

MFC的坐标转换GetClientRect/GetWindowRect/ClientToScreen/GetCursorPos/ScreenToClient

注意：windows的屏幕左上点为基点，也即原点，向右x轴正向，向下y轴正向

注意：区分两个概念 (假设显示器1024*768)

屏幕-左上点：1024*768的屏幕[b](显示器)，屏幕(显示器)的左上点 就是 (0,0)[/b]

窗口-左上点：指我们所打开程序窗口的客户区左上点，窗口可能居中展示，此时窗口左上点和屏幕左上点是不一致的。例如：我们双击了notepad图标，它弹出了Notepad窗口，窗口的大小是600 * 400(假设)，位于屏幕中央。Notepad窗口的左上点计算为：(1024-600)/2, (768-400)/2 --**(212, 184)**，也就是窗口左上点对于 显示器来说，坐标是(212, 184)，当然相对于其本身来说仍是(0,0)了。

1. MFC多数方法传入的point，他们的取值为相对的坐标(以当前"窗口客户区左上点"为基点的相对坐标)

```
void CDlgSelectCity::OnLButtonDown(UINT nFlags, CPoint point)
```

转换为windows的坐标，使用**this->ClientToScreen**，把相对于**this窗口客户区**的坐标转换为以当前屏幕的**[b]左上点**为基点的坐标[/b]

```
ClientToScreen(&point);
```

+ point {x=489 y=132} CPoint 转换前相对于窗口的坐标

+ screentPoint {x=678 y=251} CPoint 转换后相对于屏幕的坐标

(注意：使用哪个组件或窗口.clientToScreen，得出那个窗口坐标向屏幕坐标的转换，例如m_statPic.clientToScreen转换staticPic组件内部的坐标到屏幕坐标)

2. GetCursorPos所获取到的坐标，为以"屏幕客户区左上点"为基点的坐标

```
//也或我们获取的CPoint
```

```
POINT apoint;
```

```
GetCursorPos(&apoint);
```

+ apoint {x=678 y=251} tagPOINT

3. GetClientRect 获取的为相对于自身的客户区范围（相对于自身窗口客户区而言，左上点为[0,0]，可以看到获取的结果 top=0, left=0)

获取的为相对于自身左上点的left,right,top,bottom。Points to a RECT structure or aCRect object to receive the client coordinates. Theleft andtop members will be 0. Theright

andbottom members will contain the width and height of the window. m_PicDraw.GetClientRect(&picClientRect);+ picClientRect {top=0 bottom=512 left=0 right=690} tagRECT

注意：像Dialog的窗口，它的GetClientRect的范围是比GetWindowRect取出的范围小的，因为GetClientRect去除了边界的信息：提供的是可供使用的内部大小。The client coordinates specify the upper-left and lower-right corners of the client area.

如下面这个：

+ clientRect {top=0 bottom=510 left=0 right=902} CRect 相对于自身的范围

4. GetWindowRect获取到的为相对于“屏幕左上点”的范围(包括其边框及客户区所有范围)

获取的为相对于屏幕左上点的left,right,top,bottom范围的Points to a**CRect** object or a RECT structure that will receive the**screen** coordinates of the upper-left and lower-right corners.

m_PicDraw.GetWindowRect(&picWindowRect);

+ picWindowRect {top=119 bottom=631 left=401 right=1091} tagRECT

5. ScreenToClient/ClientToScreen坐标系转换

5.1 Client-area/Screen coordinates

这两个函数完成相对坐标系的转换

The **ScreenToClient** function converts the**screen coordinates** of a specified point on the screen to**client-area coordinates**

The **ClientToScreen** function converts the**client-area coordinates** of a specified point to**screen coordinates**.

注意：哪个组件来调用，就会参考哪个组件的坐标系

m_static.**ClientToScreen** 就会 参考m_static组件的当前位置，转换出来 新的坐标

例如m_static左上点在屏幕的(100,100)处，(0,0)(100,100)转换出来为(100,100)(200,200)

m_static.**ScreenToClient** 同样是以 m_static组件的当前位置，作为参考值

例如m_static左上点在屏幕的(100,100)处，(200,200)(300,300)转换出来为(100,100)(200,200)

使用m_static.ScreentToClient和 this->ScreentToClient 转同样的数据结果是不同的，原因就是参考坐标 是 调用方的左上点.

client-area coordinates

组件相当于 窗口/组件 区域内左上点（此处非窗口的左上点，为窗口内部区域的左上点）的坐标。

常用与动态创建组件的时候，例如XTP库中很多组件create的时候，都会需要一个CRect参数，这个参数的坐标系就是**client-area coordinates**

screen coordinates

组件相当于屏幕左上点的坐标。

通过GetWindowRect所获取的为该坐标系的数据。

5.2 应用场景-坐标转换

当创建一个XTP组件的时候，如果参考另一组件的位置，我们可以使用m_static.GetWindowRect获得屏幕坐标，在通过this->ScreenToClient获取区域坐标。

区域坐标就可以用于创建XTP组件。

5.3 应用场景-坐标转换-移动窗口内组件

```
CRect rect;

m_edit.GetWindowRect(rect); // 获取m_edit的屏幕坐标

ScreenToClient(rect); // 屏幕坐标调整到<strong>this窗体</strong>的相对坐标上类
.... // rect调整

m_edit.MoveWindows(rect); // m_edit移动窗口位置
```

Owed by: 春夜喜雨 <http://blog.csdn.net/chunyexiyu> (<http://blog.csdn.net/chunyexiyu>) 转载请标明来源

