# 这一节我们来学习文件操作是四种方法

|  |
| --- |
|  |

# 演练

## 1.创建一个MFC应用程序项目，取名: Lesson15-4way-file-op选择基于对话框的应用程序，点击完成

|  |
| --- |
|  |

## 2.设计界面，把原来的对话框上面的控件删除，然后添加一些按钮和一个文本框，把文本框拉大，在属性面板中，把Multiline属性设置为true，把Want Return设置为true，把Vertical scroll也设置为true

|  |
| --- |
|  |

## 3.然后修改按钮的ID和按钮文字

|  |
| --- |
|  |

## 4.为每一个按钮添加点击事件处理代码

|  |
| --- |
|  |

## 5.我们先来做c语言的，先读取一个叫做harry.txt的文本文件，在MFC的程序里面，你可以使用c或者c++的函数，也可以使用win32的函数，也可以都使用。没有问题不过你必须知道这么使用。c语言读取文件的代码如下：

|  |
| --- |
| void CLesson154wayfileopDlg::OnBnClickedBtnCWay()  {  // TODO: 在此添加控件通知处理程序代码  //0.先清空编辑框内容  //m\_edit.SetWindowTextW(\_T(""));  SetDlgItemText(IDC\_EDIT1,\_T(""));  FILE \*pFile;  int i, numread;  char line[256];  //1.打开文件  pFile = fopen("harry.txt","r");  CString content(\_T("\*\*\*\*\*\*\*\*\*C\*\*\*\*\*\*\*\*\*\r\n"));    //2.读取文件内容，需要使用循环  while(fgets(line,256,pFile)!=NULL)  {  //3.将读取到的每一行添加到content中  content +=line;  content +=\_T("\r\n");  }  //4.读取完毕需要关闭文件句柄  fclose(pFile);  //5.显示内容到编辑框  //m\_edit.SetWindowTextW(content);//可以这么写  SetDlgItemText(IDC\_EDIT1,content); //也可以这么写  } |

## 6.然后我们来做c++的方式，注意需要包含<fstream>

|  |
| --- |
| void CLesson154wayfileopDlg::OnBnClickedBtnCppWay()  {  // TODO: 在此添加控件通知处理程序代码  char line[256];  //1.先清空编辑框内容  //m\_edit.SetWindowTextW(\_T(""));  SetDlgItemText(IDC\_EDIT1,\_T(""));  CString content(\_T("\*\*\*\*\*\*\*\*\*CPP Way Of Read File\*\*\*\*\*\*\*\*\*\r\n"));  //2.创建一个文件输入流对象  ifstream ifs("harry.txt");  //3.读取文件，需要利用循环，把读取到的每一行添加到content中  while(ifs.getline(line,256)!=NULL)  {  content += line;  content +=\_T("\r\n");  }  //4.关闭我流  ifs.close();  //5.显示内容到编辑框  //m\_edit.SetWindowTextW(content);//可以这么写  SetDlgItemText(IDC\_EDIT1,content); //也可以这么写  } |

## 7.Win32方式的代码如下，注意需要先清空字符数组使用ZeroMemory函数，然后在显示内容之前需要把“\n“替换为”\r\n”,还有，ReadFile函数会把文件内容一次全部读完，需要定义大一点的字符数组。

|  |
| --- |
| void CLesson154wayfileopDlg::OnBnClickedBtnWin32Way()  {  // TODO: 在此添加控件通知处理程序代码  //win32方式可以一次把整个文件读完。  HANDLE hFile;  CHAR szBuffer[4096];  DWORD dwRead;  ZeroMemory(szBuffer,4096);  //1.先清空编辑框内容  //m\_edit.SetWindowTextW(\_T(""));  SetDlgItemText(IDC\_EDIT1,\_T(""));  CString content(\_T("\*\*\*\*\*\*\*\*\*Win32 Way Of Read File\*\*\*\*\*\*\*\*\*\r\n"));  //2.打开文件，使用CreateFile函数  hFile = CreateFile(\_T("harry.txt"),GENERIC\_READ,FILE\_SHARE\_READ,NULL,  OPEN\_EXISTING,FILE\_ATTRIBUTE\_NORMAL,NULL);  if(INVALID\_HANDLE\_VALUE == hFile)  {  MessageBox(\_T("Error Creating File"));  }  //3.读取文件，使用ReadFile函数  ReadFile(hFile,szBuffer,4096,&dwRead,NULL);  content +=szBuffer;  content +=\_T("\r\n");  //content.Replace(\_T("\n"),\_T("\r\n"));  //4.关闭文件句柄  CloseHandle(hFile);  //5.显示内容到编辑框  //m\_edit.SetWindowTextW(content);//可以这么写  SetDlgItemText(IDC\_EDIT1,content); //也可以这么写  } |

## 8.下面我们来使用MFC的文件相关类来读取文件，这里使用CStdioFile类

|  |
| --- |
| void CLesson154wayfileopDlg::OnBnClickedBtnMfcWay()  {  // TODO: 在此添加控件通知处理程序代码  CString line; //保存每一次读取的一行文本的变量  //1.先清空编辑框内容  //m\_edit.SetWindowTextW(\_T(""));  //SetDlgItemText(IDC\_EDIT1,\_T(""));  CString content(\_T("\*\*\*\*\*\*\*\*\*MFC Way Of Read File\*\*\*\*\*\*\*\*\*\r\n"));  //2.读取文件，使用CStdioFile类，需要循环  CStdioFile file(\_T("harry.txt"),CFile::modeRead);  while(file.ReadString(line))  {  content +=line;  content +=\_T("\r\n");  }  //3.关闭文件句柄  file.Close();    //4.显示内容到编辑框  //m\_edit.SetWindowTextW(content);//可以这么写  SetDlgItemText(IDC\_EDIT1,content); //也可以这么写  } |

# 这一节的学习到此为止，Lesson15-4way-file-opDlg.cpp的完整代码如下

|  |
| --- |
| // Lesson15-4way-file-opDlg.cpp : 实现文件  //  #include "stdafx.h"  #include "Lesson15-4way-file-op.h"  #include "Lesson15-4way-file-opDlg.h"  #include<fstream>  #ifdef \_DEBUG  #define new DEBUG\_NEW  #endif  using namespace std;  // 用于应用程序“关于”菜单项的CAboutDlg 对话框  class CAboutDlg : public CDialog  {  public:  CAboutDlg();  // 对话框数据  enum { IDD = IDD\_ABOUTBOX };  protected:  virtual void DoDataExchange(CDataExchange\* pDX); // DDX/DDV 支持  // 实现  protected:  DECLARE\_MESSAGE\_MAP()  };  CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)  {  }  void CAboutDlg::DoDataExchange(CDataExchange\* pDX)  {  CDialog::DoDataExchange(pDX);  }  BEGIN\_MESSAGE\_MAP(CAboutDlg, CDialog)  END\_MESSAGE\_MAP()  // CLesson154wayfileopDlg 对话框  CLesson154wayfileopDlg::CLesson154wayfileopDlg(CWnd\* pParent /\*=NULL\*/)  : CDialog(CLesson154wayfileopDlg::IDD, pParent)  {  m\_hIcon = AfxGetApp()->LoadIcon(IDR\_MAINFRAME);  }  void CLesson154wayfileopDlg::DoDataExchange(CDataExchange\* pDX)  {  CDialog::DoDataExchange(pDX);  DDX\_Control(pDX, IDC\_EDIT1, m\_edit);  }  BEGIN\_MESSAGE\_MAP(CLesson154wayfileopDlg, CDialog)  ON\_WM\_SYSCOMMAND()  ON\_WM\_PAINT()  ON\_WM\_QUERYDRAGICON()  //}}AFX\_MSG\_MAP  ON\_BN\_CLICKED(IDC\_BTN\_C\_WAY, &CLesson154wayfileopDlg::OnBnClickedBtnCWay)  ON\_BN\_CLICKED(IDC\_BTN\_CPP\_WAY, &CLesson154wayfileopDlg::OnBnClickedBtnCppWay)  ON\_BN\_CLICKED(IDC\_BTN\_WIN32\_WAY, &CLesson154wayfileopDlg::OnBnClickedBtnWin32Way)  ON\_BN\_CLICKED(IDC\_BTN\_MFC\_WAY, &CLesson154wayfileopDlg::OnBnClickedBtnMfcWay)  END\_MESSAGE\_MAP()  // CLesson154wayfileopDlg 消息处理程序  BOOL CLesson154wayfileopDlg::OnInitDialog()  {  CDialog::OnInitDialog();  // 将“关于...”菜单项添加到系统菜单中。  // IDM\_ABOUTBOX 必须在系统命令范围内。  ASSERT((IDM\_ABOUTBOX & 0xFFF0) == IDM\_ABOUTBOX);  ASSERT(IDM\_ABOUTBOX < 0xF000);  CMenu\* pSysMenu = GetSystemMenu(FALSE);  if (pSysMenu != NULL)  {  CString strAboutMenu;  strAboutMenu.LoadString(IDS\_ABOUTBOX);  if (!strAboutMenu.IsEmpty())  {  pSysMenu->AppendMenu(MF\_SEPARATOR);  pSysMenu->AppendMenu(MF\_STRING, IDM\_ABOUTBOX, strAboutMenu);  }  }  // 设置此对话框的图标。当应用程序主窗口不是对话框时，框架将自动  // 执行此操作  SetIcon(m\_hIcon, TRUE); // 设置大图标  SetIcon(m\_hIcon, FALSE); // 设置小图标  // TODO: 在此添加额外的初始化代码  return TRUE; // 除非将焦点设置到控件，否则返回TRUE  }  void CLesson154wayfileopDlg::OnSysCommand(UINT nID, LPARAM lParam)  {  if ((nID & 0xFFF0) == IDM\_ABOUTBOX)  {  CAboutDlg dlgAbout;  dlgAbout.DoModal();  }  else  {  CDialog::OnSysCommand(nID, lParam);  }  }  // 如果向对话框添加最小化按钮，则需要下面的代码  // 来绘制该图标。对于使用文档/视图模型的MFC 应用程序，  // 这将由框架自动完成。  void CLesson154wayfileopDlg::OnPaint()  {  if (IsIconic())  {  CPaintDC dc(this); // 用于绘制的设备上下文  SendMessage(WM\_ICONERASEBKGND, reinterpret\_cast<WPARAM>(dc.GetSafeHdc()), 0);  // 使图标在工作区矩形中居中  int cxIcon = GetSystemMetrics(SM\_CXICON);  int cyIcon = GetSystemMetrics(SM\_CYICON);  CRect rect;  GetClientRect(&rect);  int x = (rect.Width() - cxIcon + 1) / 2;  int y = (rect.Height() - cyIcon + 1) / 2;  // 绘制图标  dc.DrawIcon(x, y, m\_hIcon);  }  else  {  CDialog::OnPaint();  }  }  //当用户拖动最小化窗口时系统调用此函数取得光标  //显示。  HCURSOR CLesson154wayfileopDlg::OnQueryDragIcon()  {  return static\_cast<HCURSOR>(m\_hIcon);  }  void CLesson154wayfileopDlg::OnBnClickedBtnCWay()  {  // TODO: 在此添加控件通知处理程序代码  //0.先清空编辑框内容  //m\_edit.SetWindowTextW(\_T(""));  //SetDlgItemText(IDC\_EDIT1,\_T(""));  FILE \*pFile;  char line[256];  //1.打开文件  pFile = fopen("harry.txt","r");  CString content(\_T("\*\*\*\*\*\*\*\*\*C\*\*\*\*\*\*\*\*\*\r\n"));    //2.读取文件内容，需要使用循环  while(fgets(line,256,pFile)!=NULL)  {  //3.将读取到的每一行添加到content中  content +=line;  content +=\_T("\r\n");  }  //4.读取完毕需要关闭文件句柄  fclose(pFile);  //5.显示内容到编辑框  //m\_edit.SetWindowTextW(content);//可以这么写  SetDlgItemText(IDC\_EDIT1,content); //也可以这么写  }  void CLesson154wayfileopDlg::OnBnClickedBtnCppWay()  {  // TODO: 在此添加控件通知处理程序代码  char line[256];  //1.先清空编辑框内容  //m\_edit.SetWindowTextW(\_T(""));  //SetDlgItemText(IDC\_EDIT1,\_T(""));  CString content(\_T("\*\*\*\*\*\*\*\*\*CPP Way Of Read File\*\*\*\*\*\*\*\*\*\r\n"));  //2.创建一个文件输入流对象  ifstream ifs("harry.txt");  //3.读取文件，需要利用循环，把读取到的每一行添加到content中  while(ifs.getline(line,256)!=NULL)  {  content += line;  content +=\_T("\r\n");  }  //4.关闭我流  ifs.close();  //5.显示内容到编辑框  //m\_edit.SetWindowTextW(content);//可以这么写  SetDlgItemText(IDC\_EDIT1,content); //也可以这么写  }  void CLesson154wayfileopDlg::OnBnClickedBtnWin32Way()  {  // TODO: 在此添加控件通知处理程序代码  //win32方式可以一次把整个文件读完。  HANDLE hFile;  CHAR szBuffer[4096];  DWORD dwRead;  ZeroMemory(szBuffer,4096);  //1.先清空编辑框内容  //m\_edit.SetWindowTextW(\_T(""));  //SetDlgItemText(IDC\_EDIT1,\_T(""));  CString content(\_T("\*\*\*\*\*\*\*\*\*Win32 Way Of Read File\*\*\*\*\*\*\*\*\*\r\n"));  //2.打开文件，使用CreateFile函数  hFile = CreateFile(\_T("harry.txt"),GENERIC\_READ,FILE\_SHARE\_READ,NULL,  OPEN\_EXISTING,FILE\_ATTRIBUTE\_NORMAL,NULL);  if(INVALID\_HANDLE\_VALUE == hFile)  {  MessageBox(\_T("Error Creating File"));  }  //3.读取文件，使用ReadFile函数  ReadFile(hFile,szBuffer,4096,&dwRead,NULL);  content +=szBuffer;  content +=\_T("\r\n");  //content.Replace(\_T("\n"),\_T("\r\n"));  //4.关闭文件句柄  CloseHandle(hFile);  //5.显示内容到编辑框  //m\_edit.SetWindowTextW(content);//可以这么写  SetDlgItemText(IDC\_EDIT1,content); //也可以这么写  }  void CLesson154wayfileopDlg::OnBnClickedBtnMfcWay()  {  // TODO: 在此添加控件通知处理程序代码  CString line; //保存每一次读取的一行文本的变量  //1.先清空编辑框内容  //m\_edit.SetWindowTextW(\_T(""));  //SetDlgItemText(IDC\_EDIT1,\_T(""));  CString content(\_T("\*\*\*\*\*\*\*\*\*MFC Way Of Read File\*\*\*\*\*\*\*\*\*\r\n"));  //2.读取文件，使用CStdioFile类，需要循环  CStdioFile file(\_T("harry.txt"),CFile::modeRead);  while(file.ReadString(line))  {  content +=line;  content +=\_T("\r\n");  }  //3.关闭文件句柄  file.Close();    //4.显示内容到编辑框  //m\_edit.SetWindowTextW(content);//可以这么写  SetDlgItemText(IDC\_EDIT1,content); //也可以这么写  } |