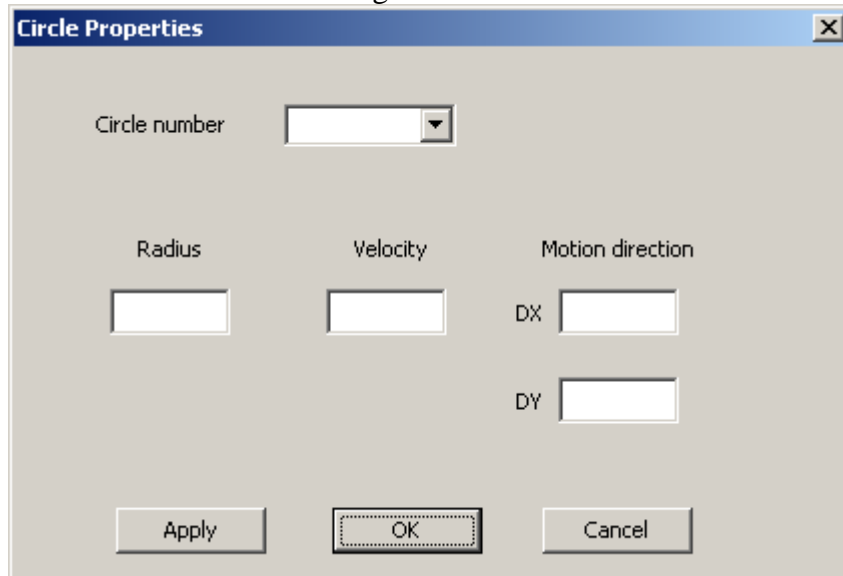


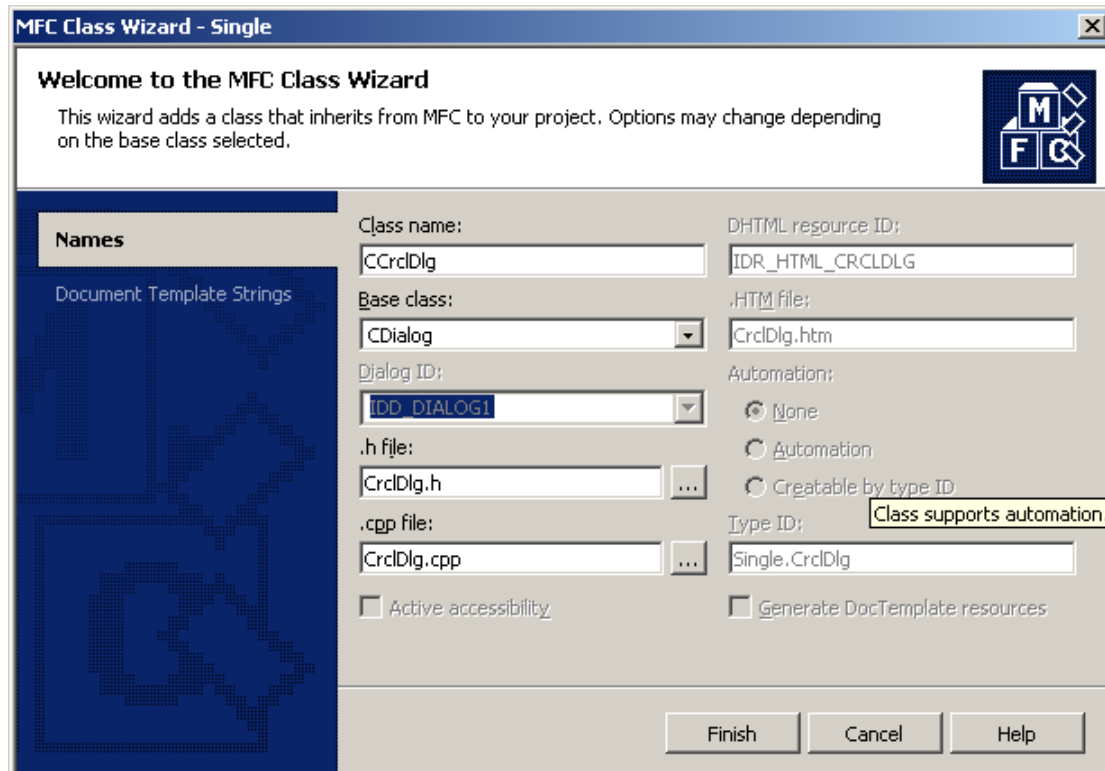
Resource view – insert dialog:



The 'Circle Properties' dialog box has a title bar with a close button. It contains a 'Circle number' dropdown menu. Below it are three columns: 'Radius' with a text box, 'Velocity' with a text box, and 'Motion direction' with 'DX' and 'DY' text boxes. At the bottom are 'Apply', 'OK', and 'Cancel' buttons.

Set “Visible” property to true.

Add class:



The 'MFC Class Wizard - Single' dialog box has a title bar with a close button. It contains a 'Welcome to the MFC Class Wizard' section with a description and an MFC logo. Below is a 'Names' section with a tree view showing 'Document Template Strings'. The main area has fields for 'Class name' (CCrclDlg), 'Base class' (CDialog), 'Dialog ID' (IDD\_DIALOG1), '.h file' (CrclDlg.h), and '.cpp file' (CrclDlg.cpp). It also has fields for 'DHTML resource ID' (IDR\_HTML\_CRCLDLG), '.HTML file' (CrclDlg.htm), 'Automation' (None selected), 'Type ID' (Single.CrclDlg), and checkboxes for 'Active accessibility' and 'Generate DocTemplate resources'. A tooltip 'Class supports automation.' is visible over the 'Automation' section. At the bottom are 'Finish', 'Cancel', and 'Help' buttons.

Variables for Edit controls:

int Radius (min 10, max 100), float Velocity (min 1, max200), DeltaX, DeltaY;

for Combo Box – int CircleNo:

Add Member Variable Wizard - Single

Welcome to the Add Member Variable Wizard  
This wizard adds a member variable to your class, struct, or union.

Access:	<input type="checkbox"/> Control variable	
Variable type:	Control ID:	Category:
float	IDC_EDIT3	Value
Variable name:	Control type:	Max chars:
Velocity	EDIT	
	Min value:	Max value:
	1	200
	.h file:	.cpp file:
Comment (// notation not required):		
Finish Cancel Help		

Add Member Variable Wizard - Single

Welcome to the Add Member Variable Wizard  
This wizard adds a member variable to your class, struct, or union.

Access:	<input checked="" type="checkbox"/> Control variable	
Variable type:	Control ID:	Category:
int	IDC_COMBO1	Value
Variable name:	Control type:	Max chars:
CircleNo	COMBOBOX	
	Min value:	Max value:
	.h file:	.cpp file:
Comment (// notation not required):		
Enter the comment to appear above the v declaration.		
Finish Cancel Help		

In class CView define CCrclDlg variable Dlg (insert `#include "Crcldlg.h"`). Add variable `bool DlgState` (1 – dialog is activated, 0 – not). Add dialog initialization in CView constructor:

```
CSingleView::CSingleView(): DlgState(false), Dlg(this).
```

In class CCrclDlg override `OnInitDialog()`:  
`BOOL CCrclDlg::OnInitDialog()`

```

{
    CDialog::OnInitDialog();

    // TODO: Add extra initialization here
    CSingleDoc *pDoc = ((CSingleView*)m_pParentWnd)->GetDocument();
    CComboBox *pCB = (CComboBox*) GetDlgItem(IDC_COMBO1);
    // Initialize ComboBox
    CString str;
    for (int i = 0; i < pDoc->arrayBall.GetSize(); i++)
    {
        str.Format(_T("circle %d"), i+1);
        pCB->AddString(str);
    }
    pCB->SetCurSel(CircleNo);

    SetDlgData();

    return TRUE; // return TRUE unless you set the focus to a
control
    // EXCEPTION: OCX Property Pages should return FALSE
}

```

In class `CCrclDlg` add member function `void SetDlgData(void)` and add `#include "SingleDoc.h"`, `#include "SingleView.h"`, `#include "Ball.h"`:

```

void CCrclDlg::SetDlgData()//int i /*=0*/
{
    CSingleDoc *pDoc = ((CSingleView*)m_pParentWnd)->GetDocument();
    if (pDoc->arrayBall.GetSize()>0)
    {
        CBall *pBall = (CBall*) pDoc->arrayBall[CircleNo];
        DeltaX = pBall->DeltaX;
        DeltaY = pBall->DeltaY;
        Radius = pBall->Radius;
        Velocity = pBall->Velocity;
        UpdateData(FALSE);
    }
}

```

In class `CCrclDlg` add member function `CloseDialog`:

```

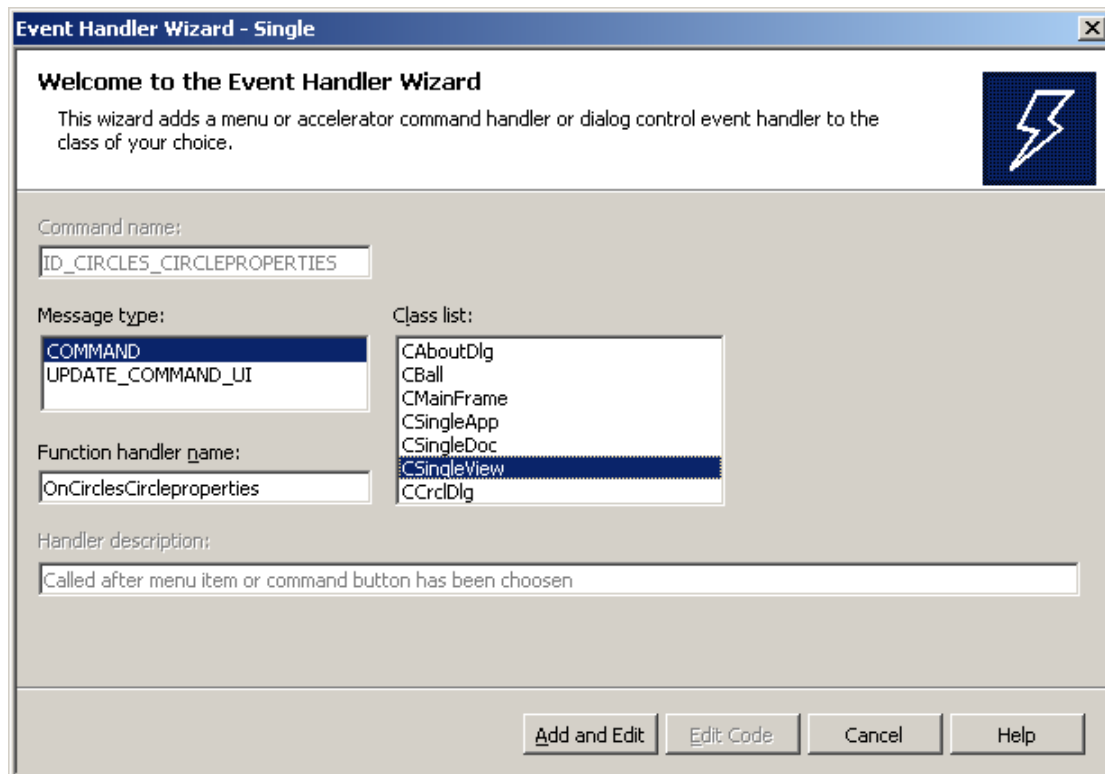
void CCrclDlg::CloseDialog(void)
{
    CComboBox *pCB = (CComboBox*) GetDlgItem(IDC_COMBO1);
    pCB->ResetContent();
    CircleNo = 0;

    DestroyWindow();
    ((CSingleView*)m_pParentWnd)->DlgState = 0;
}

```

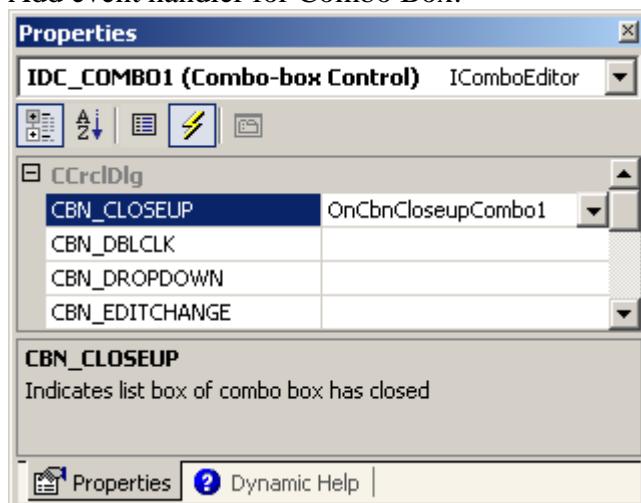
Override `CCrclDlg::OnOK()` and `CCrclDlg::OnCancel()`, call `CloseDialog()` there, and delete `CDialog::OnOK()`, `CDialog::OnCancel()` (since they call `EndDialog`).

Menu redactor – add menu “Circle – Circle Properties” and its handler:



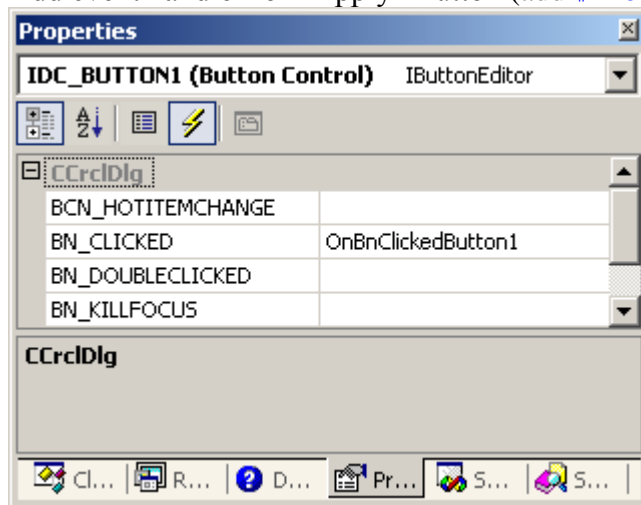
```
void CSingleView::OnCirclesCircleproperties ()
{
    // TODO: Add your command handler code here
    if (!DlgState)
    {
        Dlg.Create(Dlg.IDD, this);
        DlgState = 1;
    }
}
```

Add event handler for Combo Box:



```
void CCrclDlg::OnCbnCloseupCombo1 ()
{
    // TODO: Add your control notification handler code here
    CComboBox *pCB = (CComboBox*) GetDlgItem(IDC_COMBO1);
    CircleNo = pCB->GetCurSel();
    SetDlgData();
}
```

Add event handler for “Apply” Button (add `#include <math.h>` to `Crc1Dlg.cpp`):



```
void CCrc1Dlg::OnBnClickedButton1()
{
    // TODO: Add your control notification handler code here
    UpdateData(TRUE);

    float DeltaLength = sqrt(DeltaX * DeltaX +
                             DeltaY * DeltaY);

    CSingleDoc *pDoc = ((CSingleView*)m_pParentWnd)->GetDocument();
    CBall *pBall = (CBall*) pDoc -> arrayBall[CircleNo];

    pBall->DeltaX = DeltaX / DeltaLength;
    pBall->DeltaY = DeltaY / DeltaLength;
    pBall->Radius = Radius;
    pBall->Velocity = Velocity;

    m_pParentWnd->KillTimer(CircleNo+1);
    m_pParentWnd->SetTimer(CircleNo+1, (int) 1000 / Velocity, NULL);

    pDoc->SetModifiedFlag();
}
```

Modify `CSingleView::OnLButtonDown`:(add before `pDoc->SetModifiedFlag`)

```
if (DlgState)
{
    CComboBox *pCB = (CComboBox*) Dlg.GetDlgItem(IDC_COMBO1);
    CString str;
    str.Format(_T("circle %d"), pDoc->arrayBall.GetSize());
    pCB->AddString(str);
}
```

and `CSingleView::OnRButtonDown`:(add before `pDoc->SetModifiedFlag`)

```
if (DlgState)
{
    if (Dlg.CircleNo == pDoc->arrayBall.GetSize())
    {
        Dlg.CircleNo = pDoc->arrayBall.GetSize() - 1;
        Dlg.SetDlgData();
    }
}
```

```
        CComboBox *pCB = (CComboBox*) Dlg.GetDlgItem(IDC_COMBO1);  
        pCB->DeleteString(pDoc->arrayBall.GetSize());  
    }
```

If you do not see the dialog when running the application, check the Visibility property of the dialog in the resource editor (have to be TRUE). Alternatively, you can call "ShowWindow" after Create(...).