

# CDialogBar 的使用技巧

吕少鹏 朱绍文 张德喜 (华中师范大学电子与计算机研究所 430070)

**摘要:**本文针对在 Visual C++ 编程环境中使用 CDialogBar 时遇到的几个困难,给出了相应的解决技巧。

**关键词:** Visual C++ CDialogBar

## 一、引言

在 Visual C++ 编程环境中, CDialogBar 是 MFC 类库提供的一个无模式对话框控制条,它可以象工具条一样操作方便,又可以象对话框一样,加上各种标准的 Windows 控件。使用好 CDialogBar,可以使程序界面更加美观,操作更方便。

一般情况下,简单地使用 CDialogBar 时可以按如下步骤进行。首先,用对话框编辑器定义一个 WS\_CHILD 风格而无其他风格的对话框模板,此模板不必具备 WS\_VISIBLE 网络。在应用代码中先调用构造函数构造 CDialogBar 对象,然后调用 CDialogBar 的 Create 函数创建对话框条,并将此赋着于 CDialogBar 对象。但有时 CDialogBar 上安装的控件较多较复杂时,则很需要重新设计 CDialogBar 的派生类。另外,当在控制条上装上控制按钮时,按钮总是呈灰色,处于禁止状态,无法响应鼠标的按下事件。最后,许多程序员总是希望能随意地控制 C dialog Bar 的显示和隐藏。以上几种情况往往让一般的程序员大伤脑筋而不得其解。其实上述问题都可以用简单的方法很好地加以解决。作者通过摸索掌握了其中的使用技巧,介绍如下:

### 1. CDialogBar 的加入及其派生类的设计

选择 App Studio 菜单 Insert 下的 Component, 在 Component Gallery 对话框中选择 DialogBar。按下 Insert 按钮后,系统将在主框架 CMainFrame 类中自动加上插入 DialogBar 的程序代码。查看头文件,有 CDialogBar mwndMyDialogBar,知道此时程序中是直接使用了 CDialogBar。打开 CDialogBar 的对话框模板资源,可以在上面加入各种所需的 Windows 控件。打开 Class Wizard,当弹出 Adding a class 对话框时,选择 Create a new class 项,在类名输入对话框中输入 CDialogCtrl 作为类名,按下 OK 键后系统将生成 Cdialog 的派生类 CDialogCtrl。通过修改 CDialogCtrl 类的内容,可以得到 CDialogBar 的派生

类。具体修改如下:

(1)打开 CDialogCtrl 的头文件,将 class CDialogCtrl: public CDialog 改为 class CDialogCtrl: public CDialogBar。同时将构造函数的声明 CDialogCtrl(Wnd \* pParent = NULL)改为 CDialogCtrl();

(2)打开 CDialogCtrl 的实现文件,将构造函数的定义 CDialogCtrl::CDialogCtrl(CWnd \* pParent): CDialog(CDialogCtrl::IID, pParent) {} 改为 CDialogCtrl(): CDialogBar() {}。同时必须将该文件中的所有 CDialog 用 CDialogBar 替代。

(3)打开 CMainFrame 类的头文件,将 CDialogBar mwndMyDialogBar 中的 CDialogBar 改为 CDialogCtrl。

(4)利用 Class Wizard 往 CDialogCtrl 中加入各个控件的事件函数,则此时的 CDialogCtrl 便成为 CDialogBar 的派生类,我们可以在这个类里面完成响应控件各种事件的任务。

### 2. 激活按钮的方法

运行上述程序,如果对话框条中装有按钮,将发现按钮始终呈灰色,处于禁止状态,不能响应鼠标的动作。解决这个问题很简单,具体步骤如下:

(1)打开 CMainFrame 的实现文件,在 BEGIN-MESSAGE-MAP 中加入事件宏 ON-BN-CLICKED(IDC\_BUTTON, OnButton),其中 IDC\_BUTTON 为该 Button 的 ID 号。

(2)在 CMainFrame 类中定义 Void OnButton() 函数。

经过如上过程编译运行程序,将发现对话框条中的相应按钮已经激活。按下按钮,将调用 CDialogBar 中的 OnButton() 函数(而不是 CMainFrame 中的 OnButton())。

### 3. CDialogBar 的显示与隐藏

就象 Word 中的工具栏一样,编程者往往希望程序中能随意地显示或隐藏控制条。下面的代码演示了通过菜单项的选择来对控制条的显示和隐藏进行操作。

(1)在程序菜单中加入 CDialogBar 菜单。

(2) 在 CMainFrame 类中为上述菜单项定义如下事件函数:

```
void CMainFrame::OnViewDialogbar()  
{  
    CWnd * pBar;  
    if (pBar = GetDescendantWindow ( IDD-MYDI-  
ALOGBAR))  
    {  
        pBar -> ShowWindow ( pBar -> GetStyle ()  
& WS-VISIBLE) == 0);  
        RecalcLayout();  
    }  
    void CMainFrame:: OnUpdateViewDialogbar ( CCm-  
dUI * pCmdUI)  
    {  
        CWnd * pBar;  
        pBar = GetDescendantWindow ( IDD-MYDIALOG-  
BAR);
```

```
if ( pBar = GetDesscendantWindow ( IDD-MYDI-  
ALOGBAR))  
{  
    int i = pBar -> GetStyle() & WS-VISIBLE;  
    pCmdUI -> SetCheck (( pBar -> GetStyle ()  
& WS-VISIBLE)! = );  
}
```

代码中的 IDD-MYDIALOGBAR 是控制条的 ID 号。上述程序通过菜单项的选择来操作控制条的显示和隐藏。读者完全可以使用其中的代码, 在程序中对控制条的显示和隐藏进行任意控制。

### 参考文献

- [1] Microsoft Corporation 著, 张军、赵先瑞译, Microsoft Visual C++ for Win32 大全(三), 清华大学出版社, 1996, 2。

(来稿时间: 1998 年 4 月)