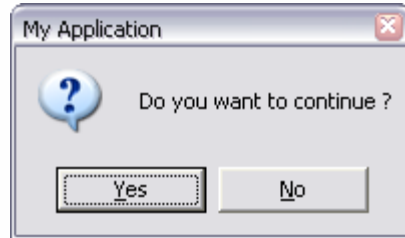
- ❖ Để tạo lập hộp thông báo trên ta sử dụng hàm **MessageBox** như sau:

```
int nRes = MessageBox ("Do you want to save the changes to  
Document1 ?", "Microsoft Office Word", MB_YESNOCANCEL)
```

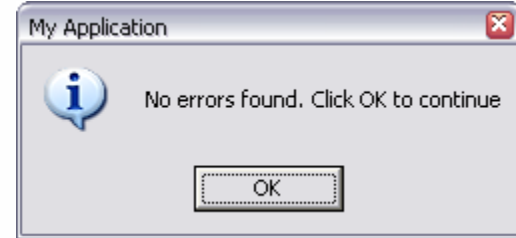
- ❖ Chọn "Yes" → **nRes = IDYES**
- ❖ Chọn "No" → **nRes = IDNO**
- ❖ Chọn "Cancel" → **nRes = IDCANCEL**

Các kiểu biểu tượng trong hộp thông báo

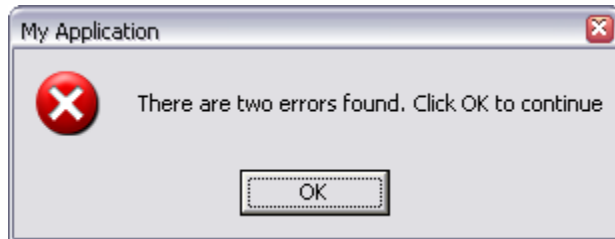
MB_ICONQUESTION



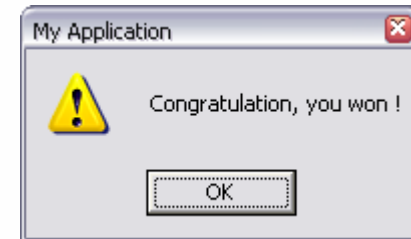
MB_ICONINFORMATION



MB_ICONSTOP



MB_ICONEXCLAMATION



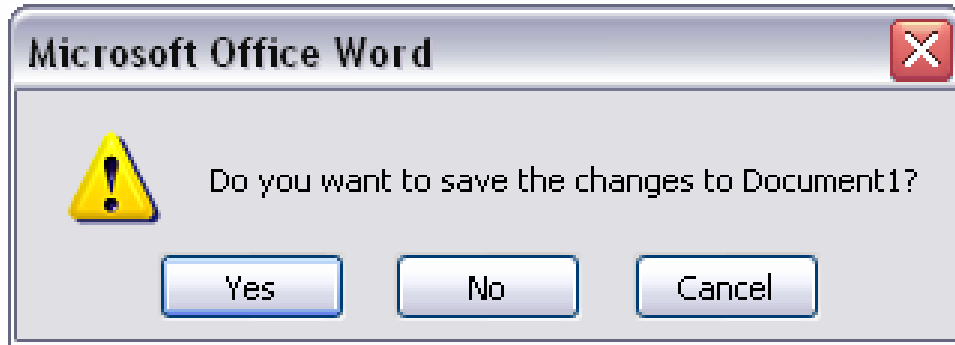
Message Box - Hộp thông báo



- ❖ Để tạo lập hộp thông báo với Icon như trên ta sử dụng hàm **MessageBox** như sau:

```
int nRes = MessageBox ("Do you want to continue ?", "My  
Application", MB_YESNO | MB_ICONQUESTION )
```

Message Box - Hộp thông báo



- ❖ Để tạo lập hộp thông báo với Icon như trên ta sử dụng hàm MessageBox như sau:

```
int nRes = MessageBox ("Do you want to save the changes to  
Document1 ?", "Microsoft Office Word", MB_YESNOCANCEL |  
MB_ICONEXCLAMATION )
```

