## 5.1 菜单和加速键

菜单ID MENU [DISCARDABLE]

BEGIN

菜单项定义

…

END

MENUITEM “打开(&O)…\tCtrl+O”, 命令ID [,选项]

CHECKED,GRAYED,INACTIVE,MENUBAR,MENUBARBREAK

MENUITEM SEPARATOR

POPUP “菜单文字” [选项] GRAYED,INACTIVE,HELP

菜单项定义

…

END

加速键ID ACCELERATOR

BEGIN

“^字母”, 命令ID

“字母”, 命令ID, ASCII, [ALT|CTRL|SHIFT]

“VK\_XX”, 命令ID, VIRKEY, [ALT|CTRL|SHIFT]

ASCII值, 命令ID, ASCII, [ALT|CTRL|SHIFT]

END

## 5.4 对话框

**1. 对话框资源定义**

对话框ID DIALOG DISCARDABLE x, y, nWidth, nHeight

[属性]

BEGIN

{子窗口控件}

END

属性：CAPTION, CLASS, STYLE, EXTSTYLE, FONT, MENU

**2. 创建对话框**

INT\_PTR DialogBoxParam(hInstance, lpTemplateName, hWndParent, lpDialogFunc, dwInitParam)

HWND CreateDialogParam

**3. 对话框过程**

**4. 使用子窗口控件**

（1）字符串控件的定义

CONTROL 文本, ID, 类, 风格, x, y, nWidth, nHeight, 附加风格

空间名称 文本, ID, x, y, nWidth, nHeight, 风格, 附加风格

PUSHBUTTON

DEFPUSHBUTTON

CHECKBOX

AUTOCHECKBOX

STATE3

AUTO3STATE

RADIOBUTTON

AUTORADIOBUTTON

GROUPBOX

SCROLLBAR

CTEXT

LTEXT

RTEXT

EDITTEXT

ICON

COMBOBOX

LISTBOX

（2）通用方法

BOOL EnableWindow(hWnd, bEnable);

非0表示窗口以前无效，0表示以前有效

BOOL ShowWindow(hWnd, nCmdShow);

非0表示窗口以前可见，0表示以前隐藏

int GetWindowText(hWnd, lpString, nMaxCount);

nMaxCount包含NULL，返回值不包含NULL

BOOL SetWindowText(hWnd, lpString);

UINT GetDlgItemText(hDlg, nID, lpString, nMaxCount);

nMaxCount包含NULL，返回值不包含NULL

BOOL SetDlgItemText(hDlg, nID, lpString);

UINT GetDlgItemInt(hDlg,nID,lpTranslated,bSigned);

BOOL SetDlgItemInt(hDlg,nID,uValue,bSigned);

BOOL MoveWindow(hWnd,x,y,nWidth,nHeight,bRepaint);

HWND GetDlgItem(hDlg, nID);

int GetDlgCtrlID(hWnd);

LONG GetWindowLong(hWnd,GWL\_ID);

LRESULT SendDlgItemMessage(hDlg,nID,uMsg,wParam,lParam);

HWND GetNextDlgTabItem(hDlg,hCtl,bPrevious);

HWND GetNextDlgGroupItem(hDlg,hCtl,bPrevious);

（3）单选钮和复选钮

UINT IsDlgButtonChecked(hDlg,nID);

或发送BM\_GETCHECK消息

BOOL CheckDlgButton(hDlg,nID,uCheck);

或发送BM\_SETCHECK

BOOL CheckRadioButton(hDlg,idFirst,idLast,idCheck);

或者更简单的办法——使用AUTORADIOBUTTON。

（4）静态控件

对于文本框，文本长度超过边界的时候是自动换行的，如果指定SS\_SIMPLE就不会换行。

静态控件可以用来构筑简单的线条和图形。

CONTROL "", -1, "Static", SS\_ETCHEDHORZ | WS\_CHILD | WS\_VISIBLE, 60, 65, 110, 1

CONTROL BMP\_ID, IDC\_BMP, "Static", SS\_BITMAP | WS\_CHILD | WS\_VISIBLE, 5, 5, 40, 95

用来改变图片框图片的语句是：

STM\_SETIMAGE,IMAGE\_BITMAP IMAGE\_ICON IMAGE\_CURSOR hBmp

（5）文本编辑框

文本编辑框默认是单行的，如果指定ES\_MULTILINE就会变成多行的，这时可以加上一个WS\_HSCROLL风格显示一个垂直滚动条。

当文本编辑框有WS\_VSCROLL或WS\_HSCROLL时，可以向它发送EM\_LIMITTEXT消息来设定最大长度：

EM\_LIMITTEXT, maxLength, NULL

要初始为只读的，加ES\_READONLY，要在运行时变为只读的，可以发送EM\_SETREADONLY：

EM\_SETREADONLY, TRUE | FALSE, NULL

（6）滚动条

WM\_xSCROLL消息的参数如下所示：

wParam的低16位 ＝ nScrollCode ；动作码

wParam的高16位 ＝ nPos ；滚动条当前位置

lParam ＝ hwndScrollBar ；滚动条控件的窗口句柄

以水平滚动条为例，处理滚动条消息的代码一般是如下结构：

.elseif eax == WM\_HSCROLL ；窗口的消息处理分支，eax为wMsg

mov eax, lParam

.if eax == hWnd滚动条1

mov eax,wParam

.if ax == SB\_LINELEFT

dec 位置变量

.elseif ax == SB\_LINERIGHT

inc 位置变量

.elseif ax == SB\_PAGELEFT

sub 位置变量,页长

.elseif ax == SB\_PAGERIGHT

add 位置变量,页长

.elseif ax == SB\_THUMBPOSITION || ax == SB\_THUMBTRACK

mov eax,wParam

shr eax,16

mov 位置变量,eax

.endif

.elseif eax == hWnd滚动条2

；处理滚动条2的代码，同上面的结构

...

.endif

当计算好新位置的时候要将位置设置回去，用户才会看到滚动条移动了，方法是向滚动条发送SBM\_SETPOS消息：

invoke SendDlgItemMessage,hWnd,IDC\_SCROLL,SBM\_SETPOS,dwPos,TRUE

最后一个参数为TRUE表示设置后重新绘画滚动条。

在初始化的时候，要给滚动条发送SBM\_SETRANGE消息来设定滚动范围

invoke SendDlgItemMessage,hWnd,IDC\_SCROLL,SBM\_SETRANGE,最小值,最大值

（7）组合框

CBS\_SIMPLE CBS\_DROPDOWN CBS\_DROPDOWNLIST CBS\_AUTOHSCROLL CBS\_LOWERCASE CBS\_UPPERCASE CBS\_SORT

CB\_ADDSTRING

CBN\_SELENDOK CBN\_EDITUPDATE CBN\_CHANGE

（8）列表框

## 5.5 字符串资源

STRINGTABLE [DISCARDABLE]

BEGIN

字符串ID1 "字符串1"

字符串ID2 "字符串2"

. . .

END

int LoadString(hInstance,uID,lpBuffer,nBufferMax);

## 5.6 版本信息资源

## 5.7 二进制资源和自定义资源

二进制资源的定义格式是：

资源ID RCDATA [DISCARDABLE]

BEGIN

数据定义

…

END

也可以用一个磁盘文件当做资源的内容：

资源ID RCDATA [DISCARDABLE] 文件名

HRSRC FindResource(hModule,lpName,lpType DEFAULT(RT\_RCDATA));

HGLOBAL LoadResource(hModule,hResInfo);

LPVOID LockResource(hResData);

自定义资源和二进制资源的区别在于可以指定资源类别为自定义的名称：

资源ID 类型ID [DISCARDABLE]

BEGIN

数据定义

…

END

或用一个磁盘文件当做资源的内容：

资源ID 类型ID [DISCARDABLE] 文件名

类型ID可以是大于255的数值（255及以下的数值由Windows使用）或字符串，

# 第6章 定时器和Windows时间

UINT\_PTR SetTimer(hWnd,nIDEvent,uElapse,lpTimerFunc);//55ms

BOOL KillTimer(hWnd,uIDEvent);

void GetLocalTime(lpSystemTime);

void GetSystemTime(lpSystemTime);

BOOL SetLocalTime(lpSystemTime);

BOOL SetSystemTime(lpSystemTime);

DWORD WINAPI GetTickCount(void);//10ms

BOOL QueryPerformanceFrequency(lpFrequency);

BOOL QueryPerformanceCounter(lpPerformanceCount);