

在VC中用CMenuXP使应用程序拥有OfficeXP风格 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/270/2021\\_2022\\_\\_E5\\_9C\\_A8V](https://www.100test.com/kao_ti2020/270/2021_2022__E5_9C_A8V)

C\_E4\_B8\_AD\_E7\_94\_c67\_270964.htm 随着Windows XP和Office XP的推出，兼具华美和清爽气质的Office XP界面风格也逐渐取代了Windows经典界面风格而成为了当前应用程序的流行

“时装”，因而如何使自己的应用程序拥有Office XP风格的界面也成为了广大程序员感兴趣的问题。为此法国程序

员Jean-Michel LE FOL针对于Visual C编写了CMenuXP类，使用CMenuXP类就可以轻松使自己的应用程序拥有Office XP风格

的界面。下面我就以详细介绍一下使用CMenuXP的步骤。

首先新建一个名叫“SDIXP”的单文档的MFC应用程序。在我们能使用CMenuXP之前，必须进行一下几步操作：1.

将CMenuXP的源文件和头文件所在的文件夹（一般放在名叫“Tools”）放到SDIXP源文件的目录下。2.将Visual C切换到

“文件视图（FileView）”，加入CMenuXP的源文件和头文件。具体可以这样进行：在FileView的树状目录下，展开SDI

files，在Source Files目录上点击右键，创建新目录（“New Folder...”），不妨取名叫“Tools”，再在Tools目录上点击

右键，选择加入文件（“Add Files to Folder...”），将Tools文件夹下的所有源文件（\*.cpp）全部加进来。同理，在文件

视图的Header Files目录下新建一个叫做“Tool Headers”，再将Tools文件夹下的所有头文件（\*.h）全部加到Tool Headers

目录里。3.（1）在控制菜单的类的头文件（对于SDI程序通常是MainFrm.h）里加入所需要的CMenuXP类的头文件名：

```
#include "Tools/MenuXP.h" #include "Tools/ToolBarXP.h"
```

#include "Tools/StatusBarXP.h" #include

"Tools/ComboBoxXP.h" 再在MainFrm.h里加入下面的宏：  
DECLARE\_MENUXP() 上述语句可以加在CMainFrame类声明里的宏定义 "DECLARE\_MESSAGE\_MAP()" 的后面。（2）在控制菜单的类的源文件（MainFrm.cpp）里加入：  
IMPLEMENT\_MENUXP(类名, 基类名). 对于SDI工程就是IMPLEMENT\_MENUXP(CMainFrame, CFrameWnd). 上述语句加在CMainFrame开头的

"IMPLEMENT\_DYNCREATE(CMainFrame, CFrameWnd)" 后面。（3）接着继续在消息映射里如下加入：

```
BEGIN_MESSAGE_MAP(className, baseClass) // ...
```

```
ON_MENUXP_MESSAGES() //加入这一行
```

```
END_MESSAGE_MAP() （4）在继承于CWinApp的类（这里就是CSDIXPApp）中的InitInstance()函数里加入：
```

```
CMenuXP::InitializeHook(). 可以加在
```

"AfxEnableControlContainer()." 语句后面。然后再为CSDIXPApp类重载ExitInstance()函数，可以在“类视图（ClassView）”里点右键加入。再在ExitInstance()函数里加入：  
CMenuXP::UninitializeHook(). 以上步骤做完之后可以按F7编译一下，如果一切没有问题我们就可以开始使用CMenuXP来美化我们的菜单和工具栏了。首先，我们来将菜单和菜单项XP化。这很简单，只要在CMainFrame::OnCreate()函数里加入：  
CMenuXP::UpdateMenuBar(this).

CMenuXP::SetXPLookNFeel(this). 就OK了。现在你编译运行一下，看看你的程序菜单是不是改头换面了？如果你想为自己菜单项定制图标，我们可以用CMenuXP::SetMenuItemImage

(UNIT nID, HIMAGELIST hImgList, int nIndex)函数来实现。为此，我们新建一个菜单“水果”和菜单项“苹果”（ID设置为ID\_MENUITEM\_APPLE），打开ClassWizard为ID\_MENUITEM\_APPLE添加消息处理函数OnMenuItemApple()。然后再导入一个位图资源“APPLE.bmp”，其ID设置为IDB\_APPLE。接着我们为CMainFrame添加CImageList类型的成员变量m\_imgList，再在CMainFrame::OnCreate()函数的“CMenuXP::SetXPLookNFeel(this).”后面加入：  
m\_imgList.Create(IDB\_APPLE, 0, 0, RGB(192,192,192)).  
CMenuXP::SetMenuItemImage(ID\_MENUITEM\_APPLE, m\_imgList, 0). 其中m\_imgList.Create()函数中的第2,3个参数是位图文件的图像拷贝的左上角坐标，最后一个参数是图像的背景色。建议使用16×16的位图。SetMenuItemImage()函数的最后一个参数指的是m\_imgList的中元素的索引，这里因为m\_imgList只包含一个位图，故取零。编译后运行，看看“水果|苹果”菜单项前面是不是多了一个苹果的图标？下面我们将使工具栏和状态栏XP化，这个更加easy了。只要将CMainFrame类声明中的 CStatusBar m\_wndStatusBar. CToolBar m\_wndToolBar. 变成 CStatusBarXP m\_wndStatusBar. CToolBarXP m\_wndToolBar. 就搞定了！不信，你编译运行一下看看 CMenuXP还可以将按钮和下拉列表框变成Office XP风格的。为此我们新建一个对话框IDD\_APPLE。OK和Cancel按钮保持不动，我们只增添一个下拉列表框控件IDC\_COMBO1。然后按Ctrl W打开ClassWizard为该对话框新建一个CAppleDlg类。再在ClassWizard中的Member Variable里

为IDC\_COMBO1添加Control类型的变量m\_cmbx1。接着为CAppleDlg重载OnInitDialog()用以初始化控件：`BOOL CAppleDlg::OnInitDialog() { CDialog::OnInitDialog(). m_cmbx1.AddString(“红苹果”). m_cmbx1.AddString(“青苹果”). return TRUE. }`然后在“苹果”菜单项的消息处理函数中使苹果对话框弹出：`void CMainFrame::OnMenuItemApple() { // TODO: Add your command handler code here CAppleDlg dialog. dialog.DoModal(). }`如果这时编译运行，就会看到打开“水果|苹果”菜单项后弹出一个对话框，但上面的控件还不是Office XP风格的，下面我们就着手使其XP化，其实这同样也是很easy的！先在AppleDlg.h文件中包含“Tools/ComboBoxXP.h”和“Tools/ButtonXP.h”，再找到定义m\_cmbx1的位置，将CComboBox m\_cmbx1. 变成CComboBoxXP m\_cmbx1. 这样下拉列表框就有XP风格了，不信你编译运行试试。关于OK和Cancel按钮就稍微特殊一点，因为它们是直接用IDOK和IDCANCEL标识的默认按钮，但为了使它们XP化，我们还是得为这两个按钮添加控制变量。因此，在CAppleDlg类的声明中添加：`CButtonXP m_btnOK. CButtonXP m_btnCancel.`再在CAppleDlg::DoDataExchange()函数中添加数据交换：`DDX_Control(pDX, IDOK, m_btnOK). DDX_Control(pDX, IDCANCEL, m_btnCancel).`这样，OK和Cancel按钮也都XP化了。大家编译运行一下，看看效果怎么样。总之，使用Jean-Michel LE FOL的CMenuXP给用Visual C 开发的应用程序换上Office XP风格的界面还是很不错的，操作起来也比较简便。唯一不足的就是其他许多控件，如文本框等还是没有Office XP风格，这就期待着原作者以及我们

大家继续完善CMenuXP类。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)