**暨南大学本科实验报告专用纸**

课程名称 windows编程实验 成绩评定

实验项目名称 MFC单文档界面框架程序 指导教师 刘晓翔

实验项目编号 实验二 实验项目类型 设计型 实验地点 304

学生姓名 陈 伟 学号 2022100092

学院 智能科学与工程学院 系 人工智能 专业 人工智能

实验时间 2025 年6月4日上午～6月14日上午温度 28 ℃湿度

1. **实验目的**

学习和掌握单文档界面框架程序（SDI）的基本结构与初始化过程。

1. **实验内容和要求**

利用Visual C++6.MFC APP Wizard生成一个单文档界面框架程序（SDI）。通过设置断点、单步跟踪调试的方式，剖析MFC框架程序的初始化过程

1. **主要仪器设备**

仪器：计算机

实验环境： Windows XP + Visual C++6.0

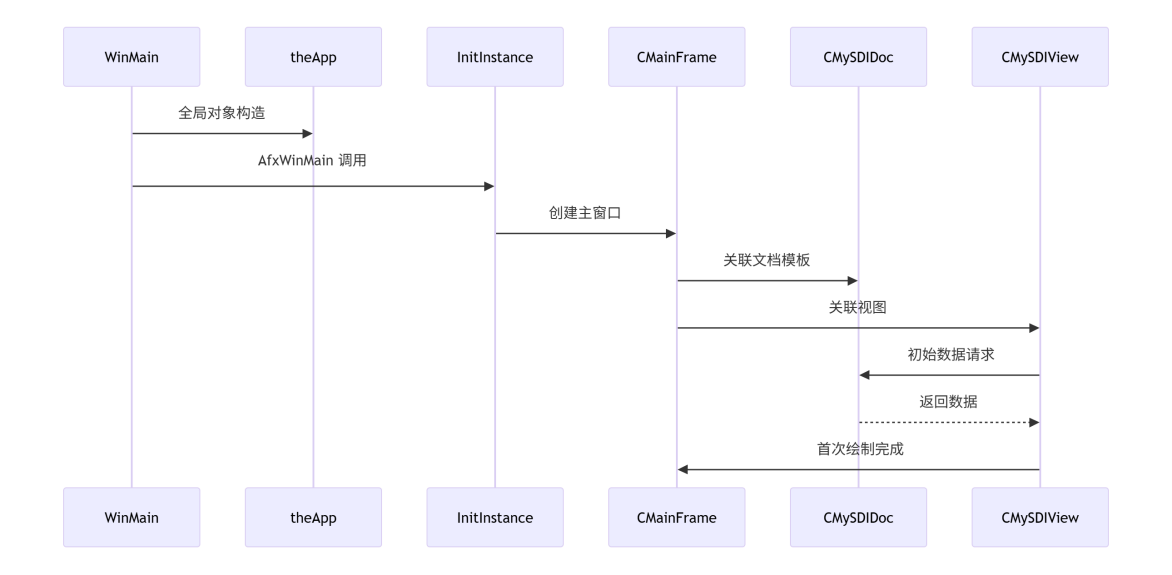
1. **实验原理**

MFC（MicrosoftFoundationClasses），即微软基础类库，是VC++（Visual

C++）开发环境的核心组成部分。MFC深刻地将C++的面向对象编程理念与WindowsAPI相融合，它几乎完整地用C++封装了WindowsSDK（软件开发工具包）中的结构和功能。MFC不仅为开发者提供了丰富的类库，还构建了一个全面的应用程序框架。这个框架极大地简化了软件开发的复杂性，自动处理了如窗口、工具栏、菜单等GUI元素的创建、注册和管理等繁琐工作，使开发者无需再陷入繁琐的WindowsAPI调用中。

通过MFC，软件开发者可以专注于实现业务逻辑，而无需担心底层GUI元素

的实现细节。这种封装和抽象极大地减少了开发者的工作量，提高了开发效率。



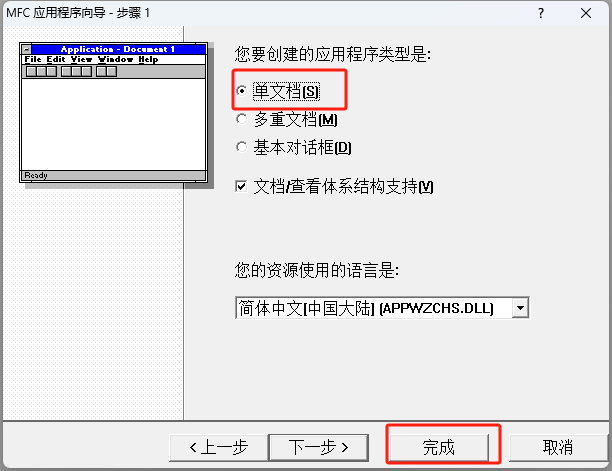
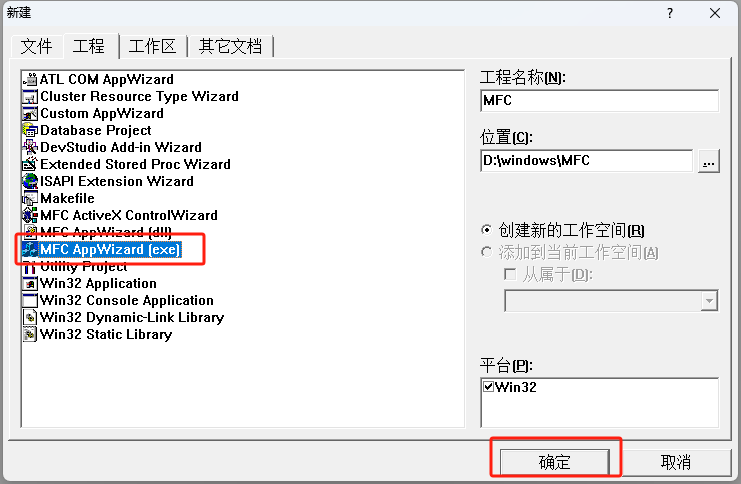
1. **实验步骤与调试**

第一步新建MFC单文档项目

打开 Visual C++ 6.0​​，选择 File → New → Projects。

选择 MFC AppWizard (exe)，输入项目名称（如 MySDI），点击 OK。

选择 Single Document（单文档），点击 Finish 生成默认框架

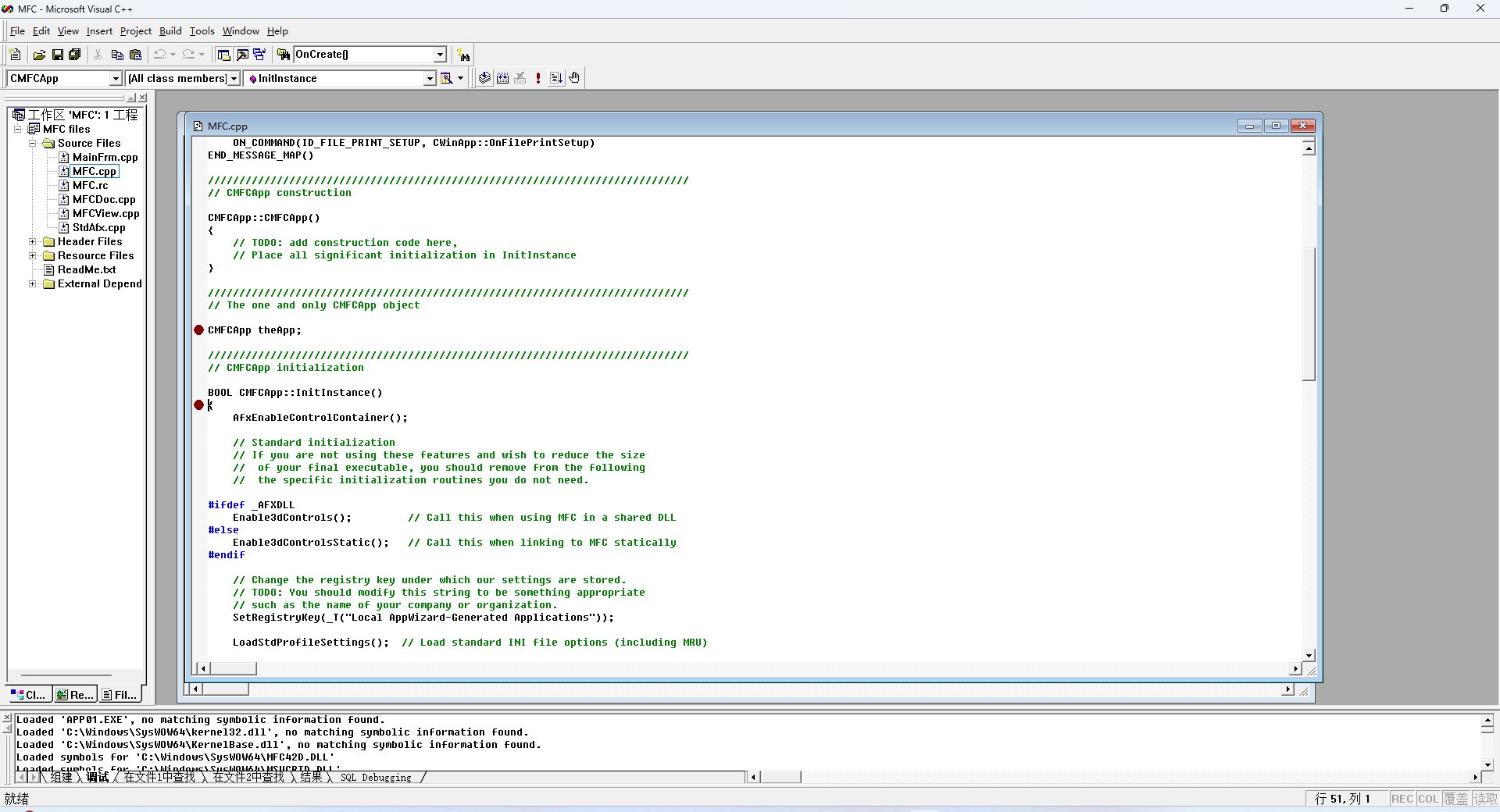


（1）全局对象 theApp 的构造​​

文件：MFC.cpp​​

断点位置：MFCApp theApp;（全局对象）

调试观察：



程序启动时，theApp 先于 WinMain 构造。

MFCSDIApp 继承自 CWinApp，初始化应用程序实例。

(2) WinMain 入口（MFC 内部实现）​​

文件：appmodul.cpp（MFC 内部）​​

断点位置：AfxWinMain（MFC 封装的 WinMain）

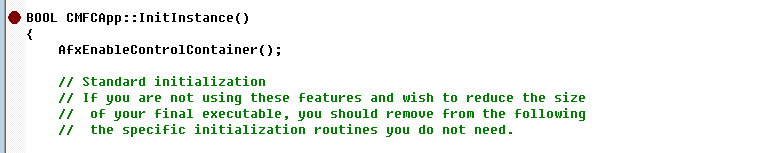
MFC 隐藏了标准 WinMain，实际入口是 AfxWinMain。

调用 AfxWinInit 初始化 MFC 内部状态（如注册窗口类、加载资源）。

(3) CMFCApp::InitInstance()​​

文件：MFC.cpp​​

断点位置：CMFCApp::InitInstance()

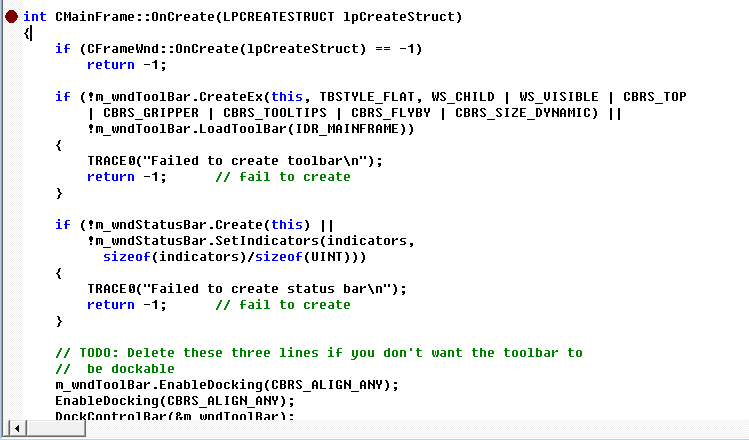


(4) CMainFrame::OnCreate()​​

文件：MainFrm.cpp​​

断点位置：CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)

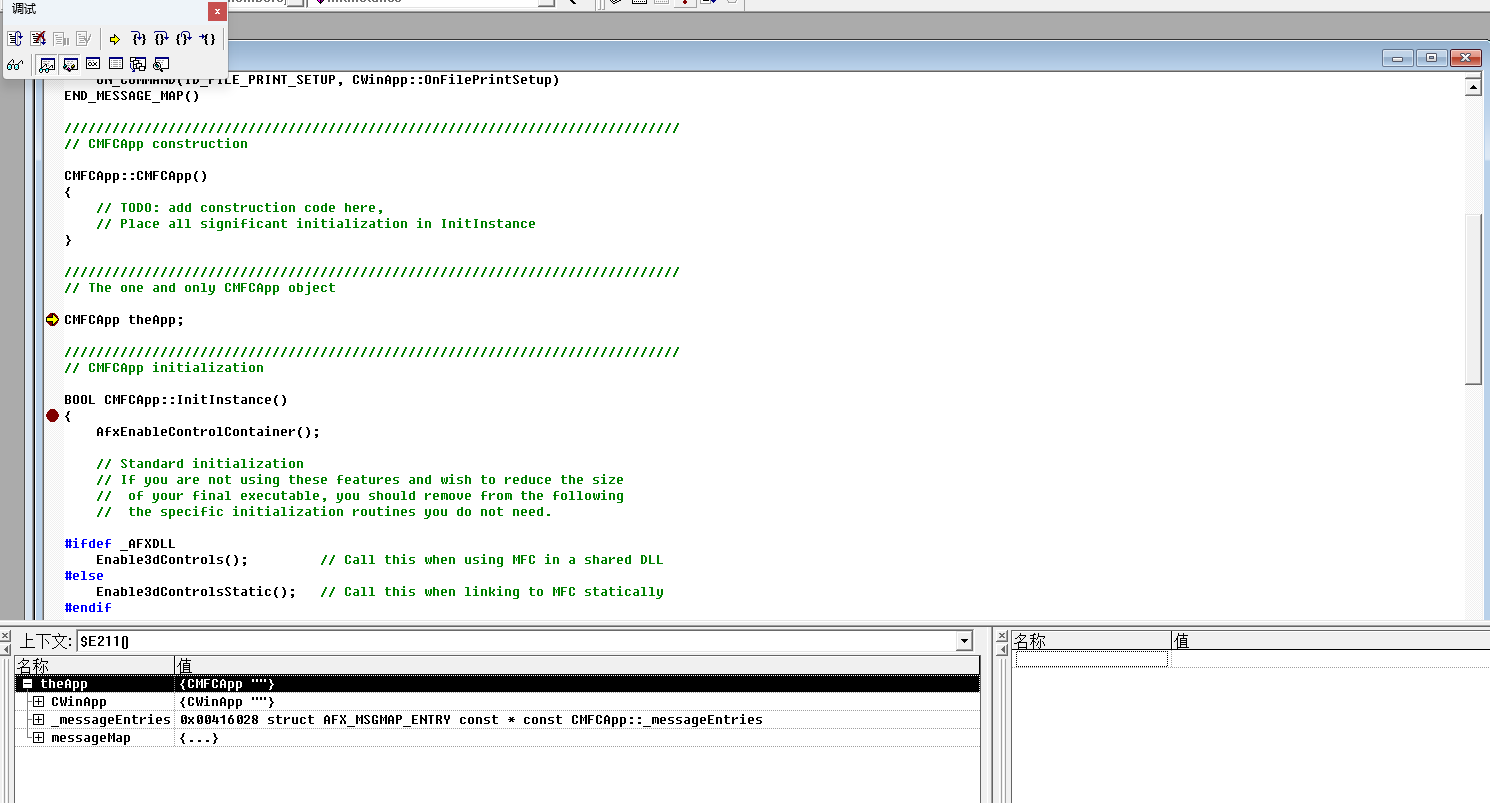
调试观察：

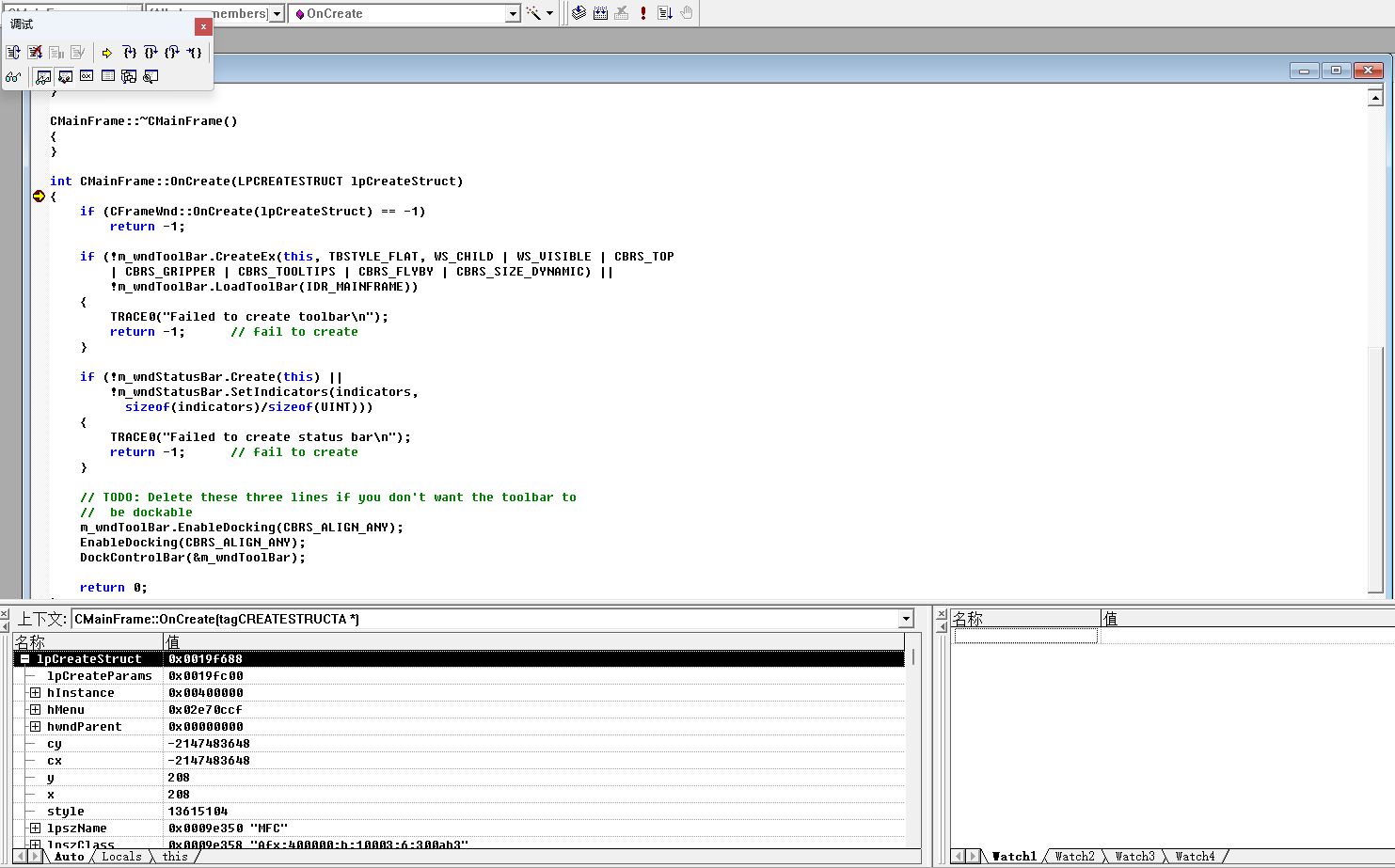


调用基类 CFrameWnd::OnCreate 创建窗口。

初始化工具栏、状态栏

1. **实验结果与分析**





MFC SDI 程序的初始化是自顶向下的：

全局CWinApp对象构造。

AfxWinMain 调用InitInstance。

注册文档模板并创建主框架窗口。

文档和视图动态绑定，完成首次绘制。

核心设计模式：

Document/View 架构：分离数据管理和显示逻辑。

框架窗口：CFrameWnd 管理用户界面（工具栏、状态栏）

1. **源程序**

MFC.cpp

// MFC.cpp : Defines the class behaviors for the application.

//

#include "stdafx.h"

#include "MFC.h"

#include "MainFrm.h"

#include "MFCDoc.h"

#include "MFCView.h"

#ifdef \_DEBUG

#define new DEBUG\_NEW

#undef THIS\_FILE

static char THIS\_FILE[] = \_\_FILE\_\_;

#endif

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

// CMFCApp

BEGIN\_MESSAGE\_MAP(CMFCApp, CWinApp)

//{{AFX\_MSG\_MAP(CMFCApp)

ON\_COMMAND(ID\_APP\_ABOUT, OnAppAbout)

// NOTE - the ClassWizard will add and remove mapping macros here.

// DO NOT EDIT what you see in these blocks of generated code!

//}}AFX\_MSG\_MAP

// Standard file based document commands

ON\_COMMAND(ID\_FILE\_NEW, CWinApp::OnFileNew)

ON\_COMMAND(ID\_FILE\_OPEN, CWinApp::OnFileOpen)

// Standard print setup command

ON\_COMMAND(ID\_FILE\_PRINT\_SETUP, CWinApp::OnFilePrintSetup)

END\_MESSAGE\_MAP()

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

// CMFCApp construction

CMFCApp::CMFCApp()

{

// TODO: add construction code here,

// Place all significant initialization in InitInstance

}

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

// The one and only CMFCApp object

CMFCApp theApp;

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

// CMFCApp initialization

BOOL CMFCApp::InitInstance()

{

AfxEnableControlContainer();

// Standard initialization

// If you are not using these features and wish to reduce the size

// of your final executable, you should remove from the following

// the specific initialization routines you do not need.

#ifdef \_AFXDLL

Enable3dControls(); // Call this when using MFC in a shared DLL

#else

Enable3dControlsStatic(); // Call this when linking to MFC statically

#endif

// Change the registry key under which our settings are stored.

// TODO: You should modify this string to be something appropriate

// such as the name of your company or organization.

SetRegistryKey(\_T("Local AppWizard-Generated Applications"));

LoadStdProfileSettings(); // Load standard INI file options (including MRU)

// Register the application's document templates. Document templates

// serve as the connection between documents, frame windows and views.

CSingleDocTemplate\* pDocTemplate;

pDocTemplate = new CSingleDocTemplate(

IDR\_MAINFRAME,

RUNTIME\_CLASS(CMFCDoc),

RUNTIME\_CLASS(CMainFrame), // main SDI frame window

RUNTIME\_CLASS(CMFCView));

AddDocTemplate(pDocTemplate);

// Parse command line for standard shell commands, DDE, file open

CCommandLineInfo cmdInfo;

ParseCommandLine(cmdInfo);

// Dispatch commands specified on the command line

if (!ProcessShellCommand(cmdInfo))

return FALSE;

// The one and only window has been initialized, so show and update it.

m\_pMainWnd->ShowWindow(SW\_SHOW);

m\_pMainWnd->UpdateWindow();

return TRUE;

}

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

// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog

{

public:

CAboutDlg();

// Dialog Data

//{{AFX\_DATA(CAboutDlg)

enum { IDD = IDD\_ABOUTBOX };

//}}AFX\_DATA

// ClassWizard generated virtual function overrides

//{{AFX\_VIRTUAL(CAboutDlg)

protected:

virtual void DoDataExchange(CDataExchange\* pDX); // DDX/DDV support

//}}AFX\_VIRTUAL

// Implementation

protected:

//{{AFX\_MSG(CAboutDlg)

// No message handlers

//}}AFX\_MSG

DECLARE\_MESSAGE\_MAP()

};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)

{

//{{AFX\_DATA\_INIT(CAboutDlg)

//}}AFX\_DATA\_INIT

}

void CAboutDlg::DoDataExchange(CDataExchange\* pDX)

{

CDialog::DoDataExchange(pDX);

//{{AFX\_DATA\_MAP(CAboutDlg)

//}}AFX\_DATA\_MAP

}

BEGIN\_MESSAGE\_MAP(CAboutDlg, CDialog)

//{{AFX\_MSG\_MAP(CAboutDlg)

// No message handlers

//}}AFX\_MSG\_MAP

END\_MESSAGE\_MAP()

// App command to run the dialog

void CMFCApp::OnAppAbout()

{

CAboutDlg aboutDlg;

aboutDlg.DoModal();

}

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

// CMFCApp message handlers

Mainframe.cpp

// MainFrm.cpp : implementation of the CMainFrame class

//

#include "stdafx.h"

#include "MFC.h"

#include "MainFrm.h"

#ifdef \_DEBUG

#define new DEBUG\_NEW

#undef THIS\_FILE

static char THIS\_FILE[] = \_\_FILE\_\_;

#endif

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

// CMainFrame

IMPLEMENT\_DYNCREATE(CMainFrame, CFrameWnd)

BEGIN\_MESSAGE\_MAP(CMainFrame, CFrameWnd)

//{{AFX\_MSG\_MAP(CMainFrame)

// NOTE - the ClassWizard will add and remove mapping macros here.

// DO NOT EDIT what you see in these blocks of generated code !

ON\_WM\_CREATE()

//}}AFX\_MSG\_MAP

END\_MESSAGE\_MAP()

static UINT indicators[] =

{

ID\_SEPARATOR, // status line indicator

ID\_INDICATOR\_CAPS,

ID\_INDICATOR\_NUM,

ID\_INDICATOR\_SCRL,

};

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

// CMainFrame construction/destruction

CMainFrame::CMainFrame()

{

// TODO: add member initialization code here

}

CMainFrame::~CMainFrame()

{

}

int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)

{

if (CFrameWnd::OnCreate(lpCreateStruct) == -1)

return -1;

if (!m\_wndToolBar.CreateEx(this, TBSTYLE\_FLAT, WS\_CHILD | WS\_VISIBLE | CBRS\_TOP

| CBRS\_GRIPPER | CBRS\_TOOLTIPS | CBRS\_FLYBY | CBRS\_SIZE\_DYNAMIC) ||

!m\_wndToolBar.LoadToolBar(IDR\_MAINFRAME))

{

TRACE0("Failed to create toolbar\n");

return -1; // fail to create

}

if (!m\_wndStatusBar.Create(this) ||

!m\_wndStatusBar.SetIndicators(indicators,

sizeof(indicators)/sizeof(UINT)))

{

TRACE0("Failed to create status bar\n");

return -1; // fail to create

}

// TODO: Delete these three lines if you don't want the toolbar to

// be dockable

m\_wndToolBar.EnableDocking(CBRS\_ALIGN\_ANY);

EnableDocking(CBRS\_ALIGN\_ANY);

DockControlBar(&m\_wndToolBar);

return 0;

}

BOOL CMainFrame::PreCreateWindow(CREATESTRUCT& cs)

{

if( !CFrameWnd::PreCreateWindow(cs) )

return FALSE;

// TODO: Modify the Window class or styles here by modifying

// the CREATESTRUCT cs

return TRUE;

}

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

// CMainFrame diagnostics

#ifdef \_DEBUG

void CMainFrame::AssertValid() const

{

CFrameWnd::AssertValid();

}

void CMainFrame::Dump(CDumpContext& dc) const

{

CFrameWnd::Dump(dc);

}

#endif //\_DEBUG

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

// CMainFrame message handlers