基于 MFC 对话框程序设计中 UPDATE_COMMAND_UI 机制实现方法探讨

Researches on the Methods of Realization UPDATE_COMMAND_UI in Dialog Program's Design Based on MFC

刘东华　惠荣勤　李秋娜　(装备指挥技术学院 航天测控工程研究中心 北京 101416)

摘要:本文简要分析了基于 MFC(Microsoft Foundation Class)程序中 UPDATE_COMMAND_UI 机制的实现原理以及对话框程序中不支持此机制的原因,详细介绍了在基于 MFC 的对话框程序设计中,针对三种不同的用户界面(UI)对象(包括控件、工具栏和菜单条),实现此机制的三种不同的方法。

关键词:MFC UPDATE_COMMAND_UI 对话框

1 引言

随着 Windows 程序的广泛应用,程序界面的设计成为了不可忽视的重要部分。使用 MFC 编写 Windows 程序中,所有用户界面(UI)对象状态的维护可以依赖 UPDATE_COMMAND_UI 消息,这个消息和 WM_COMMAND 有一样的传递路线,只需要在对应类中放置其处理函数,并在函数中做出某些判断,便可以决定 UI 对象的显示状态(如:激活、失效、选择)。在闲置处理过程中,MFC 把 WM_IDLEUPDATECMDUI 消息发送给主帧窗口和其直接子窗口——工具栏、状态栏、菜单条以及其视图。随后它们依次用 ON_UPDATE_COMMAND_UI 宏更新自身显示状态。然而,由于使用 MFC 自带的 ClassWizard 生成的对话框窗口不是主帧窗口的子窗口,所以该窗口接收不到 WM_IDLEUPDATECMDUI 消息,也就不支持 UPDATE_COMMAND_UI 机制。本文将针对不同的用户界面(UI)对象,介绍在对话框中实现控件、工具栏和菜单条的 UPDATE_COMMAND_UI 机制的三种实现方法。

2 控件对象实现 UPDATE_COMMAND_UI 机制

为了实现对话框中控件对象的 UPDATE_COMMAND_UI 机制,可以利用 WM_KICKDLE 消息。当对话框处于空闲进程时,这个消息自动发送给 MFC 对话框,对话框将调用 UpdateDialogControls( )函数,在这个函数中依次向对话框中的所有的控件发送 ON_UPDATE_COMMAND_UI 消息。UpdateDialogControls( )函数的原型是:

```
void CWnd::UpdateDialogControls(CCmdTarget* pTarget, BOOL bDisableIfNoHandler);
```

第一个参数 pTarget 用来标识要处理 ON_UPDATE_COMMAND_UI 消息的窗口指针。这里,把对话框的 this 指针作为 pTarget 变量。

第二个参数 bDisableIfNoHandler 控制 UpdateDialogControls( )的行为。如果该变量为 TRUE,则任何没有明确 UPDATE_COMMAND_UI 句柄的控件都会失效;如果为 FALSE,对那些没有明确 UPDATE_COMMAND_UI 句柄的控件根本不采取任何操作,通常把 FALSE 作为变量 bDisableIfNoHandler 的输入值。具体实现步骤如下:

第一步:由于 MFC 是在闲置处理过程中发送 WM_KICKIDLE 消息给主帧窗口和其子窗口,所以首先在对话框头文件(*.h)中要包含 <afxpriv.h> 文件,在对话框实现源文件(*.cpp)里定义 WM_KICKIDLE 私有的 MFC 消息。

第二步:**在对话框类头文件**(*.h)**中加入** OnKickIdle()**函数原型**。

```
class CDemoDlg : public CDialog
{
//…
  afx_msg void OnKickIdle();
//…
}
```

第三步:**在对话框消息映射表中,添加一个** ON_MESSAGE_VOID()**的宏,将** WM_KICKIDLE **消息映射为** OnKickIdle()。

```
BEGIN_MESSAGE_MAP(CDemoDlg, CDialog)
  //{{AFX_MSG_MAP(CDemoDlg)
  //…
  ON_MESSAGE_VOID(WM_KICKIDLE, OnKickIdle)
  //…
  //}}AFX_MSG_MAP
END_MESSAGE_MAP()
```

3　工具栏实现 UPDATE_COMMAND_UI 机制

第四步:**在** OnKickIdle **函数中添加代码,控制对话框中控件的显示状态。**

```
void CDemoDlg::OnKickIdle()
{
  UpdateDialogControls(this, FALSE);
}
```

此时,对话框就具备了 UPDATE_COMMAND_UI **机制。下面,将通过手工加入** UPDATE_COMMAND_UI **消息句柄,实现对话框中控件状态的改变。**

第五步:**在对话框类的头文件**(.h)**中加入** UPDATE_COMMAND_UI **句柄函数**。

```
afx_msg void OnUpdateChecked(CCmdUI* pCmdUI);
```

第六步:**在对话框的消息映射表中加入** ON_UPDATE_COMMAND_UI()**的宏**。

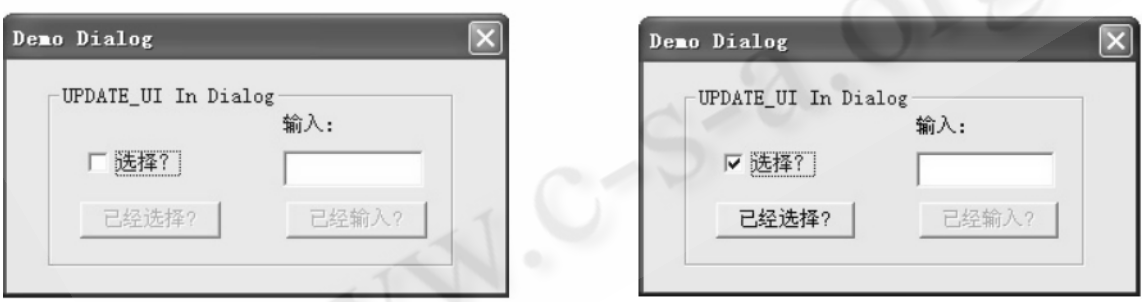
```
ON_UPDATE_COMMAND_UI(IDC_CHECKED, OnUpdateChecked)
```

第七步:**在对话框的实现文件**(*.cpp)**中添加** UPDATE_COMMAND_UI **宏的消息处理函数**。

```
void CDemoDlg::OnUpdateChecked(CCmdUI* pCmdUI)
{
  pCmdUI->Enable(m_chkCheck.GetCheck() == 1);
}
```

以上方法,实现了在一个对话框中,通过一个 Check Box **控件的状态,控制一个** CButton **对象的显示状态。当** Check Box **控件被选中时,**ID **号为** IDC_CHECKED **的按钮将变为可用**(enable)**状态;当** Check Box **控件未被选中时,此按钮将处于不可用**(disable)**状态。如图** 1 **所示。**

工具栏同普通的控件不同,它是通过 CControlBar **类默认的** WM_IDLEUPDATECMDUI **消息处理函数寻找主帧窗口,在该窗口调用** OnUpdateCmdUI()**函数,实现** UPDATE_COMMAND_UI **机制。然而,当工具栏是对话框的一部分的时候,**WM_IDLEUPDATECMDUI **消息将被送到主应用程序窗口,然后主应用程序窗口把** UPDATE_COMMAND_UI **消息发送到各种文档、视图和帧窗口。在前言部分已经提到对话框窗口不是主帧窗口的子窗口,所以对话框窗口接收不到** WM_IDLEUPDATE-



(a)　　(b)

图 1　对话框中的控件实现 UPDATE_COMMAND_UI **机制**

CMDUI 消息，也就不支持 UPDATE_COMMAND_UI 机制。

解决这个问题的方法是：首先按照第一部分所描述的方法，在闲置处理函数中发送 WM_IDLEUPDATECMDUI 消息到工具栏对象。然后自定义一个工具栏类（CDlgToolBar），在此类中通过改写 OnIdleUpdateCmdUI() 函数，使工具栏所在的对话框作为接收 WM_IDLEUPDATECMDUI 消息的主帧窗口。最后，在对话框中添加 UPDATE_COMMAND_UI 宏的消息处理函数，控制工具栏中按钮的显示状态。具体实现步骤如下：

第一步：在对话框的头文件（*.h）和执行文件（*.cpp）中添加 WM_KICKIDLE 消息处理函数。具体可参见本文第一部分中的步骤（1）–（3）。

第二步：在 OnKickIdle 函数中添加代码，使对话框中所有的子窗口接收到 WM_IDLEUPDATECMDUI 消息，其中包括工具栏。

```
void CDemoDlg::OnKickIdle()
{
    SendMessageToDescendants( WM_IDLEUPDATECMDUI );
}
```

第三步：创建 CDlgToolBar 类，这个类基于标准 CToolBar 类，在此类的头文件（*.h）中要包含 <afxpriv.h> 文件，能够处理 WM_IDLEUPDATECMDUI 消息。

第四步：在 CDlgToolBar 类头文件（*.h）中声明 WM_IDLEUPDATECMDUI 的消息处理函数。

```
afx_msg LRESULT OnIdleUpdateCmdUI(WPARAM wParam, LPARAM);
```

第五步：在 CDlgToolBar 类的消息映射表中添加一个 ON_MESSAGE() 的宏，将 WM_IDLEUPDATECMDUI 消息映射为处理 OnIdleUpdateCmdUI() 函数。

```
BEGIN_MESSAGE_MAP(CDlgToolBar, CToolBar)
//{{AFX_MSG_MAP(CDlgToolBar)
ON_MESSAGE(WM_IDLEUPDATECMDUI, OnIdleUpdateCmdUI)
//}}AFX_MSG_MAP
END_MESSAGE_MAP()
```

第六步：在 CDlgToolBar 类实现文件（*.cpp）中添加 OnIdleUpdateCmdUI() 函数。这个函数中将工具栏的父窗口（即对话框窗口）指针指定为 WM_IDLEUPDATECMDUI 消息的目标窗口。

```
LRESULT CDlgToolBar::OnIdleUpdateCmdUI(WPARAM wParam, LPARAM)
{
    CToolBar::OnIdleUpdateCmdUI(wParam, 0);
    if (IsWindowVisible())
    {
        CFrameWnd * pParent = (CFrameWnd *)GetParent();
        if (pParent)
            OnUpdateCmdUI(pParent, (BOOL)wParam);
    }
    return 0;
}
```

第七步：在对话框程序中添加基于 CDlgToolBar 类的工具栏对象。（实现方法较多，这里不再详细介绍）。

第八步：添加 UPDATE_COMMAND_UI 消息处理函数。具体可参见第一部分中的步骤（5）–（7）。

图 2 所示对话框中 Como Box 控件中“图案”的选择，将控制工具栏中对应该图像按钮的状态（弹起或按下）。

图 2　对话框中工具栏实现 UPDATE_COMMAND_UI 机制

4　菜单条实现 UPDATE_COMMAND_UI 机制

菜单条实现 UPDATE_COMMAND_UI 机制同上面两种情况稍有不同。在菜单显示的时候 WM_INITMENUPOPUP 消息被先发送以显示菜单项。MFC 程序中 CFrameWnd::OnInitMenuPopup 函数遍历菜单项并为每个菜单项调用更新命令处理函数（如果有的话）。菜单的外观被更新以反映它的状态（启用/禁用，选择/

取消选择)。菜单条的 UPDATE_COMMAND_UI 机制在基于对话框的应用程序中不能工作,因为对话框中没有 OnInitMenuPopup 处理函数,而使用 CWnd 的默认处理函数,该函数没有为菜单项调用更新命令处理函数。解决此问题的方法是:在对话框程序中手工添加 WM_INITMENUPOPUP 消息处理函数。具体步骤及主要代码如下:

第一步:在对话框类的头文件(*.h)中声明 ON_WM_INITMENUPOPUP 的消息处理函数。

afx_msg void OnInitMenuPopup(CMenu * pPopupMenu, UINT nIndex,BOOL bSysMenu);

第二步:在对话框类的消息映射表中将 ON_WM_INITMENUPOPUP 消息映射为处理 OnInitMenuPopup() 函数。

第三步:在对话框类中添加 OnInitMenuPopup 成员函数,此函数中的代码基本上同 CFrameWnd::OnInitMenuPopup() 相同,可直接复制。具体代码略。

第四步:添加 UPDATE_COMMAND_UI 消息处理函数。具体可参见第一部分中的步骤(5)-(7)。

图 3 所示,在对话框 Como Box 控件中"图案"的选择,将控制菜单条中对应该图像菜单按钮的状态(选择或取消选择)。

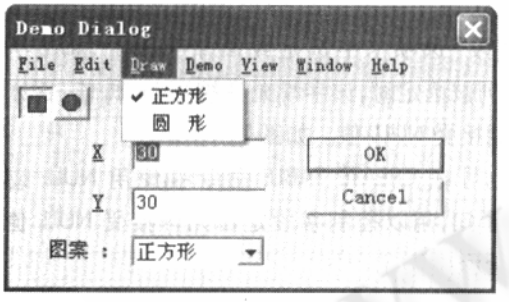


图 3　对话框中菜单条实现 UPDATE_COMMAND_UI 机制

对于右键弹出式菜单(Pop-Up Menu)在对话框中实现 UPDATE_COMMAND_UI 机制的方法同普通菜单的实现方法不同,具体可以参见文献[5]。

5　小结

针对 MFC 程序中 UPDATE_COMMAND_UI 机制的实现原理以及对话框程序的特点,文中给出了三种不同的用户界面对象在对话框程序中实现 UPDATE_COMMAND_UI 机制的三种不同方法。这三种方法适合于模态及非模态对话框程序。通过添加或修改函数的代码,比较方便地实现了对话框程序中常用界面对象的状态维护及更新。它使得 Windows 程序设计中,人机交互过程更加直观、合理,同时也使得程序界面设计更为简洁、标准。

参考文献

1 [美]Jeff Prosise. MFC Windows 程序设计(第二版)[M],清华大学出版社,2002.453-457。

2 向卫军,基于 MFC 编程的闲置处理控制[J],计算机系统应用,2002,(3):32-34。

3 [美] Eugene Kain. MFC 经典问答[M],北京 中国电力出版社,2001.257-258。

4 候俊杰,深入浅出 MFC(第二版)[M],武汉华中科技大学出版社,2001.443-446。

5 刘东华,Pop-Up Menu 在对话框中实现 UPDATE_COMMAND_UI 机制[J],电脑编程技巧与维护,2003,(1)51-53。