

# VC++ Windows程序设计

对话框及部分控件的使用

# 1.0 第一个Windows程序

- Ch9-1(hello.cpp)
- `#include <windows.h>`
- `int WINAPI WinMain (HINSTANCE hInstance, HINSTANCE hPrevInstance, PSTR szCmdLine, int iCmdShow)`
- `{`
  - `MessageBox (NULL, L"Hello, World!", L"HelloWindow", MB_OK);`
  - `return 0 ;`
- `}`

# 1.0.1 Windows程序的建立

- New->Project->Win Project->Empty Project
- `#include <windows.h>`
- 程序入口 WinMain
- MessageBox函数用于显示消息
- `MessageBox(HWND hWnd, LPCWSTR lpText, LPCWSTR lpCaption, UINT uType);`
- hWnd 窗口句柄 lpText显示的消息
- lpCaption 窗口标题 uType显示的按钮

## 1.0.2 对话框中的按钮

- 对话框中显示的按钮 MB\_开头
  - #define MB\_OK 0x00000000L
  - #define MB\_OKCANCEL 0x00000001L
  - #define MB\_ABORTRETRYIGNORE 0x00000002L
  - #define MB\_YESNOCANCEL 0x00000003L
  - #define MB\_YESNO 0x00000004L
  - #define MB\_RETRYCANCEL 0x00000005L

## 1.0.3 带消息传递的Windows程序

- 示例ch9-2(windows.cpp)

# 1.1 对话框的组成

- 在VC++中，对话框主要有对话框资源和对话框类组成：
- 对话框资源中描述的是对话框的大小、风格等特性，以及对话框中控件的类型和它们的位置。定义对话框资源最简单最有效的方法就是使用Visual C++中的资源编辑器。
- 对话框类是从Cdialog类中派生出来的子类，它提供了编程接口来管理对话框。

## 1.2 对话框的分类

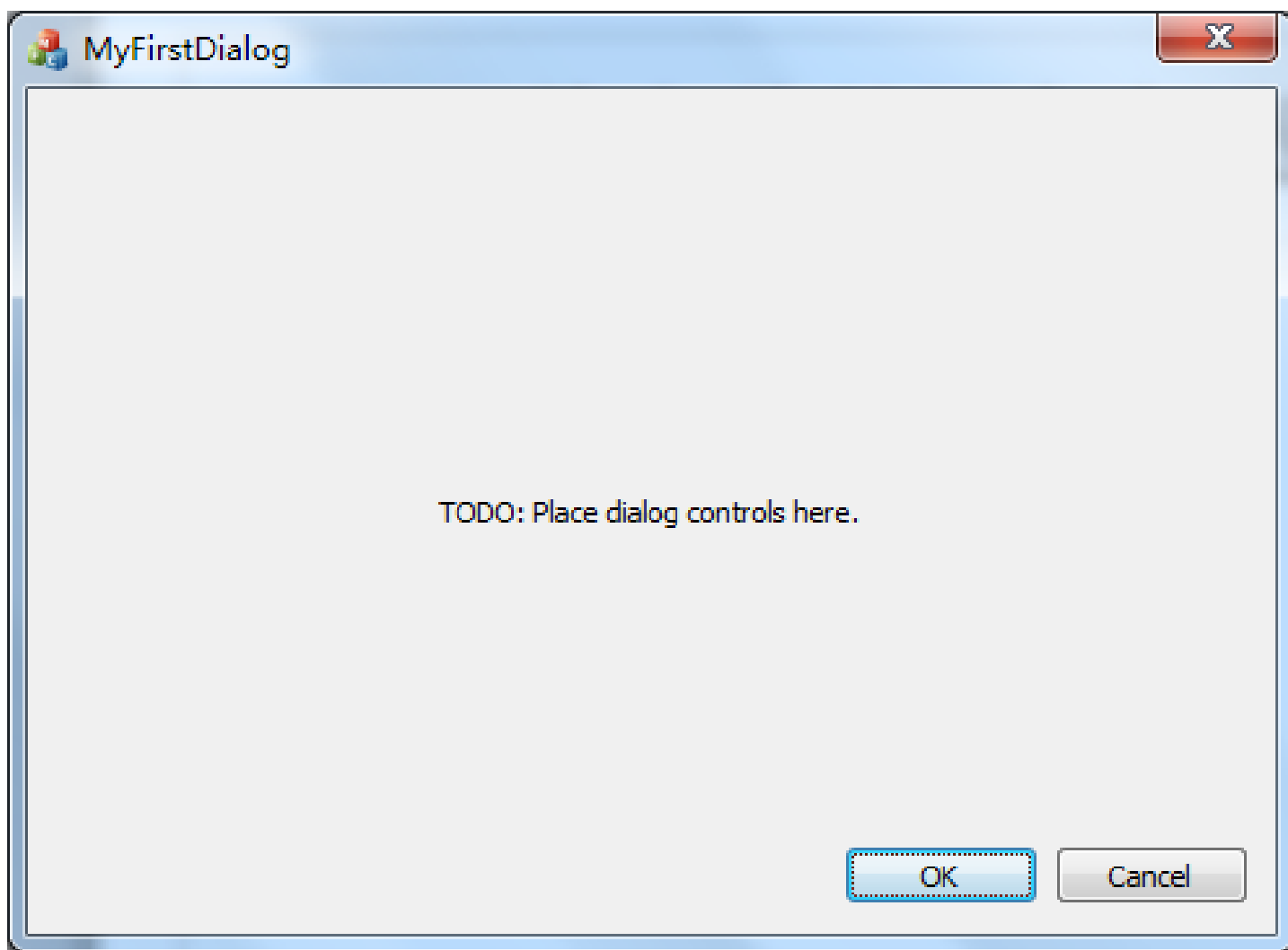
- 对话框有两种类型：模态对话框和非模态对话框
- 模态对话框是指对话框弹出后用户必须在对话框上进行操作，在退出对话框之前会阻止其他应用程序的运行。如Word中的“打开”对话框。
- 非模态对话框是指对话框在弹出后，将一直保留在屏幕上，不影响其他应用程序的运行，用户可以在对话框所在的应用程序中执行其他操作；当需要该对话框时，只需在对话框区域内点击一下即可激活。如Word中的“查找/替换”对话框

- 两种对话框在用资源编辑器设计和使用Class Wizard进行编程时的方法基本一致，仅在创建和退出对话框时的方式有所不同。
- 创建时，模态对话框有系统自动分配内存，因此退出时对话框会自动删除。非模态对话框是有用户指定内存，退出时需要用户自己来删除对话框对象。
- 退出时，两种对话框调用的终止函数不同，模态对话框通过调用Cdialog::EndDialog来终止，非模态对话框通过调用CWnd::DestroyWindow来终止。

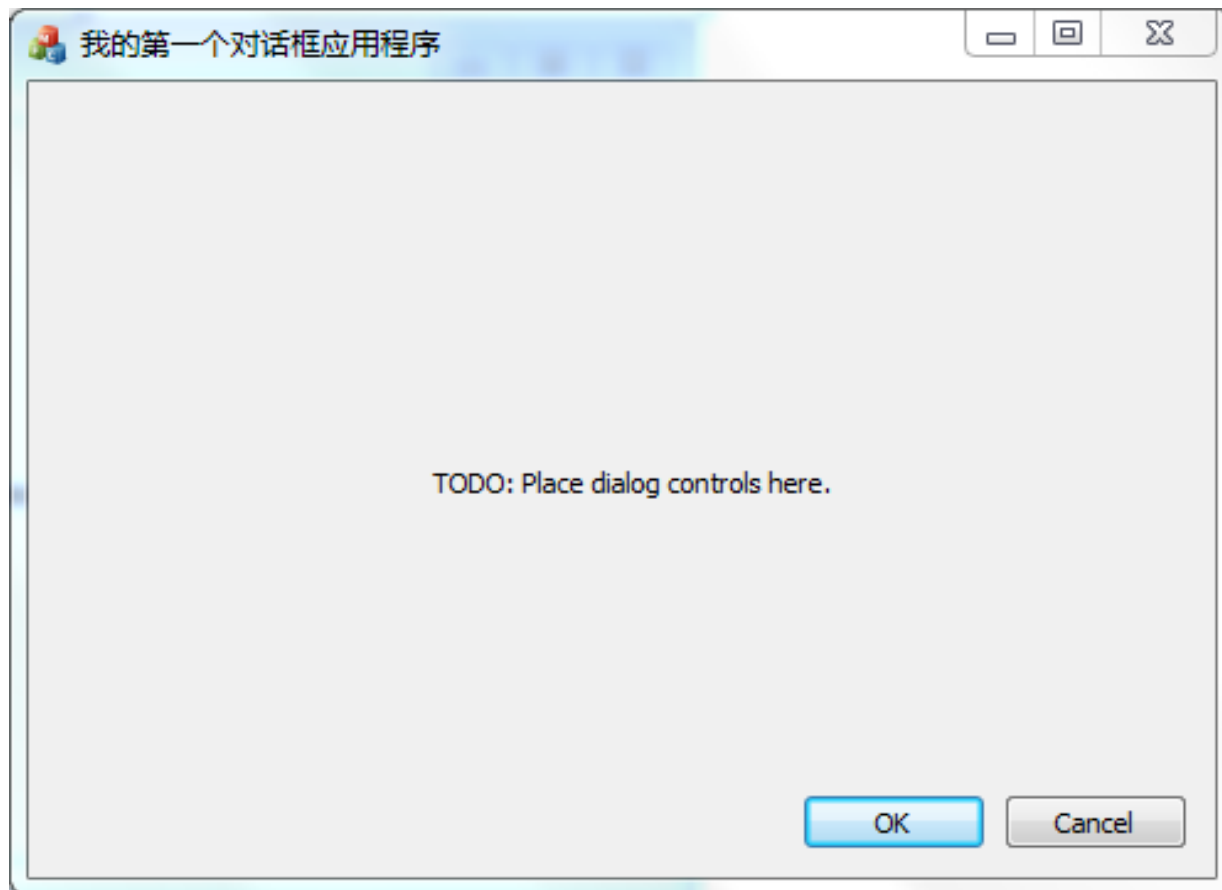


# 1.3 基于对话框的应用程序设计

- 基于对话框的应用程序结构适合于比较简单的应用程序。
- **示例演示（MyFirstDialog）**
- （1）生成基于对话框的应用程序
- 第1步：对话框中，选择生成Dialog Based（基于对话框）的应用程序即可。
- 第2步：用于设置应用程序的属性，包括是否使用版权对话框、是否需要生成上下文有关的帮助、是否使用3D风格的控件，以及是否需要ActiveX控件的支持等问题。
- 第3步：用于设置程序风格。
- 第4步：用于检查和修改拟为应用程序生成的类的有关参数。
- 即生成基于对话框的应用程序。



- （2）设置对话框界面的属性
- 编辑主对话框，按右键选择属性，则可设置对话框的属性。
- 常用的属性有：
- 基本属性（General）：字体、菜单、对话框所处位置等；
- 风格(Styles)：系统菜单、最大化、最小化、滚动条等；
- 更多风格（More Styles）、扩展风格、更多扩展风格。



- (3) 对话框的初始化
- 对话框的初始化工作一般在构造函数和 `CDialog::OnInitDialog()` 函数中完成。
- 在对话框创建时，会收到 `WM_INITDIALOG` 消息，对话框对该消息的处理函数是 `OnInitDialog()`。
- 如果必要，用 `ClassWizard` 重载此函数

- (4) 对话框数据交换和数据检验机制
- 控件与用户的数据交流和检验机制通过 `CDialog::DoDataExchange()` 函数来完成,
- 其原型为: `virtual void DoDataExchange ( CDataExchange * pDX );`
- 成员变量的值 与 控件显示的值之间的传递方向由下面两个函数决定:
- `UpdateData(TRUE);` //从对话框到变量
- `UpdateData(FALSE);` //从变量到对话框

# 1.4 常用控件

- 1.4.1 菜单
- 在基于对话框结构的应用程序上添加菜单
- （1）新建菜单资源
- 在资源编辑区中添加一新的菜单，然后编辑此菜单，添加相应的菜单项。
- （2）添加到对话框上
- 编辑对话框资源，选择对话框属性，在弹出的属性对话框中，在菜单选项中选中需要添加的菜单ID，即可。

## 1.4.1 模态对话框

- (1) 编辑对话框资源。
- (2) 为对话框建立一个新类，然后创建对话框类的对象，才能使用这个资源。
- (3) 打开模态对话框。在某类中，要使用模态对话框，则：
  - a.加入头文件：该模态对话框类的头文件；
  - b.定义模态对话框类对象： 类名 对象名；
  - c.显示该模态对话框： 对象.DoModal( )；

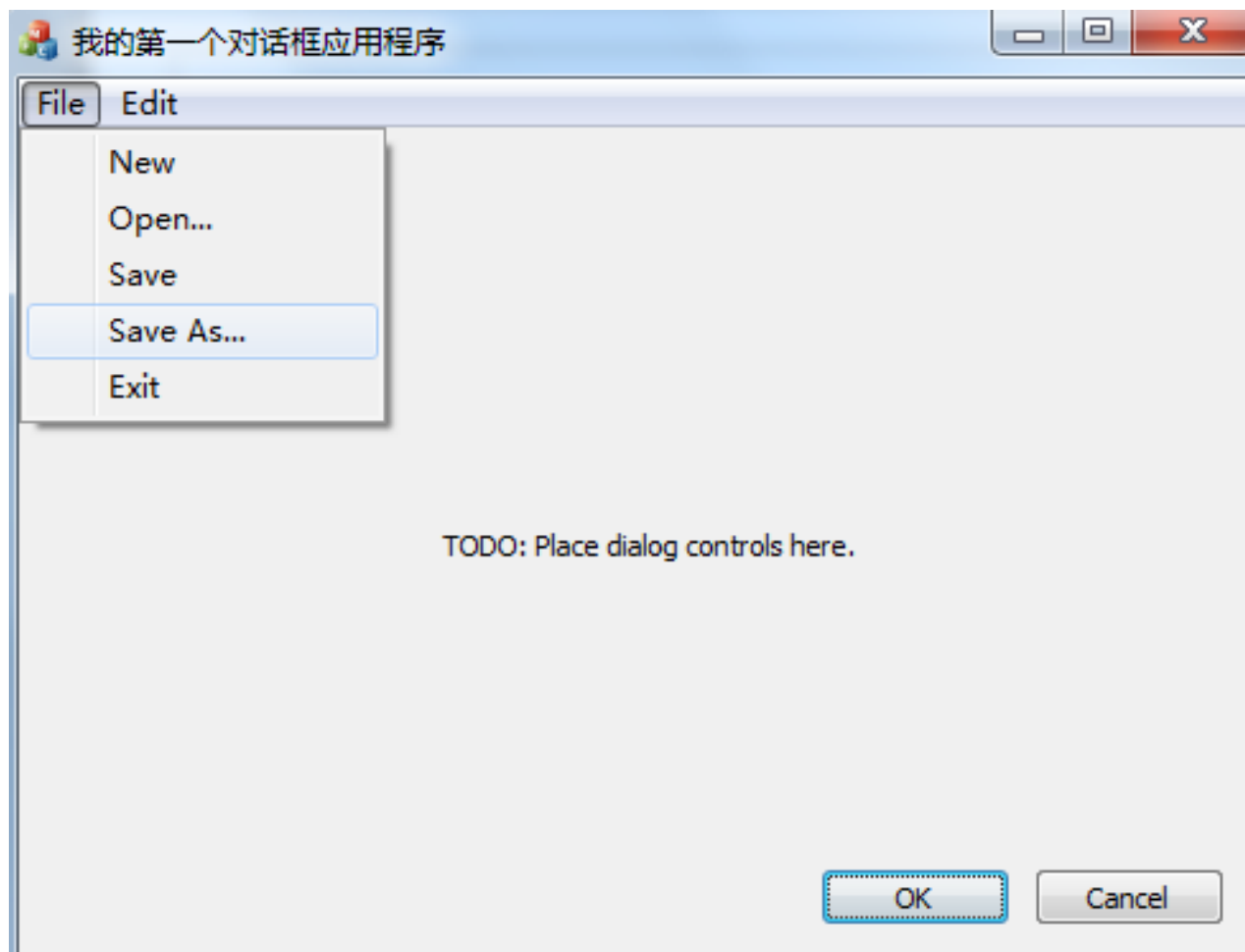


## 1.4.2 非模态对话框

- (1) (2) 同模态对话框
- (3) 在要用到的类中定义一个成员变量，即一个指向非模态对话框的指针。
- **注意：**
  - (a) 非模态对话框的模板资源必须具有Visible风格；
  - (b) 非模态对话框对象是用new操作符动态创建的；
  - (c) 通过调用CDialog::Create()函数来启动非模态对话框；
  - (d) 必须重载并重新编写对话框的OnOK()和OnCancel()函数；
  - (e) 必须在对话框关闭后，用delete操作符删除之；
  - (f) 必须有一个标志表明非模态对话框是否打开。

- CDialogModelless \*myModellessDlg = new CDialogModelless();
- if(myModellessDlg != NULL)
- {
- BOOL ret = myModellessDlg->Create(IDD\_DIALOG1,this);
- if(!ret) //Create failed.
- MessageBox(L"Error creating Dialog");
- myModellessDlg->ShowWindow(SW\_SHOW);
- }
- else
- MessageBox(L"Error Creating Dialog Object");

## 1.4.3 菜单



## 1.4.4 Button按钮

- Button
  - Toolbox拖拽到对话框
  - 设置属性（文字、ID等）
  - 双击Button设置鼠标单击事件
  - 右键->Class Wizard...->事件处理

## 1.4.5 文本相关控件

- Static Text
  - 提示
- Edit Control
  - 用户名、密码、文件名等主要信息
- Rich Edit 2.0 Control
  - 多行，设置字体格式颜色等。
  - 需要在对话框构造函数中添加如下语句：
    - `if (LoadLibraryA("RICHED20.DLL") == NULL)`
    - `{ AfxMessageBox(_T("Fail to load \\"riched20.dll\\"."),MB_OK | MB_ICONERROR);`
    - `PostMessage(WM_QUIT,0,0);}`

## 1.4.5 文本相关控件

- 控件关联变量
  - 右键 Add Variable...
    - Value/Control
- 根据控件ID获得指向该控件的指针
  - (CType\*)GetDlgItem(ControlID)
- UpdateData
  - TRUE 控件内容到值
  - FALSE 值到控件内容

## 1.4.5 公用对话框

- (1) CColorDialog (颜色选择对话框类)
- (2) CFontDialog (字体选择对话框类)

这两个可以直接这样打开使用：  
**CFontDialog f;f.DoModal();**

- (3) CPrintDialog (打印和打印设置对话框类)
- (4) CFindReplaceDialog (文本查找和替换对话框类)
- (5) CFileDialog (文件选择对话框类)

## 1.4.6 Rich Edit 2.0 Control

- 使用CFontDialog
  - UpdateData(TRUE);
  - CHARFORMAT2W cf;
  - CFontDialog f;
  - if(f.DoModal() == IDOK)
  - { f.GetCharFormat(cf);}
  - m\_RichEdit.SetSelectionCharFormat(cf);
  - UpdateData(FALSE);



# 1.4.7 CFileDialog

- 函数原型:

```
CFileDialog(BOOL bOpenFileDialog,LPCTSTR lpszDefExt=NULL,LPCTSTR  
lpszFileName=NULL,DWORD dwFlags = OFN_HIDEREADONLY | OFN_OVERWRITEPR  
OMPT,LPCTSTR lpszFilter=NULL,CWnd * pParentWnd  
=NULL);
```

参数说明:

bOpenFileDialog:为TRUE或FALSE。TRUE为打开文件；FALSE为保存文件。

lpszDefExt :为缺省的扩展名。

lpszFileName :为显示在文件名组合框的编辑框的文件名，一般可选NULL

dwFlags :为对话框风格，一般为OFN\_HIDEREADONLY | OFN\_OVERWRITEPROMPT,即隐藏只读选项和覆盖已有文件前提示。

LpszFilter :为下拉列表框中显示文件类型。

pParentWnd :一般可选NULL。

例如：在文件类型列表框的编辑框要显示“可执行文件(\*.exe)”，而在它的下拉列表框中列出“视频文件(\*.avi)、All Files(\*.\*)”内容，则变量定义如下：CFileDialog

```
m_MyOpenDialog(TRUE,"exe",NULL,OFN_HIDEREADONLY | OFN_OVERWRITEPROMPT,"可执  
行文件(*.exe) | *.exe|视频文件(*.bat) | *.bat |All Files (*.*) | *.* ||",NULL);
```

## 1.4.7 CFileDialog

- `//CFileDialog fileDlg(TRUE, L"*.*", NULL, OFN_HIDEREADONLY|OFN_OVERWRITEPROMPT, L"文本文件(*.txt)|(*.txt)|所有文件(*.*)|(*.*)||");`
- `CFileDialog fileDlg(TRUE);`
- `if(fileDlg.DoModal() == IDOK)`
- `{`
  - `m_FileName = fileDlg.GetFileName();`
  - `UpdateData(FALSE);`
- `}`

## 1.4.8 Radio Button

- Radio Button 单选按钮 多选一
  - 从ToolBox中拖拽到对话框
  - 设置属性
  - 第一个设置Group为True
  - 在第一个Radio Button上添加关联变量iRadioBn
  - 在OnInitDialog中设置默认选定按钮
    - CheckRadioButton(IDC\_RADIO1, IDC\_RADIO3, IDC\_RADIO2);
  - iRadioBn = GetCheckedRadioButton(IDC\_RADIO1, IDC\_RADIO3);

## 1.4.9 Check Box

- Check Box 复选框 多选多
  - 从ToolBox中拖拽到对话框
  - 设置属性
  - 第一个设置Group为True
  - 添加关联变量 Control类型
  - GetCheck SetCheck函数
  - ((CButton\*)GetDlgItem(IDC\_CHECK1))->SetCheck(1)/GetCheck();

# 1.4.10 Combo Box

- Combo Box 下拉列表
  - 常用于籍贯(省市自治区)、民族、学历等
  - 一般在OnInitDialog中初始化
  - 添加关联变量
    - `int iCount = ((CComboBox*)GetDlgItem(IDC_COMBO_RIGIN))->GetCount();`
    - `if(iCount < 1)`
    - `{`
    - `((CComboBox*)GetDlgItem(IDC_COMBO_RIGIN))->ResetContent();`
    - `((CComboBox*)GetDlgItem(IDC_COMBO_RIGIN))->AddString(L"Anhui");`
    - `((CComboBox*)GetDlgItem(IDC_COMBO_RIGIN))->AddString(L"Beijing");`
    - `((CComboBox*)GetDlgItem(IDC_COMBO_RIGIN))->AddString(L"Chongqin");`
    - `}`
    - `((CComboBox*)GetDlgItem(IDC_COMBO_RIGIN))->SetCurSel(0);`

## 1.4.11 其他控件

- Tree Control
- List Control
- Tab Control
- Data Time Picker
- Month Calendar Control
- Group Box
- .....

# 1.5 MFC类库

- 类
  - 类的概括、类成员的种类、以及该类成员函数、重载操作符或数据成员的基本用途
- 全局函数
- 全局变量
- 宏