

Construct an ActiveX control that displays the following window. The border is required and the horizontal and vertical lines should bisect the window.



Use the following steps to create your control.

1. Create a new control. (Don't modify any of your previous efforts, start from scratch!
2. In the OnDraw routine, put the following lines AT THE VERY BEGINNING of the routine.

```
// Set the line color
CPen MyPen;
MyPen.CreatePen(PS_SOLID,0,RGB(0,0,0));
CPen * OldPen = pdc->SelectObject(&MyPen);
// Set the background color
CBrush MyBrush;
MyBrush.CreateSolidBrush(RGB(255,255,255));
CBrush * OldBrush = pdc->SelectObject(&MyBrush);
//Drawing Commands follow this line
```
3. To draw the rectangle, use the following drawing command. Note that rcBounds is a parameter to the OnDraw routine. rcBounds gives the screen coordinates of your control's window, and this command draws a rectangle around the window.

```
pdc->Rectangle(&rcBounds);
```
4. To draw the cross hairs, you must compute the horizontal and vertical centers of the window. Use the following code to do this.

```
// Note: vertical coordinates get larger as you go down.
long VertMid = rcBounds.left + ((rcBounds.right-rcBounds.left+1)/2);
long HorzMid = rcBounds.top + ((rcBounds.bottom-rcBounds.top+1)/2);
```
5. Use these two new coordinates to draw the lines, using the following commands.

```
pdc->MoveTo(VertMid,rcBounds.top);
pdc->LineTo(VertMid,rcBounds.bottom);
pdc->MoveTo(rcBounds.left,HorzMid);
pdc->LineTo(rcBounds.left,HorzMid);
```
6. After drawing the cross hairs, YOU MUST add the following cleanup code to restore the previous line and background colors, and delete the system objects you created in step 2 above.

```
pdc->SelectObject(OldPen);
pdc->SelectObject(OldBrush);
MyBrush.DeleteObject( );
MyPen.DeleteObject( );
```
7. Compile your control and test it at this point, to make sure it draws the box correctly.

8. Now we will add a feature to draw the box using fat lines instead of thin ones.
9. First add the following variable to your CxCtrl class, where *x* is the name of your project. (DO NOT MAKE THIS A GLOBAL VARIABLE!!!!!!!!!!!!!!!!!!!!!!)  
    long FatLines;
10. Add the following line to the constructor of your CxCtrl class, where *x* is the name of your project.  
    FatLines = 0;
11. Go back to your OnDraw routine, and find the line that reads as follows:  
    MyPen.CreatePen(PS\_SOLID,0,RGB(0,0,0));  
Replace this line with the following *if* statement.  
    if (FatLines)  
    {  
        // parameter 2 is the only difference  
        MyPen.CreatePen(PS\_SOLID,5,RGB(0,0,0));  
    }  
    else  
    {  
        MyPen.CreatePen(PS\_SOLID,0,RGB(0,0,0));  
    }
12. Compile your control, and test it at this point. NOTHING SHOULD CHANGE in the drawing.
13. Go to the View Menu and select “Class Wizard.” Go to the third tab, Automation, as in project 2, and click the *Add Method* button.
14. Type “Fat” in the *External Name* box, and select *void* from the *Return Type* box. (Type a *v* in this box to quick-select *void*.)
15. In the Parameters box, type *IsFat* in the Name column, and *long* in the Type column. (Just an *l* will do for the type.)
16. Click *OK*, then click *Edit Code*.
17. In the Fat subroutine, replace the *TODO* line with the following two lines.  
    FatLines = IsFat;  
    Invalidate( );
18. Compile your control and test it. Note, The border may not get as fat as the cross-hairs. This is OK.
19. Turn in “The Usual”