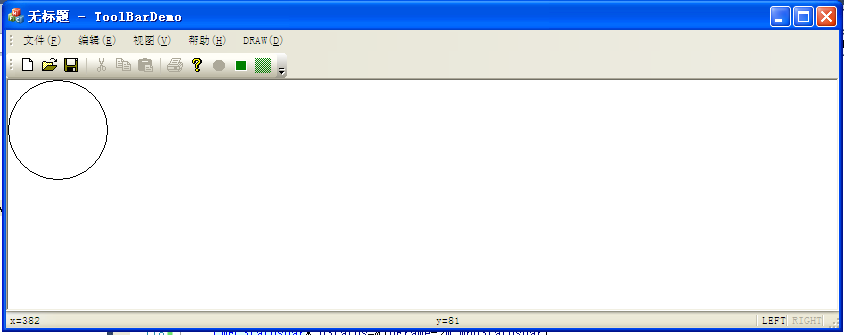
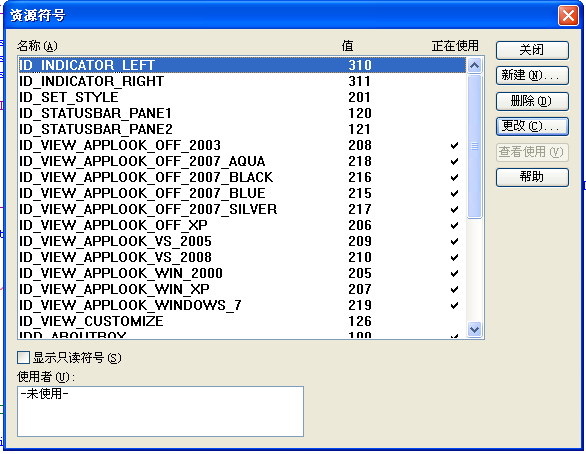
**状态栏的编程**

例程序自定义了一个状态栏，共有4格，第0格显示x坐标，第1格显示y坐标，第2、3格显示鼠标左右键是否按下，程序结果如图所示：



实现原理：

1、为资源添加新的ID符号ID\_INDICATOR\_LEFT和ID\_INDICATOR\_RIGHT：



并在资源的字符串表中为其添加标题，这是为设置窗口的预留空间。



2、修改indicators数组，在MainFrm.cpp文件中，

static UINT indicators[] =

{

ID\_SEPARATOR, // 状态指示器

ID\_SEPARATOR, // 状态指示器

ID\_INDICATOR\_LEFT,

ID\_INDICATOR\_RIGHT,

};

为indicatiors数组添加了两个状态指示ID\_SEPARATOR和两个空格ID\_INDICATOR\_LEFT和ID\_INDICATOR\_RIGHT。

3、为ID\_INDICATOR\_RIGHT和ID\_INDICATOR\_LEFT添加函数响应，由于Visual Stdio无法识别相应的命令消息ID，此处需要自已手动添加相关代码。

3.1 添加函数声明：

public:

afx\_msg void OnViewStatusBar();

afx\_msg void OnUpdateViewStatusBar(CCmdUI \*pCmdUI);

3.2 添加消息映射

ON\_UPDATE\_COMMAND\_UI(ID\_INDICATOR\_RIGHT,&CMainFrame::OnUpdateRight)

ON\_UPDATE\_COMMAND\_UI(ID\_VIEW\_STATUS\_BAR, &CMainFrame::OnUpdateViewStatusBar)

3.3 添加函数实现

void CMainFrame::OnUpdateLeft(CCmdUI\* pCmdUI)

{

pCmdUI->Enable(::GetKeyState(VK\_LBUTTON)<0);

}

void CMainFrame::OnUpdateRight(CCmdUI\* pCmdUI)

{

pCmdUI->Enable(::GetKeyState(VK\_RBUTTON)<0);

}

以上对状态栏的编程都是在原状态栏上修改的，在一些情况下还会出现问题，例如，当鼠标位于菜单和按钮上时，状态栏上还会提示相关信息。为此，应该重新再定义一个标题栏。

**改进：**

1、打开资源符号对话框，增加一个ID\_MY\_STATUS\_BAR，并在创建状态栏时使用以下代码：

if (!m\_wndStatusBar.Create(this,WS\_CHILD|WS\_VISIBLE|CBRS\_BOTTOM,ID\_MY\_STATUS\_BAR)||

!m\_wndStatusBar.SetIndicators(indicators,sizeof(indicators)/sizeof(UINT)))

{

TRACE0("未能创建状态栏\n");

return -1;

}

2、由于鼠标点击视图菜单下的状态栏按钮时，可以显示或隐藏状态栏，而由于现在使用的状态栏ID是自定义的，因此为完成这项功能，应该改写它的响应函数。

void CMainFrame::OnViewStatusBar()

{

m\_wndStatusBar.ShowWindow((m\_wndStatusBar.GetStyle()&WS\_VISIBLE)==0);

RecalcLayout();

}

当状态栏处理显示状态时，GetStyle得到的返回值包含WS\_VISIBLE标志，与WS\_VISIBLE做&运算，值为1，相反，若状态栏处理不可见状态，其返回值为0，0==0为1，因此下一个状态即可设置为可见。

3、根据状态栏是否显示，设置状态栏菜单的勾选状态。

void CMainFrame::OnUpdateViewStatusBar(CCmdUI \*pCmdUI)

{

pCmdUI->SetCheck((m\_wndStatusBar.GetStyle()&WS\_VISIBLE)!=0);

}