

附录E MFC快速参考指南

E.1 创建窗口

使用MFC

```
CWnd wnd;
```

```
Wnd.CreateEx(ExStyle,ClassName,WindowName,Style,x,y,Width,Height,Parent,Menu,Param);
```

使用API

```
HWND hwnd=::CreateWindowEx(ExStyle,ClassName,WindowName,Style,x,y,Width,Height,Parent, Menu,Instance,Param);
```

E.2 创建窗口类

快速创建

```
LPCTSTR lpszClassName=AfxRegisterWndClass(UINT nClassStyle,HCURSOR hCursor=0,HBRUSH hbrBackground=0,HICON hIcon=0);
```

WNDCLASS详细结构

```
typedef struct _WNDCLASS {
    UINT style;           //style of the class (see below)
    WNDPROC lpfnWndProc; //function called by system when it has a message
    int cbClsExtra;      //extra bytes to add to the WNDCLASS structure when registering
    int cbWndExtra;      //extra bytes to add to the Window Objects created with this class
    HANDLE hInstance;    //instance that owns this class
    HICON hIcon;         //icon to be used when window displays an icon
    HCURSOR hCursor;     //cursor to use when mouse is over a window created with this class
    HBRUSH hbrBackground; //background color to use when erasing the background area
    LPCTSTR lpszMenuName; //menu name to be used when creating a menu for a window
    LPCTSTR lpszClassName; //the name that identifies this Window Class
} WNDCLASS;
BOOL AFXAPI AfxRegisterClass(WNDCLASS* lpWndClass);
```

E.3 Window类的类型

类 风格	描 述
CS_OWNDC	为该窗口类创建的每个窗口分配唯一的设备环境
CS_PARENTDC	从系统高速缓存中检索设备环境，然后设置该设备环境的剪裁区，以组合到父窗口中，以便子窗口能绘制在父窗口上
CS_CLASSDC	分配一个设备环境给所有由该窗口类创建的窗口使用
CS_SAVEBITS	由该类创建的任何窗口的视频存储区将被保存，以便窗口移动或关闭时，不需要重画基础窗口
CS_GLOBALCLASS	如果设置该风格，则该类是应用程序的全局类，否则它是一个应用程序局部类

(续)

类 风 格	描 述
CS_VREDRAW	如果设置了垂直风格,并且窗口的垂直大小发生了变化,则整个窗口将被重画。
CS_HREDRAW	水平风格也这样
CS_NOCLOSE	禁用系统菜单中的关闭命令
CS_DBLCLKS	如果未设置该参数,并且双击由该窗口类创建的窗口,则传送给应用程序的将不是双击事件,而是两个相继完成的单击事件

E.4 通用的控件窗口类

Windows 3.1及以上版本：

类	创建的窗口	类	创建的窗口
BUTTON	按钮控件窗口	LISTBOX	列表框控件窗口
STATIC	静态控件窗口	SCROLLBAR	滚动条控件窗口
EDIT	编辑控件窗口	COMBOBOX	组合框控件窗口

Windows 95/NT及以上版本：

类	创建的窗口
RICEDIT	多信息编辑控件窗口
SysListView32	列表视图控件窗口
ComboBoxEx32	扩展组合框控件窗口
SysAnimate32	动画控件窗口
msctls_trackbar32	滑块控件窗口
SysTreeView32	树型视图控件窗口
msctls_updown32	微调按钮控件窗口
msctls_progress32	进度指示控件窗口
SysHeader32	标题控件窗口(标题控件通常驻留在列表视图控件的顶部)
SysTabControl32	Tab控件窗口
SysMonthCal32	月历控件窗口
SysDateTimePick32	日期/时间选项控件窗口
msctls_hotkey32	热键控件窗口
Tooltips_class32	工具提示控件窗口
msctls_statusbar32	状态栏窗口
ToolbarWindow32	工具栏窗口
ReBarWindow32	Rebar窗口

E.5 处理消息的顺序

函 数	描 述
AfxWndProc()	接收消息,找到消息所属的CWnd对象,然后调用AfxCallWndProc()
AfxCallWndProc()	保存消息(消息标识符和参数)供以后参考,然后调用WindowProc()
WindowProc()	发送消息到OnWndMsg(),如果未被处理,则调用DefWindowProc()
OnWndMsg()	按字节对消息进行排序,对于WM_COMMAND消息,调用OnCommand();对于WM_NOTIFY消息,调用OnNotify()。任何被遗漏的消息将是一个窗口消息。OnWndMsg()搜索类的消息映像,以找到一个能处理任何窗口消息的处理函数。如果OnWndMsg()不能找到这样的处理函数,则把消息返回到WindowProc(),由它将消息发送给DefWindowProc()

(续)

函 数	描 述
OnCommand()	OnCommand() 查看这不是一个控件通知 (IParam不是NULL) ; 如果它是, OnCommand() 试图将消息映射到制造通知的控件; 如果它不是一个控件通知 (或者如果控件拒绝映射的消息), OnCommand() 调用OnCmdMsg ()
OnNotify()	OnNotify() 也试图将消息映射到制造通知的控件; 如果映射不成功, OnNotify() 调用相同的OnCmdMsg()函数
OnCmdMsg()	根据接收消息的类, OnCmdMsg()将在一个称为命令传递 (Command Routing) 的过程中潜在地传递命令消息和控件通知。例如, 如果拥有该窗口的类是一个框架类, 则命令和通知消息也被传递到视图和文档类, 并为该类寻找一个消息处理函数

E.6 创建窗口的顺序

消息处理函数/重载函数	类 型	描 述
PreCreateWindow()	O	窗口被创建前, 允许改变创建参数
PreSubclassWindow()	O	允许首先子分类一个窗口
OnGetMinMaxInfo()	M	允许设置窗口的最大和最小尺寸
OnNcCreate()	M	发送消息以告诉窗口的客户区即将被创建
OnNcCalcSize()	M	允许改变窗口客户区的大小
ONCreate()	M	发送此消息以告诉一个窗口已经被创建
OnSize()	M	发送此消息以告诉窗口大小发生变化
OnMove()	M	发送此消息以告诉窗口在移动
OnChildNotify()	O	作为部分消息映射被调用, 告诉你父窗口即将被告知一个窗口刚刚被创建

注 O表示重载函数; M表示消息处理函数。

E.7 关闭窗口顺序

消息处理函数/重载函数	类 型	描 述
OnClose()	M	当关闭按钮被单击时, 发送此消息
OnDestroy()	M	当一个窗口即将被销毁时, 发送此消息
OnNcDestroy()	M	当一个窗口被销毁以后, 发送此消息
PostNcDestroy()	O	作为处理OnNcDestroy()的最后动作被CWnd调用

注 O表示重载函数; M表示消息处理函数。

E.8 打开模式对话框的顺序

消息处理函数/重载函数	类 型	描 述
DoModal()	O	重载DoModal()成员函数
PreSubclassWindow()	O	允许首先子分类一个窗口
OnCreate()	M	发送此消息以告诉一个窗口已经被创建
OnSize()	M	发送此消息以告诉窗口大小发生变化
OnMove()	M	发送此消息以告诉窗口在移动
OnSetFont()	M	发送此消息以允许改变对话框中控件的字体
OnInitDialog()	M	发送此消息以允许初始化对话框中的控件, 或者是创建新控件

(续)

消息处理函数/重载函数	类 型	描 述
OnMove()	M	被发送以告诉窗口在移动
OnShowWindow()	M	被ShowWindow()函数发送
OnCtlColor()	M	被父窗口发送允许改变对话框的颜色
OnChildNotify()	O	作为WM_CTLCOLOR消息的结果被发送

注 O表示重载函数；M表示消息处理函数。

E.9 关闭模式对话框的顺序

消息处理函数	类 型	描 述
OnClose()	M	当关闭按钮被单击时，被发送
OnKillFocus()	M	在一个窗口即将失去键盘输入焦点以前被发送
OnDestroy()	M	当一个窗口即将被销毁时，被发送
OnNcDestroy()	M	当一个窗口被销毁以后，被发送
PostNcDestroy()	O	作为处理OnNcDestroy()的最后动作被CWnd调用

类型 O表示重载函数；M表示消息处理函数

E.10 打开无模式对话框的顺序

消息处理/重载函数	类 型	描 述
PreSubclassWindow()	O	允许用户首先子分类一个窗口
OnCreate()	M	发送此消息以告诉一个窗口已经被创建
OnSize()	M	发送此消息以告诉窗口大小发生变化
OnMove()	M	发送此消息以告诉窗口在移动
OnSetFont()	M	发送此消息以允许改变对话框中控件的字体

注 O表示重载函数；M表示消息处理函数

E.11 指针表

使用下面的函数可以访问的类

从该 类

从该 类	使用下面的函数可以访问的类
CWinApp	CWinApp AfxGetApp()或者theApp
CMainFrame	CMainFrame AfxGetMainWnd() 或者m_pMainWnd
CChildFrame	CChildFrame GetAfxMainWnd()-> MDIGetActive() MDIGetActive()或者 GetActiveFrame()
CDocument	CDocument GetAfxMainWnd()-> MDIGetActive()-> GetActiveView()-> GetDocument() GetActiveView()-> GetDocument()
CView	CView GetAfxMainWnd()-> MDIGetActive()-> GetActiveView() POSITION pos=Get FirstViewPosition(); GetNextView(pos); (参见下面的注)
其他类	if SDI: AfxGetMainWnd()-> GetActiveView() 否则 if MDI: AfxGetMainWnd()-> MDIGetActive()-> GetActiveView() if SDI: AfxGetMainWnd()-> GetActiveView()-> GetDocument() 或者if MDI: MDIGetActive()-> GetActiveView()-> GetDocument() if SDI: AfxGetMainWnd()-> MDIGetActive()-> GetActiveView()-> GetDocument() if SDI: AfxGetMainWnd()-> MDIGetActive()-> GetActiveView() 否则 if MDI: AfxGetMainWnd()-> MDIGetActive()-> GetActiveView() GetDocument()

注 若要访问当前视图，对于一个SDI应用程序，文档类调用GetAfxMainWnd()->GetActiveView(); 对于一个MDI应用程序，文档类调用GetAfxMainWnd()->MDIGetActive()->GetActiveView()。