# 托盘功能的实现

原理：

1、Shell\_NotifyIcon函数

托盘功能的实现需要用到SHELLAPI函数Shell\_NotifyIcon来添加和删除图标，所以要在程序中包括头文件 ShellAPI.h，Shell\_NotifyIcon原型如下：

WINSHELLAPI BOOL WINAPI Shell\_NotifyIcon(

DWORD dwMessage,

PNOTIFYICONDATA pnid

);

其中参数dwMessage可以是NIM\_ADD、NIM\_DELETE、NIM\_MODIFY，分别是添加、删除、修改图标。

参数pnid是一个指向NOTIFYICONDATA的托盘图标数据结构体指针。

2、NOTIFYICONDATA结构体

NOTIFYICONDATA的定义如下：

typedef struct \_NOTIFYICONDATA {

DWORD cbSize;

HWND hWnd;

UINT uID;

UINT uFlags;

UINT uCallbackMessage;

HICON hIcon;

WCHAR szTip[64];

} NOTIFYICONDATA, \*PNOTIFYICONDATA;

其中重要的参数有：

uFlags：指定其他哪些参数有效，是如下取值的组合：

NIF\_MESSAGE：uCallbackMessage参数有效；

NIF\_ICON：hIcon参数有效；NIF\_TIP：szTip参数有效；

NIF\_STATE：dwState和dwStateMask参数有效；

NIF\_INFO：显示气泡通知，szInfo，szInfoTitle，dwInfoFlags和

uTimeout参数有效；

NIF\_GUID、NIF\_REALTIME、NIF\_SHOWTIP

常常将其设置为：NIF\_ICON|NIF\_MESSAGE|NIF\_TIP

uCallbackMessage：鼠标图标上点击时触发的消息

hIcon：图标

WCHAR szTip[64]：鼠标在图标上时显示的小气泡信息

3、弹出菜单的实现

添加菜单资源后，可用LoadMenu加载菜单资源，在需要时可用TrackPopupMenu弹出菜单。

4、实现流程

我们自定义一个消息，并用它为NOTIFYICONDATA的nCallbackMessage赋值。在窗口最小化时，我们可以用Shell\_NotifyIcon添加托盘图标并隐藏窗口，在窗口从图标还原和程序结束时，我们可以用Shell\_NotifyIcon来删除图标。当鼠标在图标上活动时，会触发自定义消息，我们在自定义消息处理函数中判断鼠标左键还是右键被按下，如果左键被按下，我们还原并激活窗口，用到的函数有ShowWindwo和SetForegroundWindow，如果是右键被按下，我们显示弹出的菜单，这里我们要用到鼠标的当前位置，获取鼠标当前位置的函数是GetCursorPos。

实现

1、定义托盘菜单句柄及托盘图标数据结构体：

HMENU hMenu=NULL; //托盘菜单句柄

NOTIFYICONDATA nid; //托盘图标数据结构体

2、定义一个自定义消息，该消息在托盘被点击时被触发：

#define WM\_MYMESSAGE WM\_USER+1

3、添加托盘菜单资源：



4、在WM\_CREATE消息处理中填充托盘数据结构体：

//托盘

nid.cbSize=sizeof(nid);

nid.hWnd=hWnd;

nid.uID=0;

nid.uFlags=NIF\_ICON|NIF\_MESSAGE|NIF\_TIP;

nid.hIcon=LoadIcon(hInst,MAKEINTRESOURCE(IDI\_MAIN));

nid.uCallbackMessage=WM\_MYMESSAGE;

lstrcpy(nid.szTip,szTitle);

//菜单

hMenu=LoadMenu(hInst,MAKEINTRESOURCE(IDR\_MENUTEST));

5、在窗口的WM\_SIZE中将窗口隐藏，并设置托盘图标：

if (wParam==SIZE\_MINIMIZED)

{

Shell\_NotifyIcon(NIM\_ADD,&nid);

ShowWindow(hWnd,SW\_HIDE);

}

6、在自定义消息WM\_MYMESSAGE的消息处理函数中，根据鼠标点击事件来弹出菜单或显示窗口：

if (lParam==WM\_LBUTTONDOWN)

{

ShowWindow(hWnd,SW\_SHOWNORMAL);

Shell\_NotifyIcon(NIM\_DELETE,&nid);

SetForegroundWindow(hWnd);

}

else if (lParam==WM\_RBUTTONDOWN)

{

POINT pt;

GetCursorPos(&pt);

SetForegroundWindow(hWnd);

TrackPopupMenu(GetSubMenu(hMenu,0),NULL,pt.x,pt.y,NULL,hWnd,NULL);

}

7、在窗口的WM\_DESTROY消息处理函数中删除图标：

Shell\_NotifyIcon(NIM\_DELETE,&nid);