# MoveWindow

**void** **MoveWindow(** **int** *x***,** **int** *y***,** **int** *nWidth***,** **int** *nHeight***,** **BOOL** *bRepaint* **=** **TRUE** **);**

**void** **MoveWindow(** **LPCRECT** *lpRect*, **BOOL** *bRepaint* = **TRUE** **);**

MoveWindow用来改变指定窗口的位置和大小, 其坐标系为：

* 对子窗口(Style: Child)来说, 父窗口客户区的左上角为坐标(0, 0)点, 一般弹出式窗口内部的控件都是子窗口。一般要配合SetParent()函数使用;
* 对弹出式窗口(Style: Popup)来说, 屏幕左上角为坐标(0, 0)点。一般主窗口可以是Popup窗口或Overlapped窗口。
* 对重叠式窗口(style: Overlapped)来说，同弹出式窗口。

设备坐标系统: (默认使用MM\_TEST坐标映射模式：左上角是(0, 0)，向右x轴增加，向下y轴增加)

* 屏幕坐标： 屏幕的左上角为(0, 0);
* 全窗口坐标：整个应用程序的左上角为(0, 0), 包括标题栏，菜单等，一般很少用。

窗口类内this-> GetWindowRect()可获得它或者使用窗口的ID。

* 客户区坐标：客户区的左上角为(0, 0)

获取窗口的位置和大小的函数有：

* **void** **GetWindowRect(LPRECT** *lpRect***) : 获取当前窗口屏幕坐标**
* **void** **GetClientRect(LPRECT** *lpRect***) : 获取窗口的的矩形大小， 一般*lpRect->top= lpRect->left = 0***
* **void** **ScreenToClient(LPRECT** *lpRect* **)** ：**屏幕坐标转换到客户区坐标**
* **void** **ClientToScreen(LPRECT** *lpRect* **) ：客户区坐标转换到屏幕坐标**
* **BOOL GetCursorPos( LPPOINT** *lpPoint***):获取鼠标的屏幕坐标**

eg：改变窗口IDC\_RADIO\_LEN\_1BYTE的宽度（有些繁琐，但这是清晰的思路）：

RECT boolRect;

//以下紧邻的三步为：获取子窗口在父窗口的坐标和位置

GetDlgItem(IDC\_RADIO\_LEN\_1BYTE)->GetWindowRect(&boolRect); //获取要移动窗口的屏幕坐标

GetDlgItem(IDC\_RADIO\_LEN\_1BYTE)->SetParent(this); //设置其父窗口

this->ScreenToClient(&boolRect); //转换屏幕坐标到父客户区坐标

boolRect.right = boolRect.left + 200; //该变宽度

GetDlgItem(IDC\_RADIO\_LEN\_1BYTE)->MoveWindow(&boolRect); //移动窗口

**以上函数全部是CWnd类的成员。**

处理**WM\_SIZE**消息：

void xxx::OnSize(UINT nType, int cx, int cy)

其中cx和cy指的是客户区的宽和高。

**注意：**

MoveWindow(rt)在移动dialog窗口的时候指定的矩形rt是指：标题栏+客户区+状态栏+菜单栏+边框+etc。

但是在该dialog的WM\_SIZE消息处理函数中传进来的cx和cy是only客户区！

所以：

cx = rt.Width() - 2\*GetSystemMetrics(SM\_CXFRAME);

cy = rt.Heigth() - GetSystemMetrics(SM\_CYCAPTION) + 2\*GetSystemMetrics(SM\_CYFRAME)

其中：

SM\_CYCAPTION：标题栏的高度；

SM\_CYFRAME：边框的高度