根据位图显示自定义形状轮廓

1、重要函数说明：

1.1 GetPixel

COLORREF CDC::GetPixel(int x,int y)const;

COLORREF CDC::GetPixel(POINT point) const;

获取特定坐标点的颜色。

其中坐标在DC中被选中的位图上点的坐标。

1.2 CRgn::CombineRgn  
int CRgn::CombineRgn( CRgn\* pRgn1, CRgn\* pRgn2, int nCombineMode );  
此成员函数通过组合两个已有的区域来创建一个新的GDI区域。所组合成的区域就像nCombineMode所指定的一样。  
两个指定的区域被组合，所得到的结果区域的句柄被保存在CRgn对象中。因此，不管CRgn中原来保存的区域是什么，都将被合并后的区域所代替。

1.3 SetWindowRgn

int CDC::SetWindowRgn( HWND hWnd,HRGN hRgn, BOOL bRedraw );

为窗口设定RNG区域，这个区域用于绘制窗体，在该区域外，将不用于绘制。

2、在h文件中定义：

private:

int BmpWidth;

int BmpHeight;

CBitmap m\_bmp;

BITMAP bmp;

其中 m\_bmp对象用于载入位图，bmp对象用于存放位图信息，BmpWidth,BmpHeight用于存储位图的尺寸，其中位图的尺寸是这样得到的（InitialDlg函数中）：

m\_bmp.LoadBitmap(IDB\_BITMAP1);

m\_bmp.GetBitmap(&bmp);

BmpWidth=bmp.bmWidth;

BmpHeight=bmp.bmHeight;

**记住：BITMAP类对象用于存储位图信息，填充信息的函数是CBitmap::GetBitmap()。**

3、利用CDC::GetPiex函数可以得到被选入的位图的各坐标颜色，扫描整幅位图，将颜色为RGB(0,0,0)的区域合并，合并方式为RGN\_DIFF,即减去。最后在剩余的RGN设置到窗口，代码如下（InitalDlg中）：

CDC \*pDC,memDC;

pDC=GetDC();

memDC.CreateCompatibleDC(pDC);

memDC.SelectObject(&m\_bmp);

CRgn Rgn,rgn;

COLORREF col;

Rgn.CreateRectRgn(0,0,BmpWidth,BmpHeight);

for (int y=0;y<BmpHeight;y++)

{

for (int x=0;x<BmpWidth;x++)

{

col=memDC.GetPixel(x,y);

if (col==RGB(0,0,0))

{

rgn.CreateRectRgn(x-1,y-1,x+1,y+1);

Rgn.CombineRgn(&Rgn,&rgn,RGN\_DIFF);

rgn.DeleteObject();

}

}

}

SetWindowRgn((HRGN)Rgn,TRUE);

4、在OnEraseBkgnd(CDC\* pDC)函数中：

CDC memDC;

memDC.CreateCompatibleDC(pDC);

memDC.SelectObject(m\_bmp);

pDC->BitBlt(0,0,BmpWidth,BmpHeight,&memDC,0,0,SRCCOPY);

return TRUE;

记住最后直接return TRUE。

5、模拟鼠标拖动的方法：

在OnLButtonDown中：

PostMessage(WM\_NCLBUTTONDOWN,HTCAPTION,MAKELPARAM(point.x,point.y));

或者：

响应WM\_CHITTEST消息，直接返回HTCAPTION;

说明：

**更方便简单的方法是使用层窗口**