Supplementary Note for Module 29
-ActiveX control test program-

Program examples compiled using Visual C++ 6.0 compiler on Windows XP Pro machine with Service Pack 2. The following are the steps to build a very simple dialog program to test the **myatldicesvr**, an ActiveX control. Make sure you already build without error our ActiveX control program, **myatldicesvr**. **Myatltest** is a dialog based MFC application.

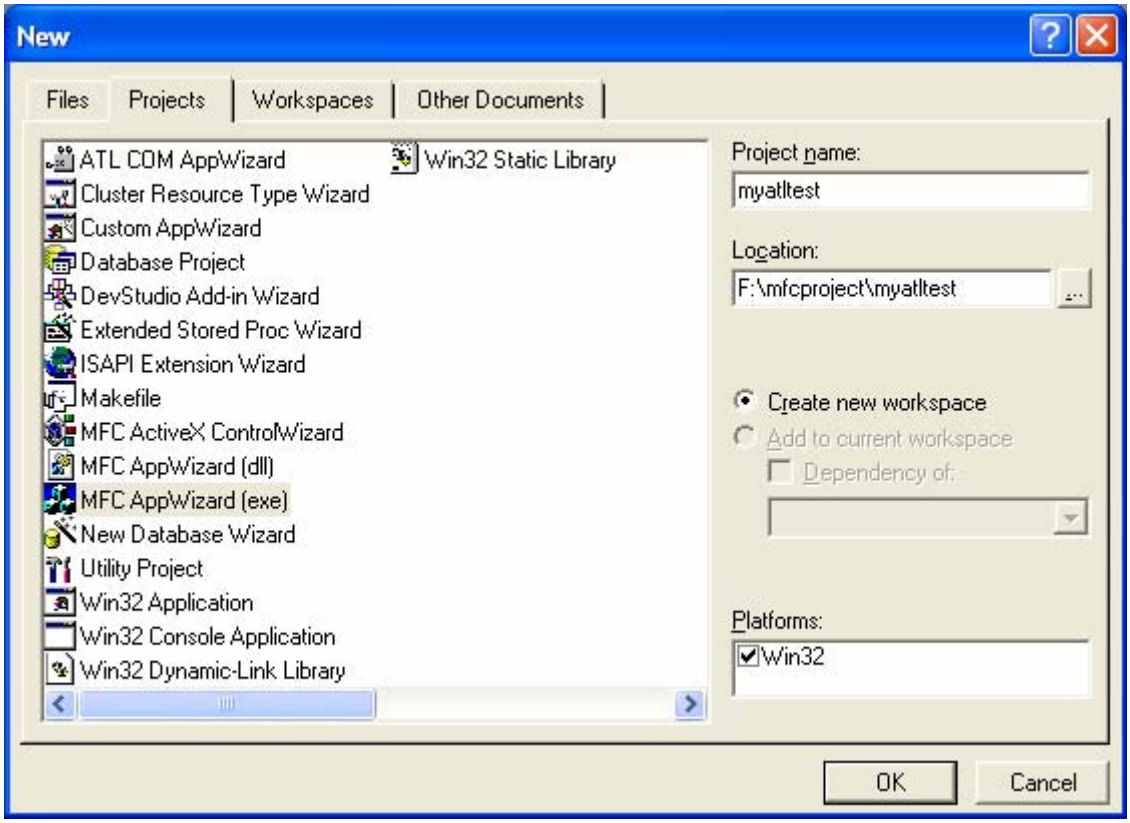

Figure 1: MYATLTEST – Visual C++ new project dialog.

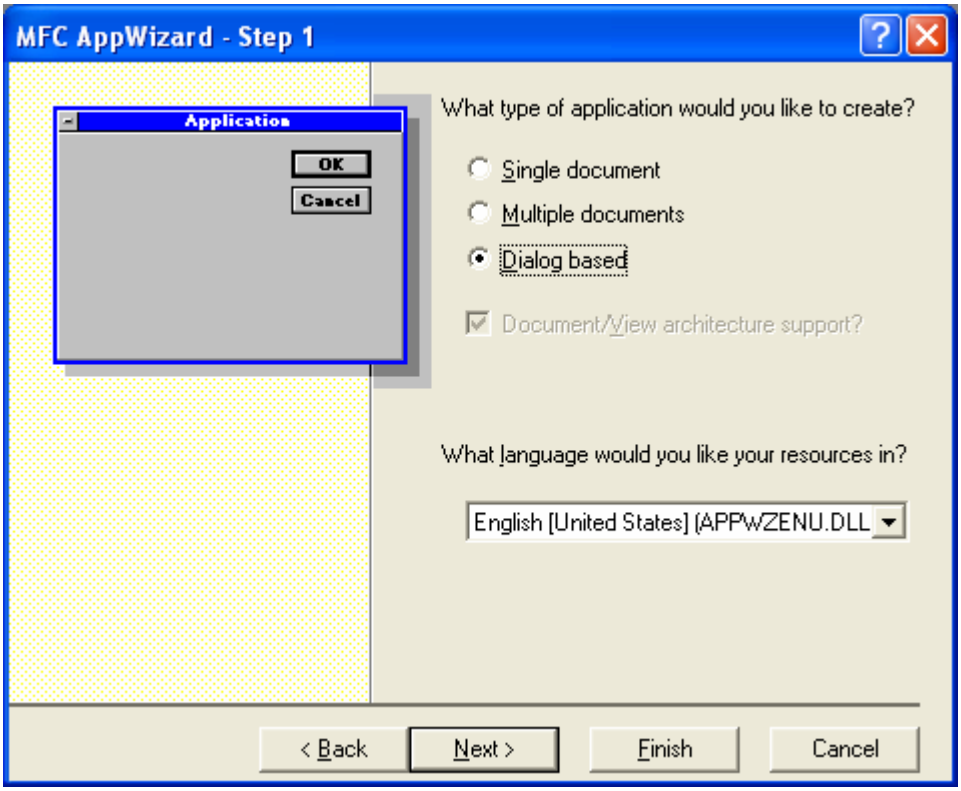

Figure 2: MYATLTEST – AppWizard step 1 of 4.

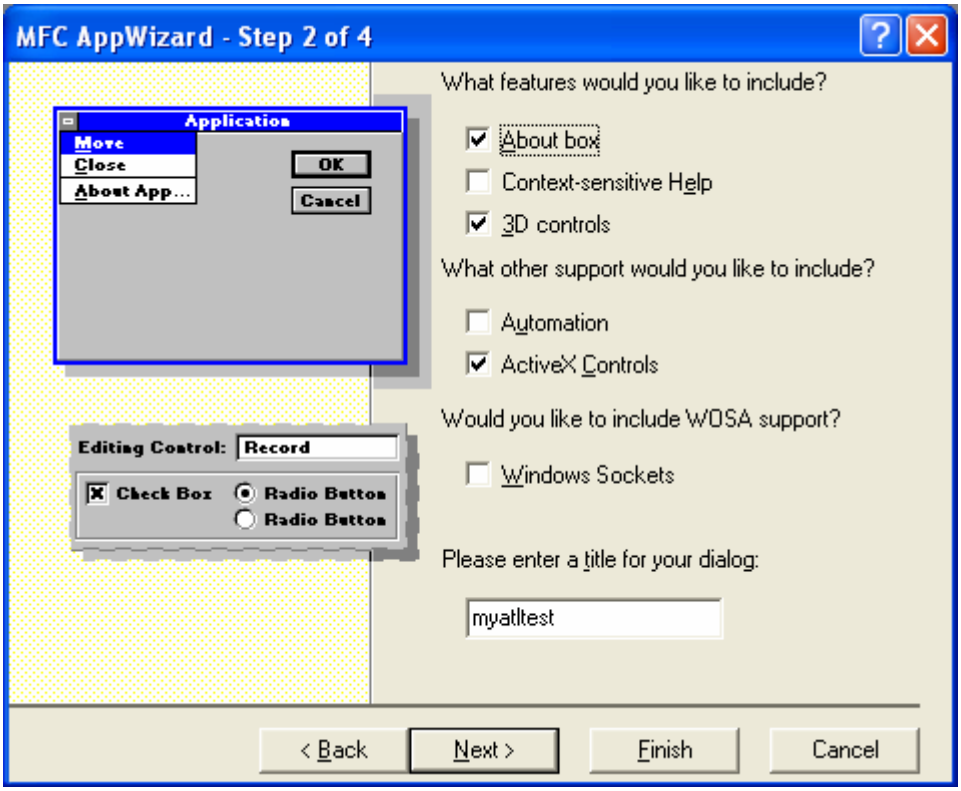

Figure 3: MYATLTEST – AppWizard step 2 of 4.

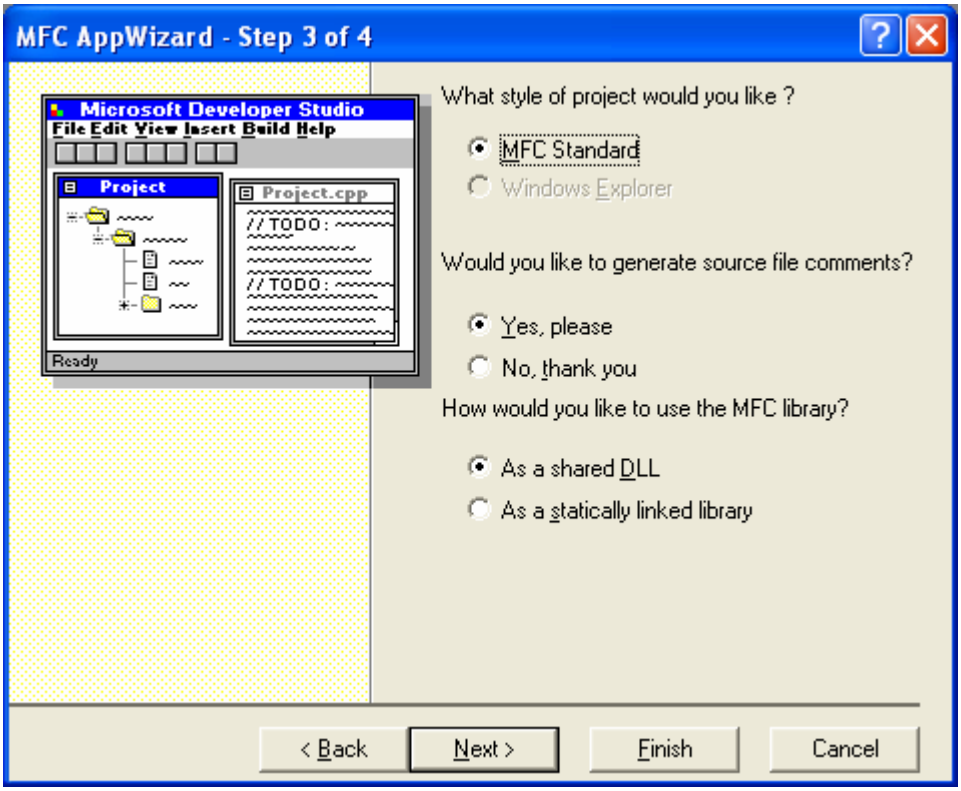

Figure 4: MYATLTEST – AppWizard step 3 of 4.

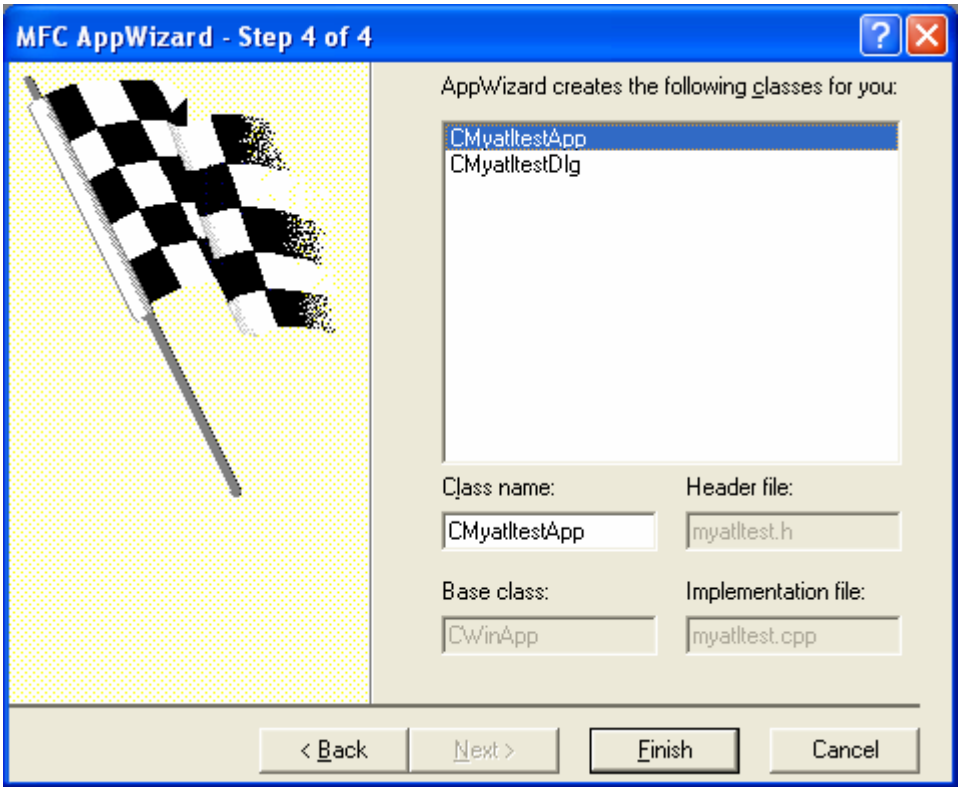

Figure 5: MYATLTEST – AppWizard step 4 of 4.

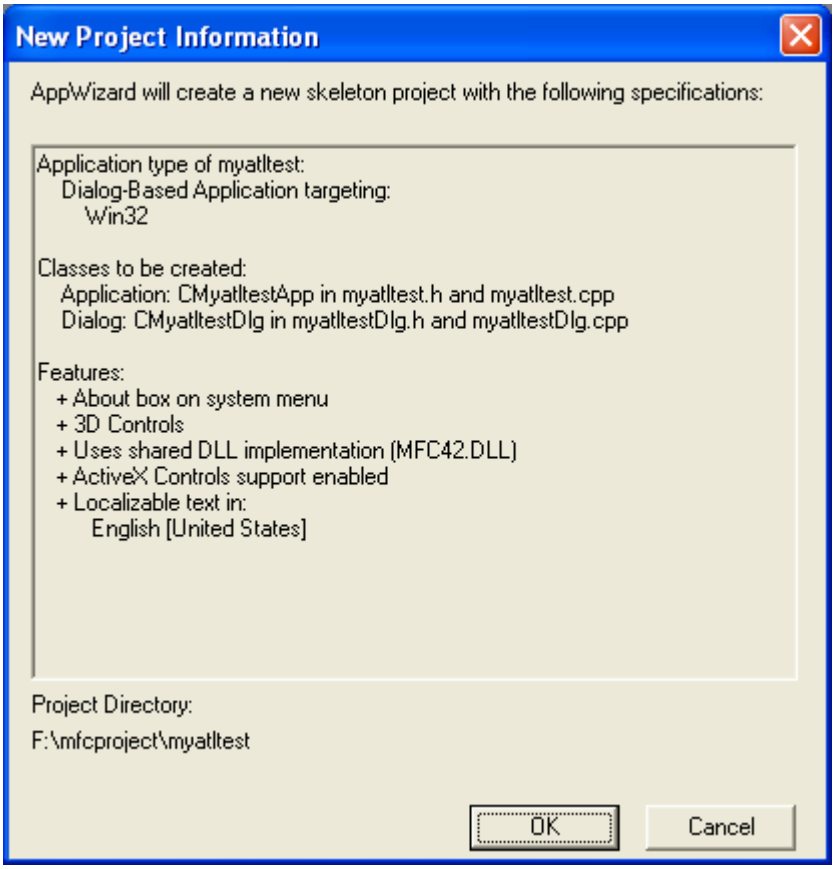

Figure 6: MYATLTEST project summary.

Next, insert the control that we already built and registered. Select the **Project Add To Project** and select **Components and Controls**.

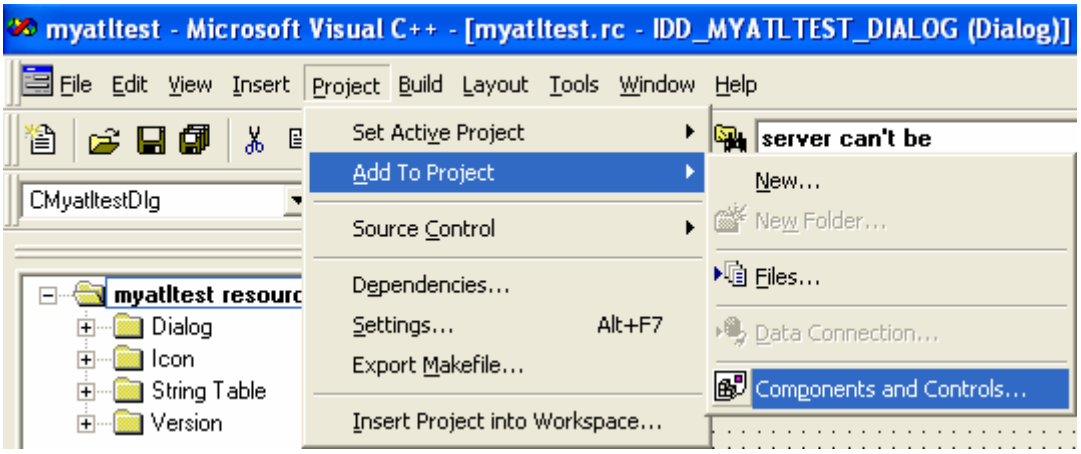

Figure 7: Adding new ATL control to project.

Browse our control, **myatldicesvr** under the **Registered ActiveX Controls** folder.

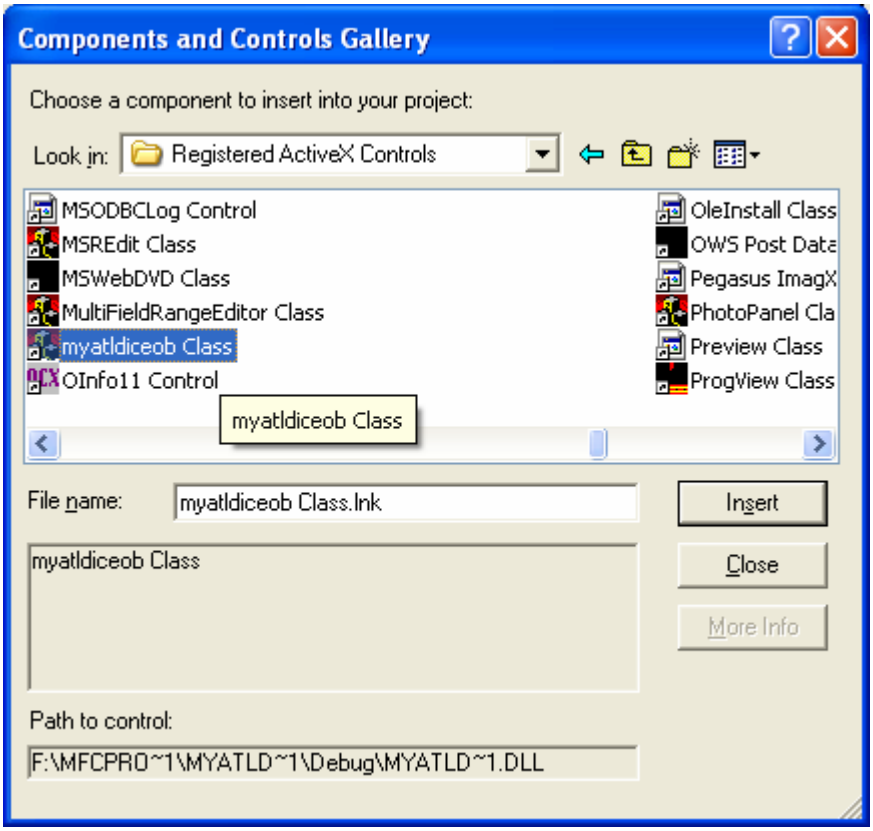

Figure 8: Selecting **myatldiceob** ATL control.

Select the control and click the **Insert** button. Just click the **OK** button for the following prompt.

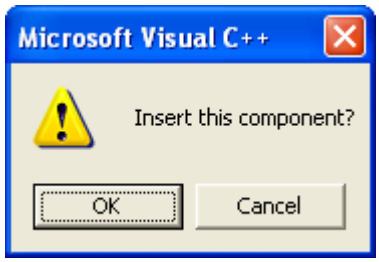

Figure 9: New component insert confirmation dialog.

Click the **OK** button for the following class confirmation prompt. Then, close the **Components and Controls Gallery** dialog.

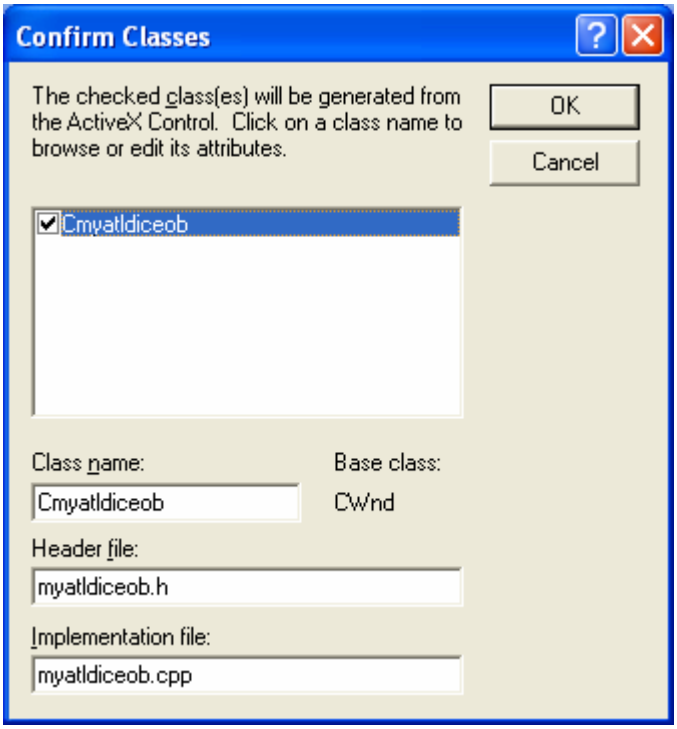

Figure 10: Class addition conformation dialog.

You can see the control at the bottom of the **controls bar**.

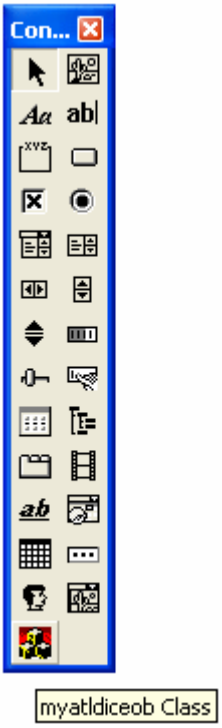

Figure 11: myatldiceob ATL control in Visual C++'s control bar.

Drag and drop the control to the dialog as shown below.

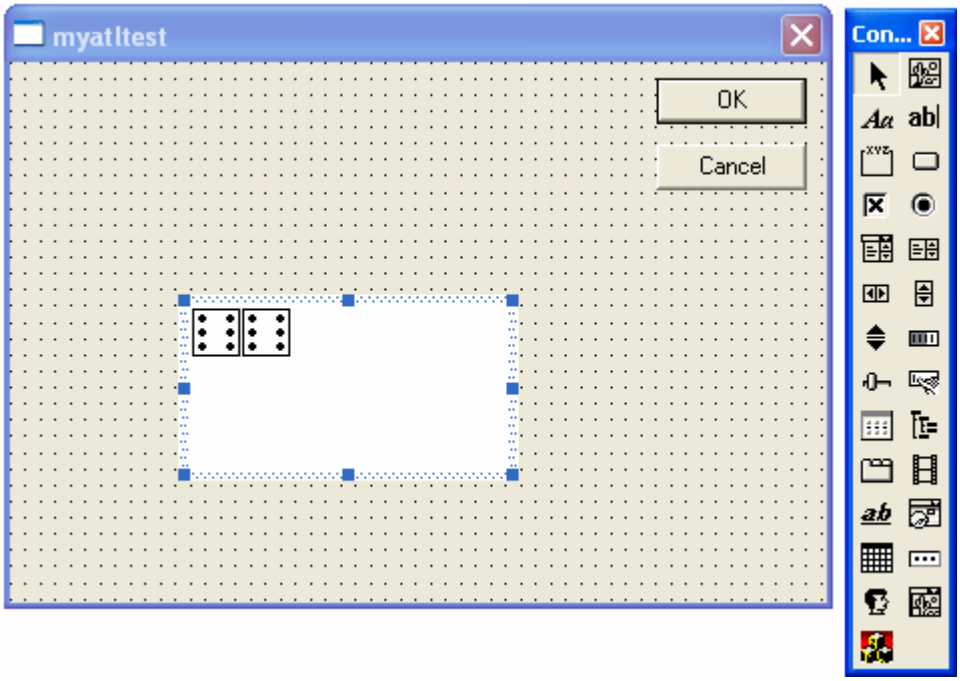

Figure 12: Inserting **myatldiceob** control to dialog.

Build and run **myatltest**. Then double click the dice image (or the white area). Can you see the action?

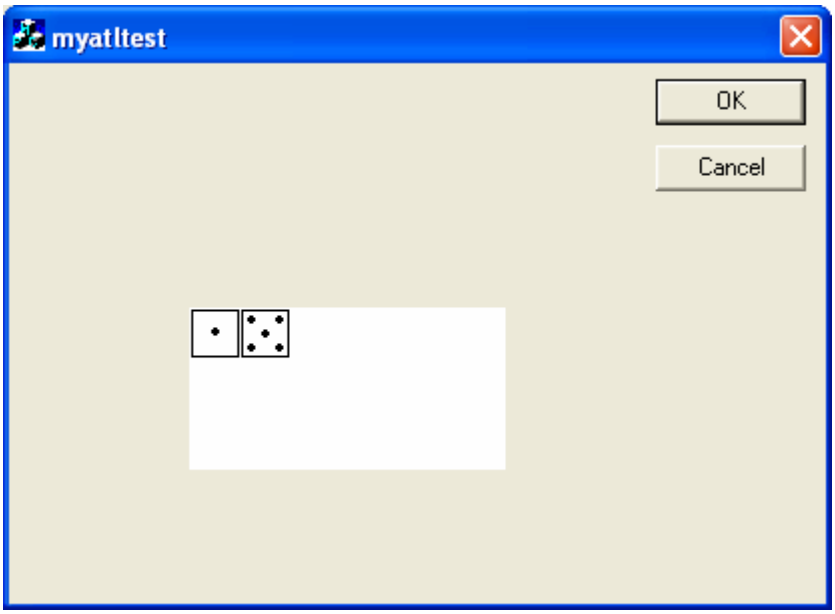

Figure 13: **myatltest** with **myatldiceob** ATL control in action.

Next, let see the **property page** that we have created. Select the dice control and right click. Then select the **Properties** context menu.

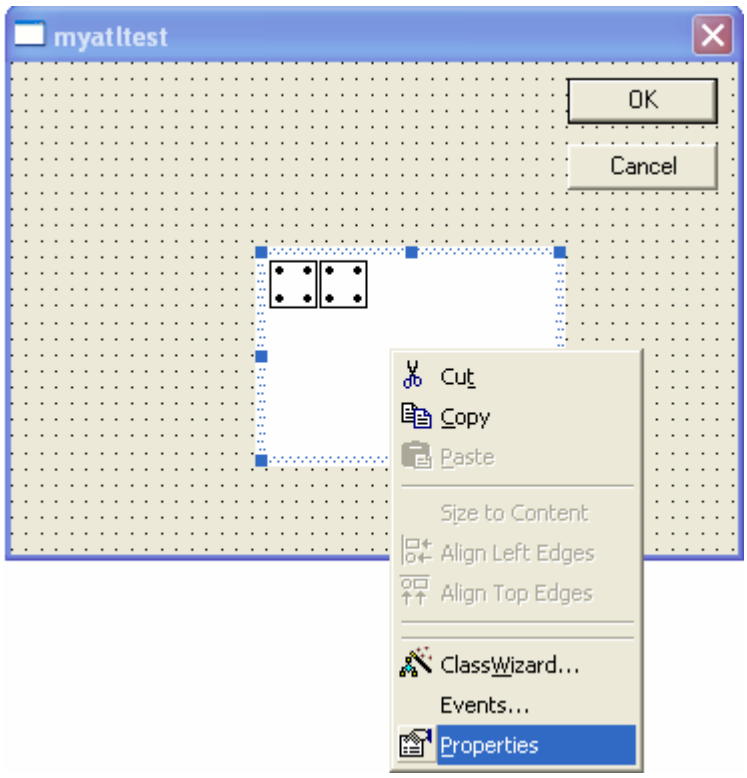

Figure 14: Invoking **myatldiceob** property page.

Click the **All** tab. Here you can see the properties that we have exposed to users.

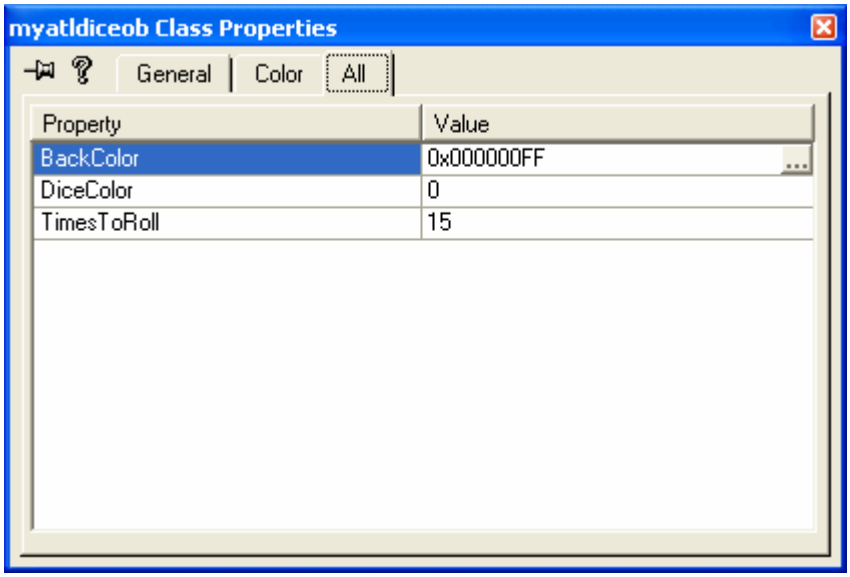

Figure 15: **myatldiceob** ATL control's property page.

Select the **BackColor** in the property column and then click the **. . .** to change the background color. Next, change the **DiceColor** to **1** (should be blue) as shown below.

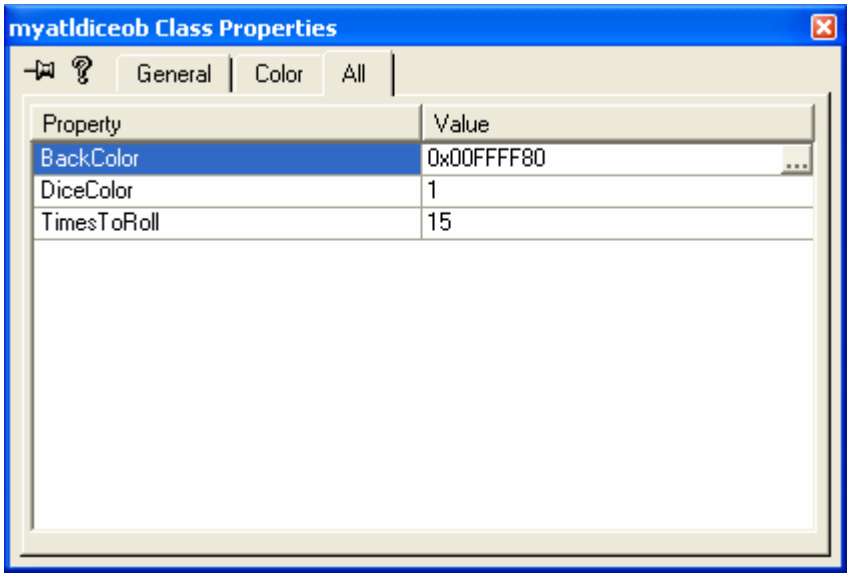

Figure 16: Changing **myatldiceob** property.

The dice changed to blue, but this is in design mode.

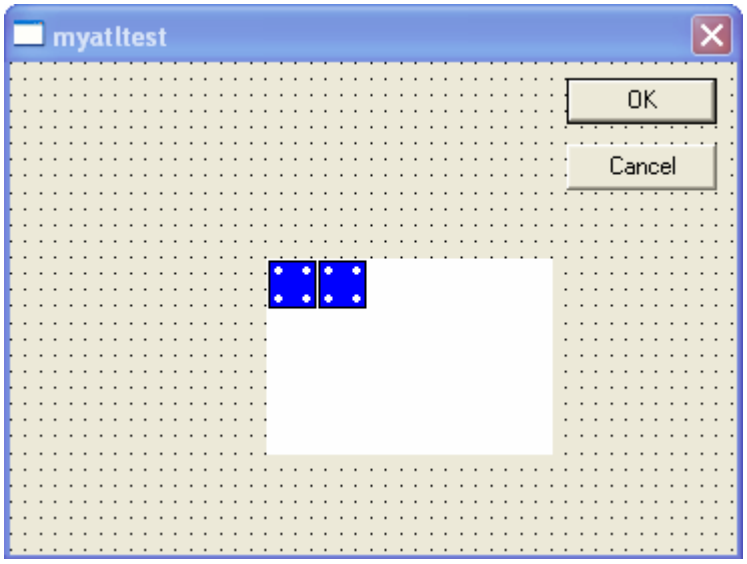

Figure 17: Changing the dice to blue color.

Build and run **myatltest**. Double click the dice or the white area.

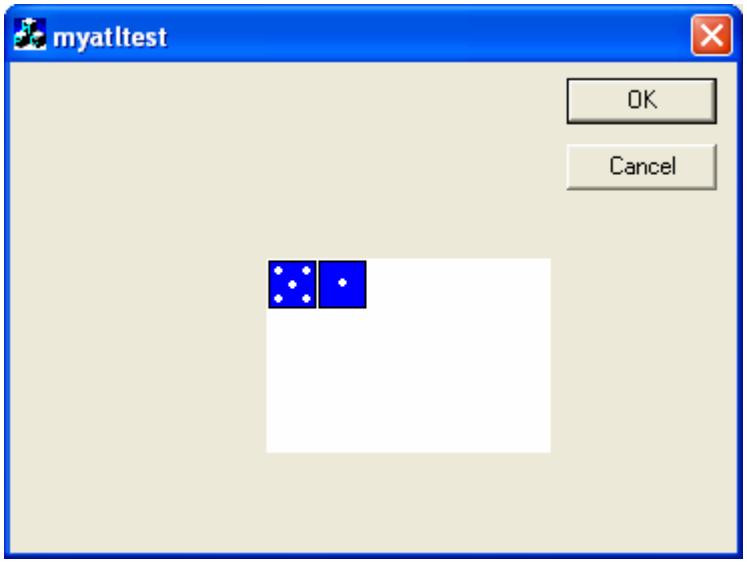

Figure 18: Blue dice in action.

Finally we change the dice color to red and the times to roll to 30. Build and run.

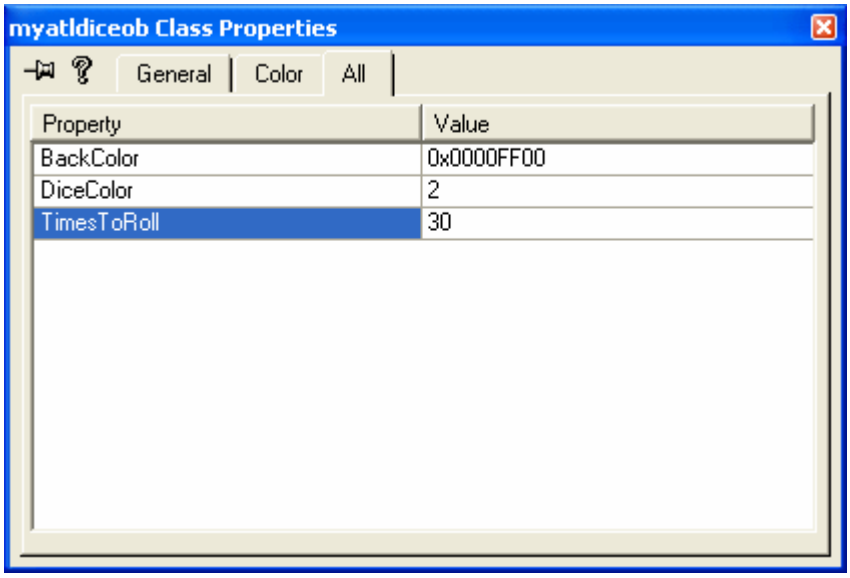

Figure 19: Changing **myatldiceob** properties.

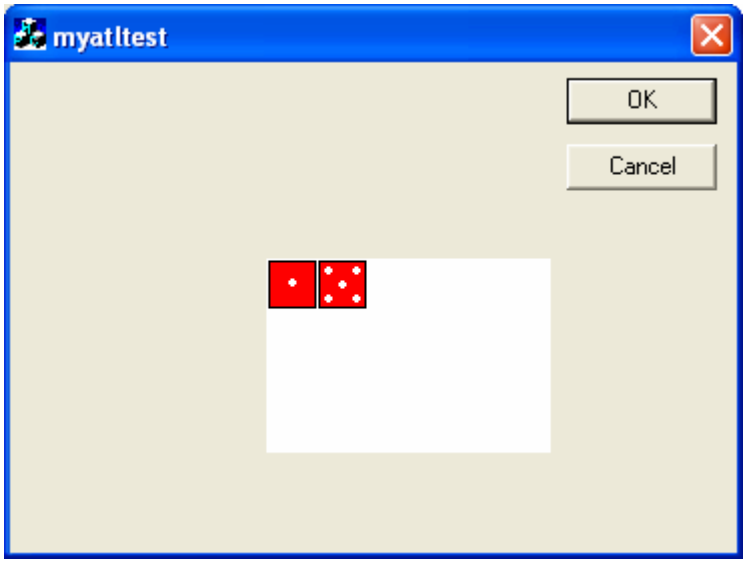

Figure 20: Red dice in action.

----------End----------