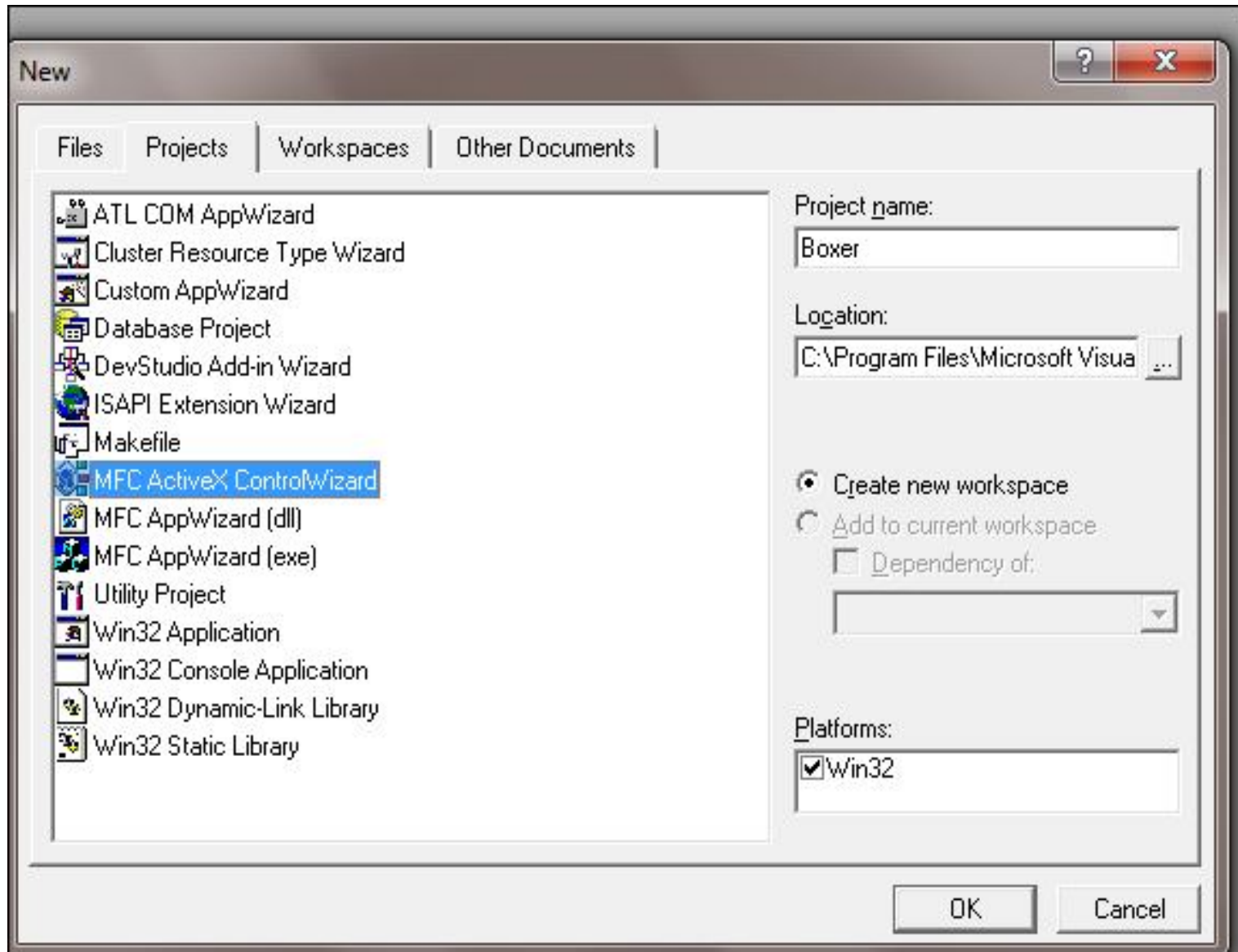


Active X Controls



MFC ActiveX ControlWizard - Step 1 of 2



How many controls would you like your project to have?

Would you like the controls in this project to have a runtime license?

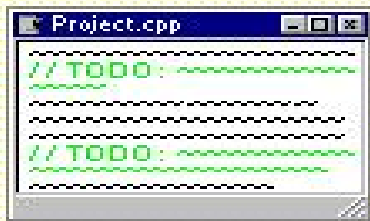
- Yes, please
- No runtime license

Would you like source file comments to be generated?

- Yes, please
- No comments

Would you like help files to be generated?

- Yes, please
- No help files



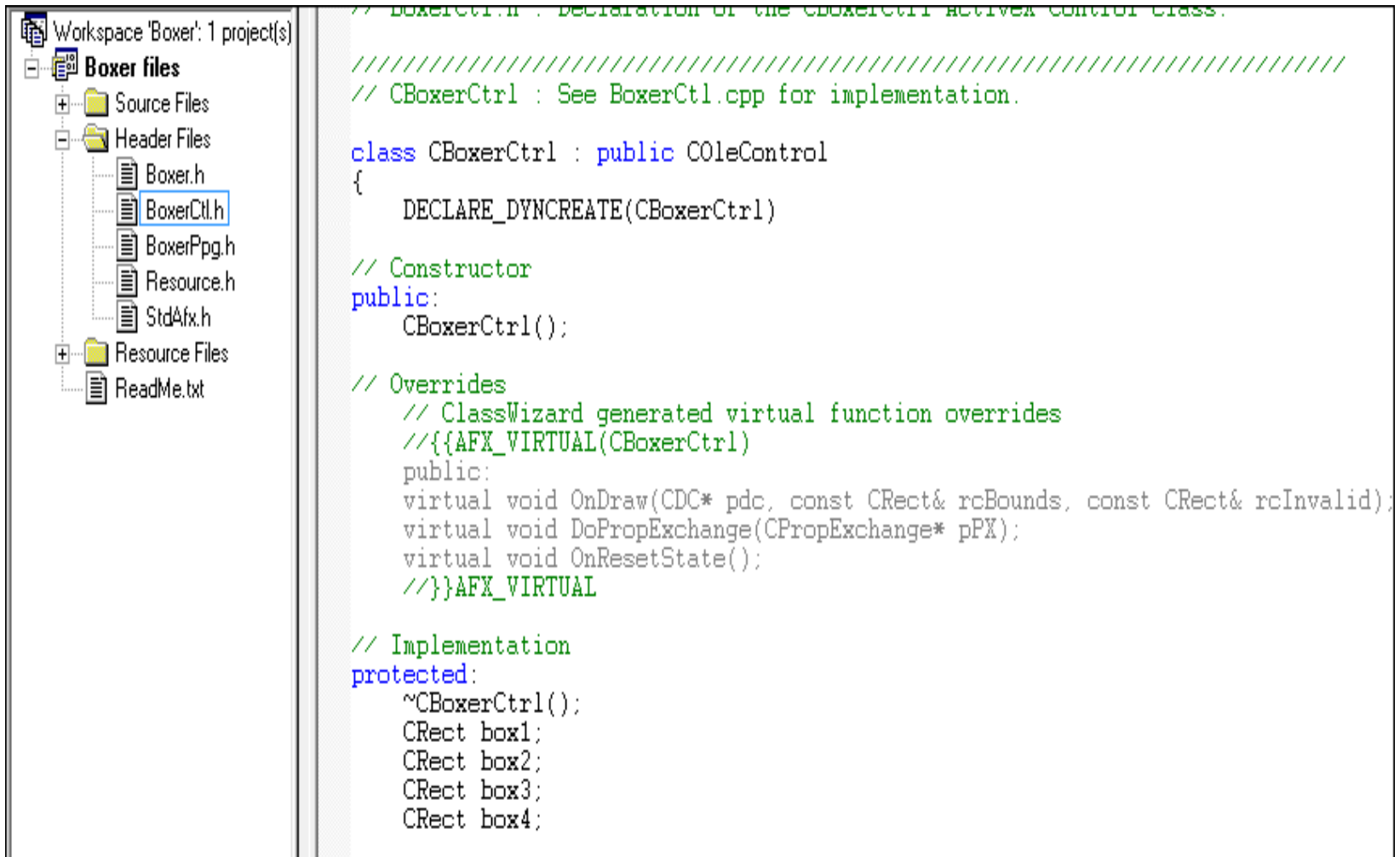
< Back

Next >

Finish

Cancel

BoxerCtl.h



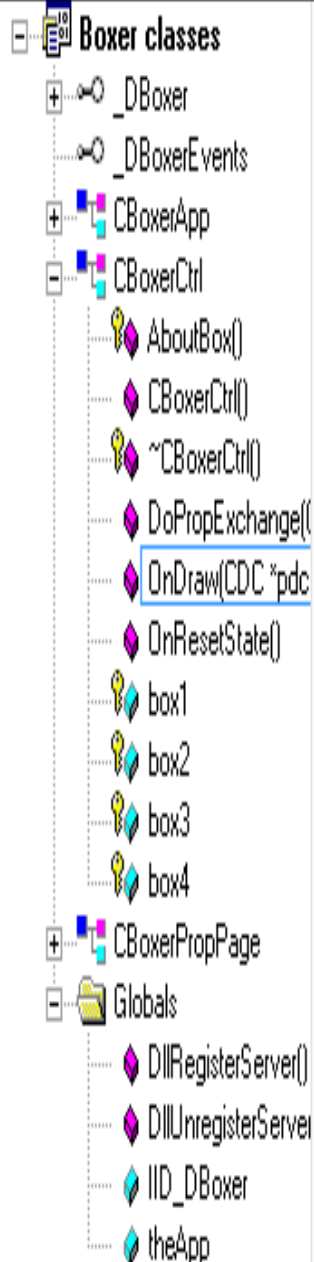
```
// BOXERCTL.H : Declaration of the CBoxerCtrl ActiveX control class.
//
/////////////////////////////////////////////////////////////////////////////
// CBoxerCtrl : See BoxerCtl.cpp for implementation.

class CBoxerCtrl : public COleControl
{
    DECLARE_DYNCREATE(CBoxerCtrl)

// Constructor
public:
    CBoxerCtrl();

// Overrides
// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CBoxerCtrl)
public:
    virtual void OnDraw(CDC* pdc, const CRect& rcBounds, const CRect& rcInvalid);
    virtual void DoPropExchange(CPropExchange* pPX);
    virtual void OnResetState();
//}}AFX_VIRTUAL

// Implementation
protected:
    ~CBoxerCtrl();
    CRect box1;
    CRect box2;
    CRect box3;
    CRect box4;
};
```



```
void CBoxerCtrl::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);

    pdc ->Rectangle(&box1);

    pdc ->Rectangle(&box2);

    pdc ->Rectangle(&box3);

    pdc ->Rectangle(&box4);

    pdc->Ellipse(rcBounds);
}
```

MFC ClassWizard

Message Maps | Member Variables | Automation | ActiveX Events | Class Info

Project:

Boxer

Class name:

CBoxerCtrl

Add Class...

Add Function

C:\...\Boxer\BoxerCtrl.h, C:\...\Boxer\BoxerCtrl.cpp

Delete Function

Object IDs:

CBoxerCtrl

Messages:

WM_INITMENU
WM_INITMENUPOPUP
WM_KEYDOWN
WM_KEYUP
WM_KILLFOCUS
WM_LBUTTONDOWNBLCLK
WM_LBUTTONDOWN

Edit Code

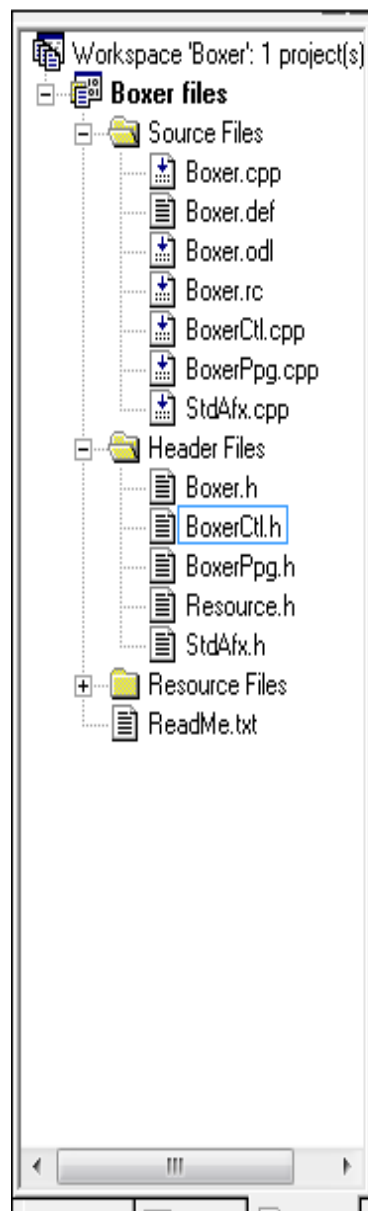
Member functions:

V DoPropExchange
V OnDraw
W OnLButtonDown ON_WM_LBUTTONDOWN
V OnResetState

Description: Indicates when left mouse button is pressed

OK

Cancel



```
class CBoxerCtrl : public COleControl
{
    DECLARE_DYNCREATE(CBoxerCtrl)

// Constructor
public:
    CBoxerCtrl();

// Overrides
// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CBoxerCtrl)
public:
    virtual void OnDraw(CDC* pdc, const CRect& rcBounds, const CRect& rcInvalid);
    virtual void DoPropExchange(CPropExchange* pPX);
    virtual void OnResetState();
//}}AFX_VIRTUAL

// Implementation
protected:
    ~CBoxerCtrl();

    CRect box1;

    CRect box2;

    CRect box3;

    CRect box4;

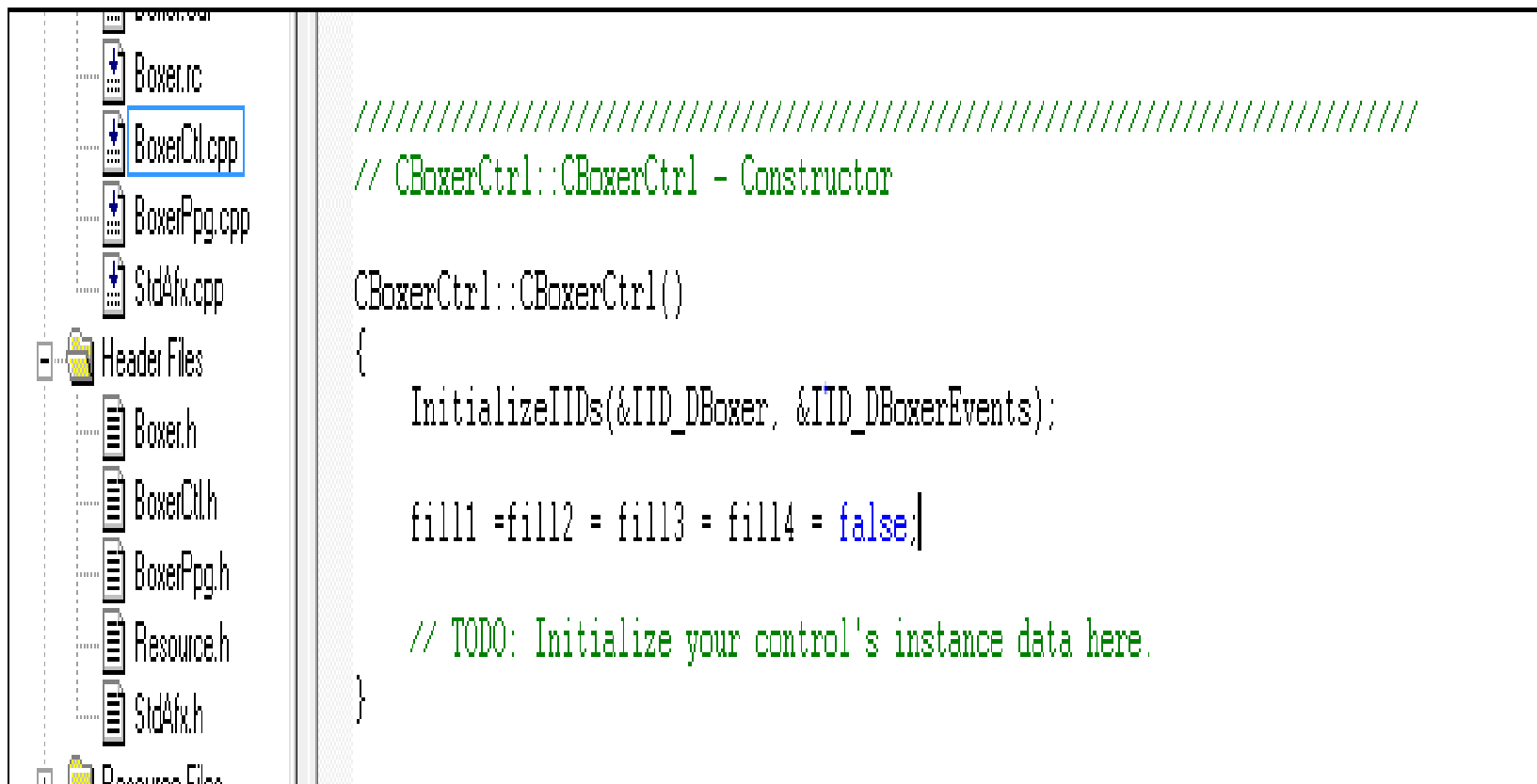
    boolean fill1;

    boolean fill2;

    boolean fill3;

    boolean fill4;
};
```

Initialization in the constructor



```
////////////////////////////////////  
// CBoxerCtrl::CBoxerCtrl - Constructor  
  
CBoxerCtrl::CBoxerCtrl()  
{  
    InitializeIDs(&IID_DBoxer, &IID_DBoxerEvents);  
  
    fill1 =fill12 = fill13 = fill14 = false;  
  
    // TODO: Initialize your control's instance data here.  
}
```


- ◆ CBoxerCtrl
- ◆ ~CBoxerCtrl()
- ◆ DoPropExchange()
- ◆ OnDraw(CDC *pdc)
- ◆ OnLButtonDown(UINT)
- ◆ OnResetState()
- ◆ box1
- ◆ box2
- ◆ box3
- ◆ box4
- ◆ CBoxerPropPage
- ◆ Globals
 - ◆ DllRegisterServer()
 - ◆ DllUnregisterServer
 - ◆ IID_DBoxer
 - ◆ theApp

```

////////////////////////////////////
// CBoxerCtrl message handlers

void CBoxerCtrl::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);
}

```

CBoxerCtrl

(All class members)

OnDraw

Workspace 'Boxer': 1 project(s)

- Boxer files
 - Source Files
 - Boxer.cpp
 - Boxer.def
 - Boxer.odl
 - Boxer.rc
 - BoxerCtrl.cpp
 - BoxerPpg.cpp
 - StdAfx.cpp
 - Header Files
 - Boxer.h
 - BoxerCtrl.h
 - BoxerPpg.h
 - Resource.h
 - StdAfx.h
 - Resource Files
 - ReadMe.txt

```
// CBoxerCtrl::OnDraw - Drawing function

void CBoxerCtrl::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);

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

    pdc->Ellipse(rcBounds);
}
```

Testing an ActiveX control in a visual C++ program

- Create a dialog based program named "Control".
- Select -> PROJECT -> Add to Project -> Components and Controls.
- Double click on newly created Active X control -> Insert -> Ok.
- Drag and drop the Active X control on the dialog editor.
- Execute the program.
- The controls works as expected.