

Visual C++ 编程经验三则

徐大宏 (湖南师范大学计算机部 410081)

摘要: 本文着重讨论了 Visual C++ 深入使用对话框编程中的两点技巧, 以及在文档视类框架应用程序中关于命令行参数的使用方法。

关键词: 对话框 命令行 可变大小

一、可变大小的对话框实现

不可否认, 对话框在应用程序中有着不可忽视的作用, 无论是应用程序与用户间的相互对话, 还是应用程序本身的信息展示, 都有对话框的身影出现。更有趣的是, 我们经常看到一些大小可变的对话框, 如: 当我们运行的应用程序出现错误时, 有时系统会弹出一个显示相关信息的对话框, 并提供三个按钮: 关闭、调试、详细资料, 我们单击详细资料按钮后, 此时在该对话框的下部下拉出它的另一部分。这是可变大小对话框的精彩运用之一。笔者在反复尝试的基础上, 悟出这一技巧的具体实现方法。

(1) 在资源编辑器中建立一对话框, 可将此对话框分为两部分: 一为较小状态部分, 二为较大状态时应增加的显示部分。将其中任意一部分用一个 Frame 类型的 Picture 控件(资源 ID 为 IDC_FRAME)包围起来, 目的是可以在程序中, 通过此控件动态地获取对话框应显示部分的大小。本文示例中用此控件围住将显示的较小部分, 并定义一控件成员变量 m_frame。

(2) 在(1)中对话框对应的对话框派生类的 OnInitDialog() 函数中加入如下代码段

```
// 用以保存整个对话框所有部分的尺寸参数
CRect rcDialog;
// 用以保存对话框较小部分的尺寸参数
CRect rcFrame;
// 取整个对话框在显示屏幕中的尺寸参数, 相对于
// 屏幕坐标系的左上角
GetWindowRect(rcDialog);
// 取对话框较小部分在显示屏幕中的尺寸参数, 相
// 对于屏幕坐标系的左上角
m_frame.GetWindowRect(rcFrame);
// 以对话框较小部分的大小创建一个矩形状区域
m_rgn.CreateRectRgn(0, 0,
    rcDialog.Width(),
    rcLine.bottom - rcDialog.top + 2);
```

// 设置对话框窗口的形状区域, 此处为较小状态时的矩形

```
SetWindowRgn((HRGN)m_rgn, TRUE);
```

此段程序代码放在 OnInitDialog() 函数, 使对话框在出现时以较小状态显示。其中的 GetWindowRect() 函数不能用 GetClientRect() 替代。

(3) 在对话框中设置一按钮, 类似于上面所述的详细资料按钮, 响应此按钮的单击事件(BN_CLICKED), 将对话框显示成较大状态窗口。使用如下代码:

```
CRect rcDialog;
GetWindowRect(rcDialog);
// 以整个对话框的大小创建一矩形区域
m_rgn.CreateRectRgn(0, 0,
    rcDialog.Width(),
    rcDialog.Height());
SetWindowRgn((HRGN)m_rgn, TRUE);
```

运行本方法的实例程序后, 对话框显示的初始状态为较小状态, 当单击按钮后, 对话框则增加显示另一部分, 实现真正可变大小的对话框。

如果将本例中 CRgn 类的成员函数 CreateRectRgn() 变换成它的另一成员函数 CreateEllipticRgn(), 那么, 我们就可以创建出一个圆形或椭圆形的对话框窗口。

二、基于对话框应用程序启动封面的制作

非对话框应用程序启动封面的制作较为简单。通过使用 Microsoft Developer Studio 的 "Project" 中的 "Add To Project" 菜单项, 插入 Developer Studio Componets and Controls Gallery 中的一个 Splash screen 控件对象, 然后编辑资源中的 ID 为 IDB_SPLASH 的位图资源即可, 无须任何编程操作。且系统会自动生成 CSplash 类加入至应用程序项目中。但在基于对话框应用程序中使用此种方法时, 系统会提示以下出错信息, 且操作不成功。

Cannot find the main frame class.

Please confirm that this project contains a class named

CMainFrame.

但创建基于对话框应用程序的启动封面可采用如下方法:

(1) 任意生成一个非对话框应用程序,并插入一个 Splash screen 对象;

(2) 将(1)中产生的 CSplash 类头文件(splash.h)及实现文件(splash.cpp),拷贝、添加至对话框应用程序的项目中;

(3) 在对话框应用程序中添加一位图资源,将 ID 定义为 IDB_SPLASH,并编辑位图;

(4) 在对话框应用程序的 CWinApp 派生类的 InitInstance()函数中加入如下代码即可。

```
CSplashWnd::EnableSplashScreen(TRUE);
```

```
CSplashWnd::ShowSplashScreen ( AfxGetMainWnd  
());
```

笔者经过反复调试运行,发觉实现启动封面的关键之处是这两个函数的调用。在普通应用程序中是由系统自动产生并添加至相应处进行调用。但对话框应用程序并不接受系统插入 Splash screen 对象这一操作,因此这两个函数的调用只能由手工加入。

三、在 MFC 文档及视类结构的程序中 使用命令行参数

虽然在 Windows 风格的系统中,应用程序已很少使用命令行参数了,但编程人员确实可以利用命令行参数,为自己的开发或者为开发的应用程序增色不少。

在使用 Developer Studio 的 Wizard 设计应用程序时,系统本身已自动地对命令行参数作了一些处理,且在应用程序 CWinApp 派生类的 InitInstance()函数加入了如下代码。

```
// 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;
```

其中 ProcessShellCommand()函数接受的参数类型为 CCommandLineInfo,这一对象的成员变量 m_nShellCommand 决定了 ProcessShellCommand()函数将执行何种动作。m_nShellCommand 是一枚举类型,可为以下值 FileNew, FileOpen, FilePrint, FilePrintTo, FileDDE, FileNothing, 这不同值分别代表不同实际动作,如创建新

文档,打开文档,打印文档等。一般情况下,当我们在命令行使用了参数之后,m_nShellCommand 的默认值将是 FileOpen,这意味着系统将把命令行参数看作是一个可供打开的文档文件名,这时往往会产生一些错误,而使应用程序无法继续运行,更谈不上我们自己亲自处理命令行参数了。

在应用程序中自己要接受并处理命令行参数之前,应该先使系统调用 ProcessShellCommand()函数时,并不将命令行参数看作是文档文件名而执行打开文档操作。方法是在 CWinApp 派生类的 InitInstance()函数中加入如下粗体部分代码:

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

```
CCommandLineInfo cmdInfo;
```

```
ParseCommandLine(cmdInfo);
```

```
if ( m_lpCmdLine[0] != '\0' )
```

```
cmdInfo.m_nShellCommand = CCommandLineIn-  
fo::FileNew;
```

```
// Dispatch commands specified on the command line
```

```
if (! ProcessShellCommand(cmdInfo))
```

```
return FALSE;
```

代码含义为:当使用了命令行的参数时,我们人为地将 ProcessShellCommand()变为执行创建新文档这一动作,从而避免错误的发生,让用户在后续代码中处理命令行参数。

然后我们可以在 CWinApp 派生类的任何位置使用其成员变量 m_lpCmdLine 获取命令行的真正参数,此参数是一个空结束的字符串,它并不包含应用程序文件名。如在其他派生类中,则可如此获取命令行参数:

```
AfxGetApp() ->m_lpCmdLine
```

四、结语

本文所描述的方法与技巧,笔者均在 Visual C++ 5.0 环境,Windows 9x / NT 系统中调试运行。笔者在开发 Windows 9x 系统下的考试软件时使用了这些技巧,确实为程序增色不少。

参考文献

- [1] Kruglinski, D.J 著,潘爱民,王国印译《Visual C++ 技术内幕》清华大学出版社,1999
- [2] Bennett, D. 著,徐军等译《Visual C++ 5 开发人员指南》机械工业出版社,1998

(来稿时间:1999 年 7 月)