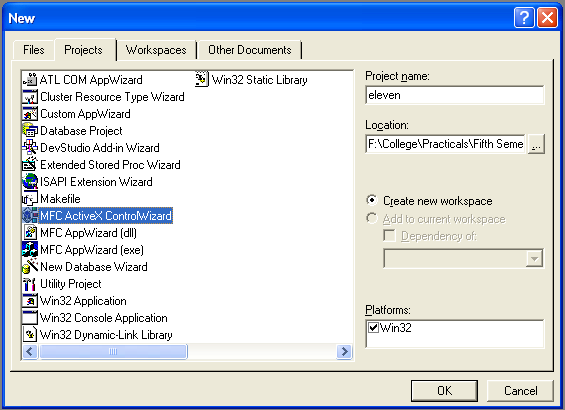
**STEPS**

1. Open Visual C++ and click the New item in the File menu and a new dialog box is opened
2. Now select “MFC ActiveX ControlWizard” entry
3. Give the new Project name in the Project name box and click OK. Clicking OK will start the Visual C++ AppWizard



1. We will accept all defaults settings. Keep pressing next until project information box is opened
2. Open OnDraw() method from the CElevenCtrl and remove the code “pdc->Elipse(rcBounds);” because it will draw ellipse in the rectangle

void CElevenCtrl::OnDraw(CDC\* pdc, const CRect& rcBounds, const CRect& rcInvalid)

{

// TODO: Replace the following code with your own drawing code.

pdc->FillRect(rcBounds,CBrush::FromHandle( (HBRUSH)GetStockObject(WHITE\_BRUSH)));

pdc->Elipse(rcBounds);

.

.

.

.

}

1. Open ElevenCtl.h from header files:

class CElevenCtrl : public COleControl

{

public:

.

.

.

protected:

~CElevenCtrl();

* CRect box1;
* CRect box2;
* CRect box3;
* CRect box4;

.

.

.

.

}

1. Open OnDraw() method from the CElevenCtrl from class view tab:

void CElevenCtrl::OnDraw(CDC\* pdc, const CRect& rcBounds, const CRect& rcInvalid)

{

// TODO: Replace the following code with your own drawing code.

pdc->FillRect(rcBounds,CBrush::FromHandle( (HBRUSH)GetStockObject(WHITE\_BRUSH)));

box1=CRect(rcBounds.left,rcBounds.top,rcBounds.right/2, rcBounds.bottom/2);

box2=CRect(rcBounds.left,rcBounds.bottom/2,rcBounds.right/2, rcBounds.bottom);

box3=CRect(rcBounds.right/2,rcBounds.top,rcBounds.right, rcBounds.bottom/2);

box4=CRect(rcBounds.right/2,rcBounds.bottom/2,rcBounds.right, rcBounds.bottom);

.

.

.

.

}

1. To draw the rectangles open OnDraw() method and write these lines of code:

void CElevenCtrl::OnDraw(CDC\* pdc, const CRect& rcBounds, const CRect& rcInvalid)

{

.

.

.

.

box4=CRect(rcBounds.right/2,rcBounds.bottom/2,rcBounds.right, rcBounds.bottom);

* pdc->Rectangle(&box1);
* pdc->Rectangle(&box2);
* pdc->Rectangle(&box3);
* pdc->Rectangle(&box4);

}

1. Now we add a mouse event handler to a ActiveX Control. Open class wizard and add LBUTTONDOWN() method to the ActiveX Control
2. Now open ElevenCtl.h from the header files:

class CElevenCtrl : public COleControl

{

public:

.

.

.

.

protected:

.

.

.

CRect box4;

* boolean fill1;
* boolean fill2;
* boolean fill3;
* boolean fill4;

.

.

.

.

}

1. Open ElevenCtl.cpp from source files folder in file view tab:

CElevenCtrl::CElevenCtrl()

{

InitializeIIDs(&IID\_DEleven, &IID\_DElevenEvents);

* fill1=fill2=fill3=fill4=false;

// TODO: Initialize your control's instance data here.

}

1. Open ElevenCtl.cpp from source files folder in file view tab to write lines of code in LBUTTONDOWN() method:

void CElevenCtrl::OnLButtonDown(UINT nFlags, CPoint point)

{

// TODO: Add your message handler code here and/or call default

* fill1=box1.PtInRect(point);
* fill2=box2.PtInRect(point);
* fill3=box3.PtInRect(point);
* fill4=box4.PtInRect(point);
* Invalidate();

COleControl::OnLButtonDown(nFlags, point);

}

1. Now open OnDraw() method to check the four flags and fill the corresponding rectangle using FillSolidRect() method:

void CElevenCtrl::OnDraw(CDC\* pdc, const CRect& rcBounds, const CRect& rcInvalid)

{

.

.

.

.

pdc->Rectangle(&box4);

* if(fill1)pdc->FillSolidRect(&box1,RGB(0,0,0));
* if(fill2)pdc->FillSolidRect(&box2,RGB(0,0,0));
* if(fill3)pdc->FillSolidRect(&box3,RGB(0,0,0));
* if(fill4)pdc->FillSolidRect(&box4,RGB(0,0,0));

}

1. For testing an ActiveX Control first build Eleven.ocx from Build menu. After this click Tools menu and select “ActiveX Control Test Container”. A window will appear and click Edit menu of that window and select “Insert New Control”, a list will appear, select “Eleven Control” item from list box. Now we are ready to check our ActiveX Control.

**OUTPUT**

