# 这一节我们来学习用MFC来创建对话框窗口

## 1.打开vs2008，新建-》项目-》c++-》MFC-MFC应用程序，项目名称：Lesson8-MFC-Dialog，点击完成

|  |
| --- |
|  |

## 2.然后会弹出创建项目向导，next-》选择基于对话框的应用程序

|  |
| --- |
|  |

## 3.点击完成，就会生成一个对话框应用程序

|  |
| --- |
|  |

## 4.我们把对话框默认的2个按钮输出，然后给对话框添加一个图片框控件，然后再添加2个按钮，点击每一个按钮在图片框中显示不一样的位图，注意，有一个问题非常坑爹，就是使用图片控件的SetBitmap设置图片程序老是崩溃，使用我们使用曲线救国，定义一个位图画刷，然后用整个画刷来填充图片框，完整代码如下

|  |
| --- |
| // MfcDlgDlg.cpp: 实现文件  //  #include "pch.h"  #include "framework.h"  #include "MfcDlg.h"  #include "MfcDlgDlg.h"  #include "afxdialogex.h"  #ifdef \_DEBUG  #define new DEBUG\_NEW  #endif  // 用于应用程序“关于”菜单项的 CAboutDlg 对话框  class CAboutDlg : public CDialogEx  {  public:  CAboutDlg();  // 对话框数据  #ifdef AFX\_DESIGN\_TIME  enum { IDD = IDD\_ABOUTBOX };  #endif  protected:  virtual void DoDataExchange(CDataExchange\* pDX); // DDX/DDV 支持  // 实现  protected:  DECLARE\_MESSAGE\_MAP()  };  CAboutDlg::CAboutDlg() : CDialogEx(IDD\_ABOUTBOX)  {  }  void CAboutDlg::DoDataExchange(CDataExchange\* pDX)  {  CDialogEx::DoDataExchange(pDX);  }  BEGIN\_MESSAGE\_MAP(CAboutDlg, CDialogEx)  END\_MESSAGE\_MAP()  // CMfcDlgDlg 对话框  CMfcDlgDlg::CMfcDlgDlg(CWnd\* pParent /\*=nullptr\*/)  : CDialogEx(IDD\_MFCDLG\_DIALOG, pParent)  {  m\_hIcon = AfxGetApp()->LoadIcon(IDR\_MAINFRAME);  }  void CMfcDlgDlg::DoDataExchange(CDataExchange\* pDX)  {  CDialogEx::DoDataExchange(pDX);  DDX\_Control(pDX, IDC\_PIC\_STATIC, m\_picCtrl);  }  BEGIN\_MESSAGE\_MAP(CMfcDlgDlg, CDialogEx)  ON\_WM\_SYSCOMMAND()  ON\_WM\_PAINT()  ON\_WM\_QUERYDRAGICON()  ON\_BN\_CLICKED(IDC\_BUTTON2, &CMfcDlgDlg::OnBnClickedButton2)  ON\_BN\_CLICKED(IDC\_BUTTON1, &CMfcDlgDlg::OnBnClickedButton1)  ON\_WM\_CREATE()  END\_MESSAGE\_MAP()  // CMfcDlgDlg 消息处理程序  BOOL CMfcDlgDlg::OnInitDialog()  {  CDialogEx::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 != nullptr)  {  BOOL bNameValid;  CString strAboutMenu;  bNameValid = strAboutMenu.LoadString(IDS\_ABOUTBOX);  ASSERT(bNameValid);  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 CMfcDlgDlg::OnSysCommand(UINT nID, LPARAM lParam)  {  if ((nID & 0xFFF0) == IDM\_ABOUTBOX)  {  CAboutDlg dlgAbout;  dlgAbout.DoModal();  }  else  {  CDialogEx::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 CMfcDlgDlg::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  {  CClientDC dc(&m\_picCtrl);  CRect rc;  m\_picCtrl.GetClientRect(&rc);  CBrush\* brush = CBrush::FromHandle(CreateSolidBrush(RGB(255, 255, 255)));  dc.FillRect(&rc, brush);  CDialogEx::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 CMfcDlgDlg::OnQueryDragIcon()  {  return static\_cast<HCURSOR>(m\_hIcon);  }  void CMfcDlgDlg::OnBnClickedButton2()  {  // TODO: 在此添加控件通知处理程序代码  CClientDC dc(&m\_picCtrl);  CRect rc;  m\_picCtrl.GetClientRect(&rc);  CBrush\* brush = nullptr;  if (!brush)  brush = CBrush::FromHandle(CreateSolidBrush(RGB(255, 255, 255)));  dc.FillRect(&rc, brush);  CBitmap bitmap;  bitmap.LoadBitmapW(IDB\_BITMAP2);  CBrush bitBrush(&bitmap);  dc.FillRect(&rc, &bitBrush);  }  void CMfcDlgDlg::OnBnClickedButton1()  {  // TODO: 在此添加控件通知处理程序代码  CClientDC dc(&m\_picCtrl);  CRect rc;  m\_picCtrl.GetClientRect(&rc);  CBrush\* brush = nullptr;  if(!brush)  brush = CBrush::FromHandle(CreateSolidBrush(RGB(255, 255, 255)));  dc.FillRect(&rc, brush);  CBitmap bitmap;  bitmap.LoadBitmapW(IDB\_BITMAP1);  CBrush bitBrush(&bitmap);  dc.FillRect(&rc, &bitBrush);    } |

# 下面是vs2008的版本

|  |
| --- |
|  |

## Lesson8-MFCdlgDlg.cpp的完整代码如下

|  |
| --- |
| // Lesson8-MFCdlgDlg.cpp : 实现文件  //  #include "stdafx.h"  #include "Lesson8-MFCdlg.h"  #include "Lesson8-MFCdlgDlg.h"  #ifdef \_DEBUG  #define new DEBUG\_NEW  #endif  // 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()  // CLesson8MFCdlgDlg 对话框  CLesson8MFCdlgDlg::CLesson8MFCdlgDlg(CWnd\* pParent /\*=NULL\*/)  : CDialog(CLesson8MFCdlgDlg::IDD, pParent)  {  m\_hIcon = AfxGetApp()->LoadIcon(IDR\_MAINFRAME);  }  void CLesson8MFCdlgDlg::DoDataExchange(CDataExchange\* pDX)  {  CDialog::DoDataExchange(pDX);  DDX\_Control(pDX, IDC\_PIC\_STATIC, m\_picCtrl);  }  BEGIN\_MESSAGE\_MAP(CLesson8MFCdlgDlg, CDialog)  ON\_WM\_SYSCOMMAND()  ON\_WM\_PAINT()  ON\_WM\_QUERYDRAGICON()  //}}AFX\_MSG\_MAP  ON\_BN\_CLICKED(IDC\_BUTTON1, &CLesson8MFCdlgDlg::OnBnClickedButton1)  ON\_BN\_CLICKED(IDC\_BUTTON2, &CLesson8MFCdlgDlg::OnBnClickedButton2)  END\_MESSAGE\_MAP()  // CLesson8MFCdlgDlg 消息处理程序  BOOL CLesson8MFCdlgDlg::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 CLesson8MFCdlgDlg::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 CLesson8MFCdlgDlg::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  {  CClientDC dc(&m\_picCtrl);  CRect rc;  m\_picCtrl.GetClientRect(&rc);  CBrush\* brush = CBrush::FromHandle(CreateSolidBrush(RGB(255, 255, 255)));  dc.FillRect(&rc, brush);  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 CLesson8MFCdlgDlg::OnQueryDragIcon()  {  return static\_cast<HCURSOR>(m\_hIcon);  }  void CLesson8MFCdlgDlg::OnBnClickedButton1()  {  // TODO: 在此添加控件通知处理程序代码  CClientDC dc(&m\_picCtrl);  CRect rc;  m\_picCtrl.GetClientRect(&rc);  CBrush\* brush = NULL;  if(!brush)  brush = CBrush::FromHandle(CreateSolidBrush(RGB(255, 255, 255)));  dc.FillRect(&rc, brush);  CBitmap bitmap;  bitmap.LoadBitmapW(MAKEINTRESOURCE(IDB\_BITMAP1));  CBrush bitBrush(&bitmap);  dc.FillRect(&rc, &bitBrush);  }  void CLesson8MFCdlgDlg::OnBnClickedButton2()  {  // TODO: 在此添加控件通知处理程序代码  CClientDC dc(&m\_picCtrl);  CRect rc;  m\_picCtrl.GetClientRect(&rc);  CBrush\* brush =NULL;  if(!brush)  brush = CBrush::FromHandle(CreateSolidBrush(RGB(255, 255, 255)));  dc.FillRect(&rc, brush);  CBitmap bitmap;  bitmap.LoadBitmapW(MAKEINTRESOURCE(IDB\_BITMAP2));  CBrush bitBrush(&bitmap);  dc.FillRect(&rc, &bitBrush);  } |

### 写法和vs2019有点不一样

### 效果

|  |
| --- |
|  |
|  |
|  |

# 小结：

## 1.如果需要在对话框初始化后就在一些绘图之类的操作，如填充背景，需要在对话框的OnPaint函数里面添加代码，如，我们想给图片框填充白色背景

|  |
| --- |
|  |