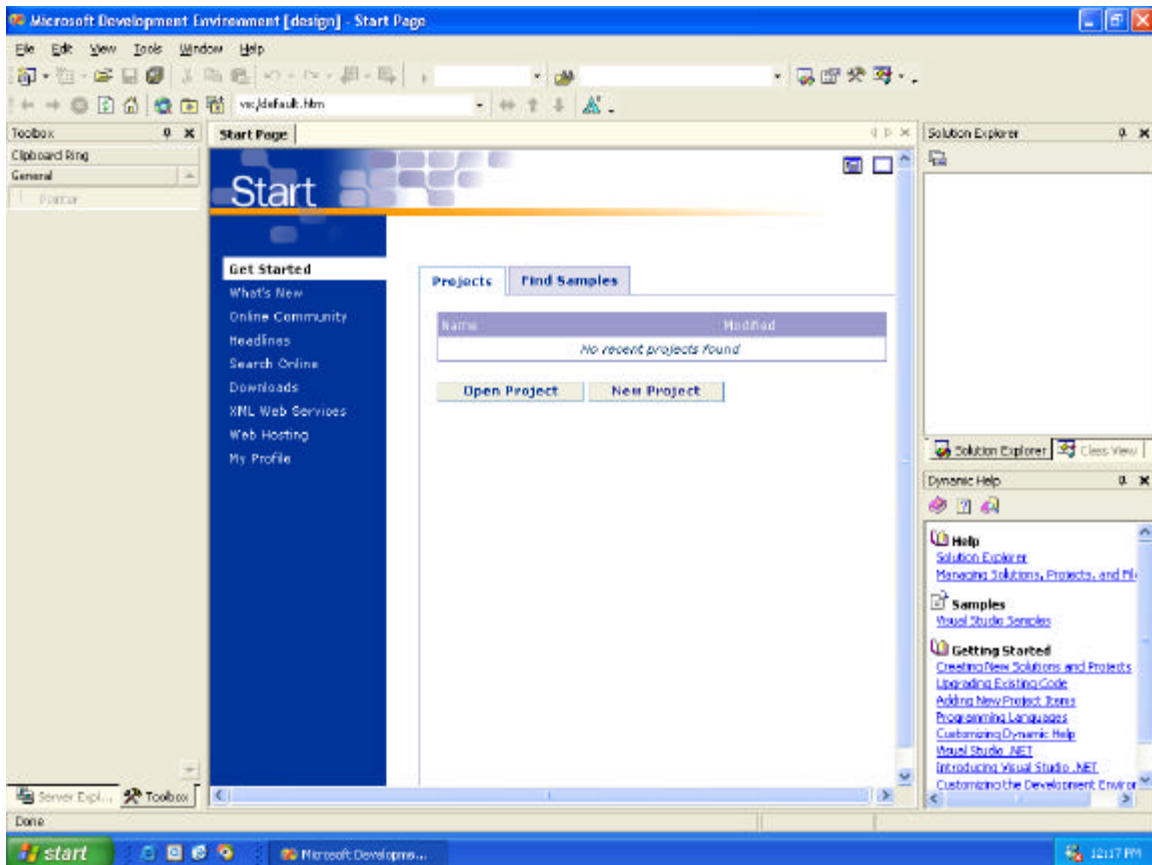


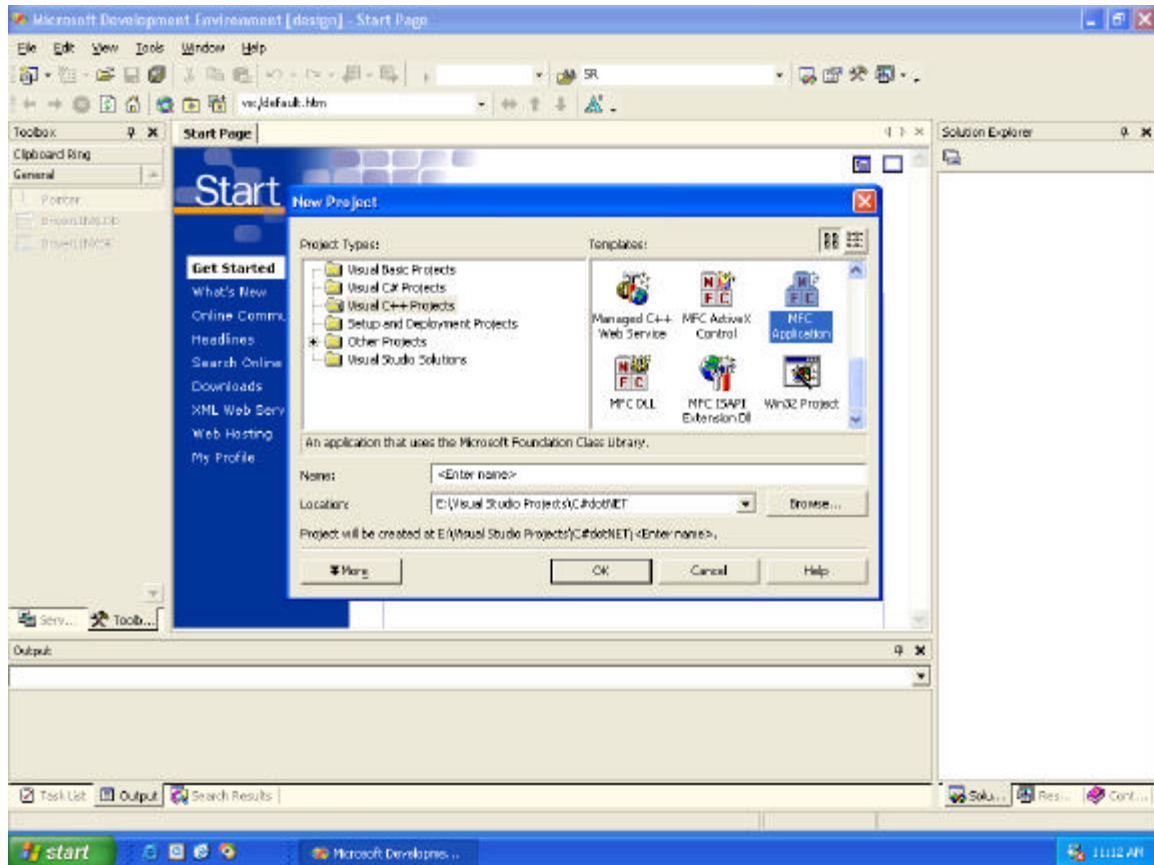
## How to use Visual C++.NET with DriverLINX

Assumptions: You've installed Microsoft Visual Studio.NET (VS.NET) and DriverLINX for a Keithley plug-in board product.

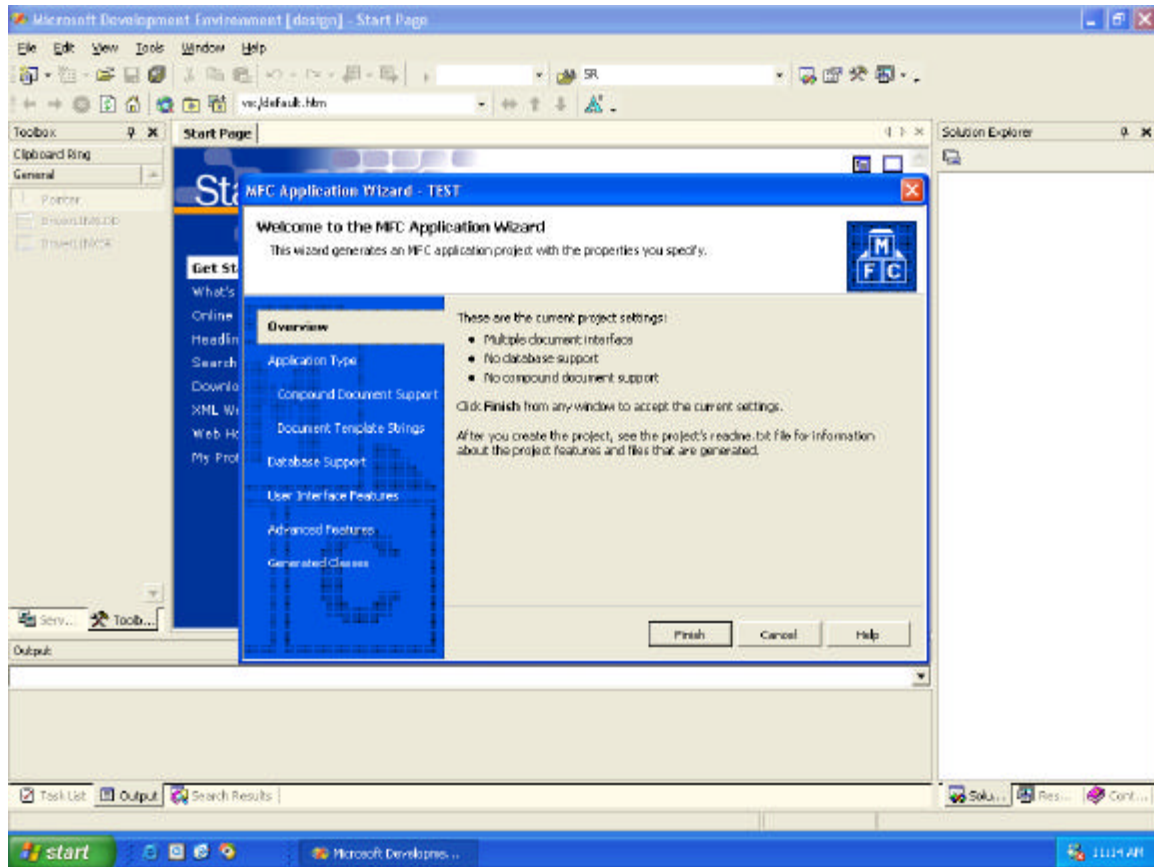
When you launch Microsoft VS.NET, you'll get the screen below after it loads.



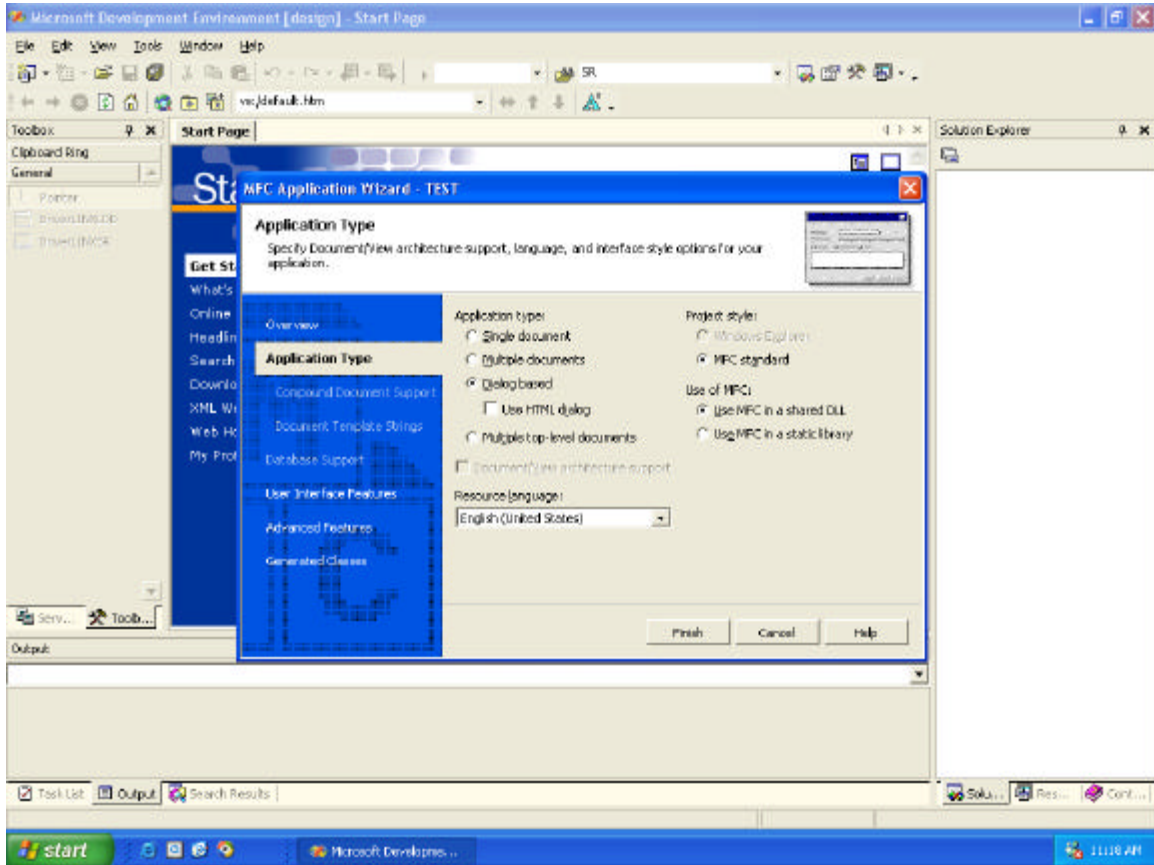
Click New Project. Choose the appropriate project for the language desired. In this case, we'll choose the Visual C++ Projects and MFC Application under Templates. Name the program and click OK. I have used the name TEST for this project.



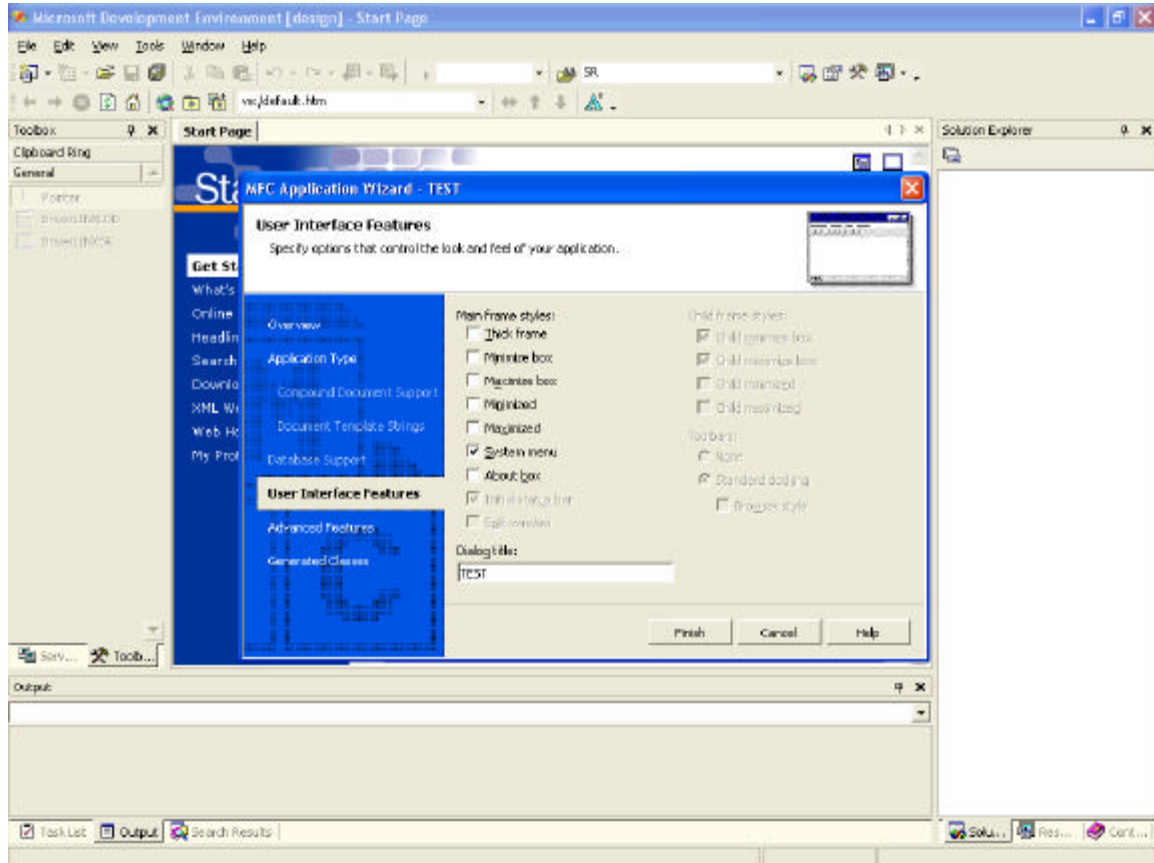
You should see the screen below.



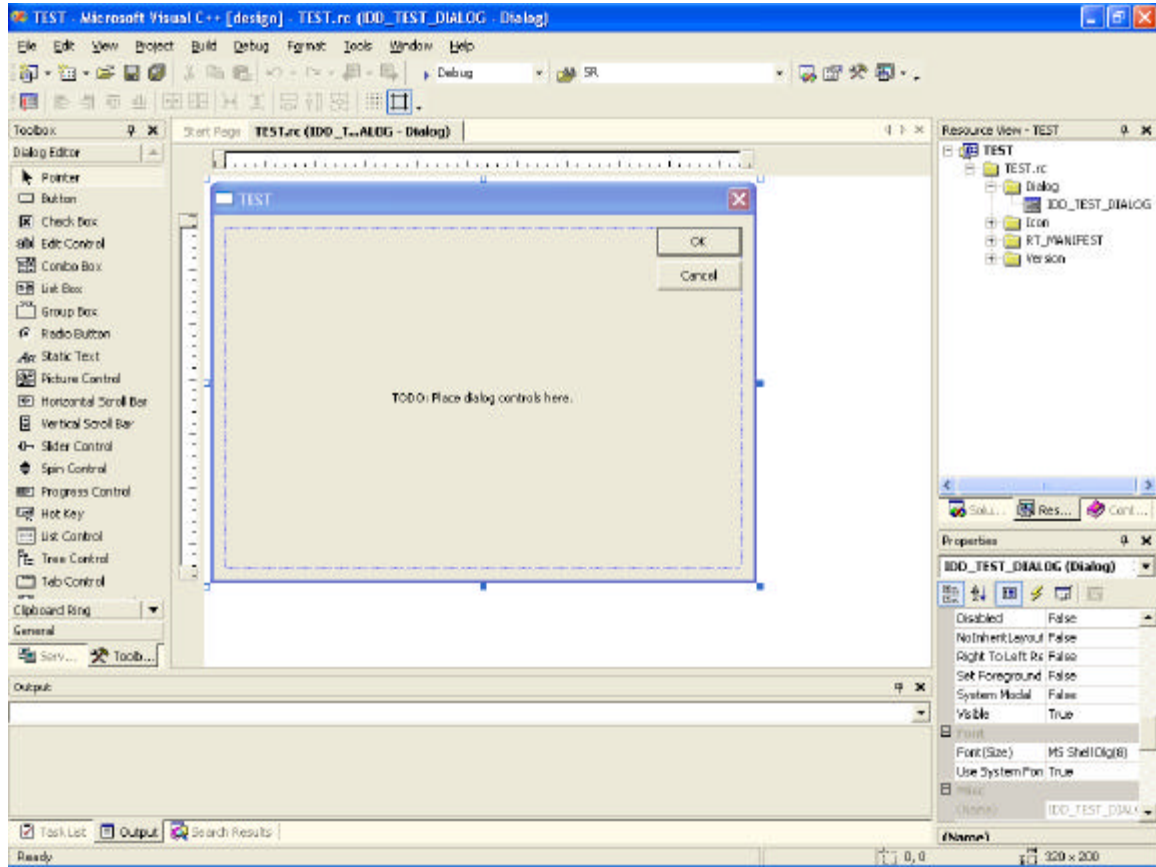
Select **Application Type** and click **Dialog based**.



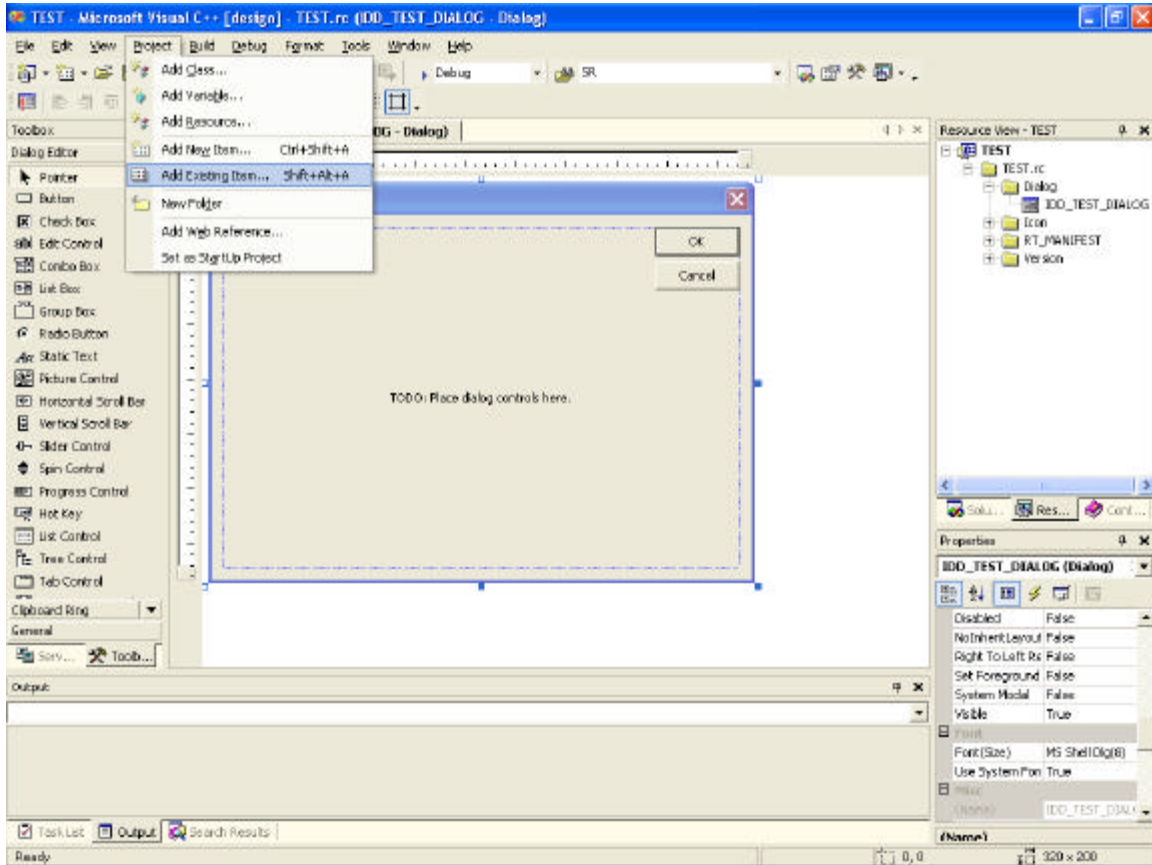
Select **User Interface Features**, uncheck **About box** and click **Finish**.



You will see the following on your screen:



Click **Project**. Select **Add Existing Item**.



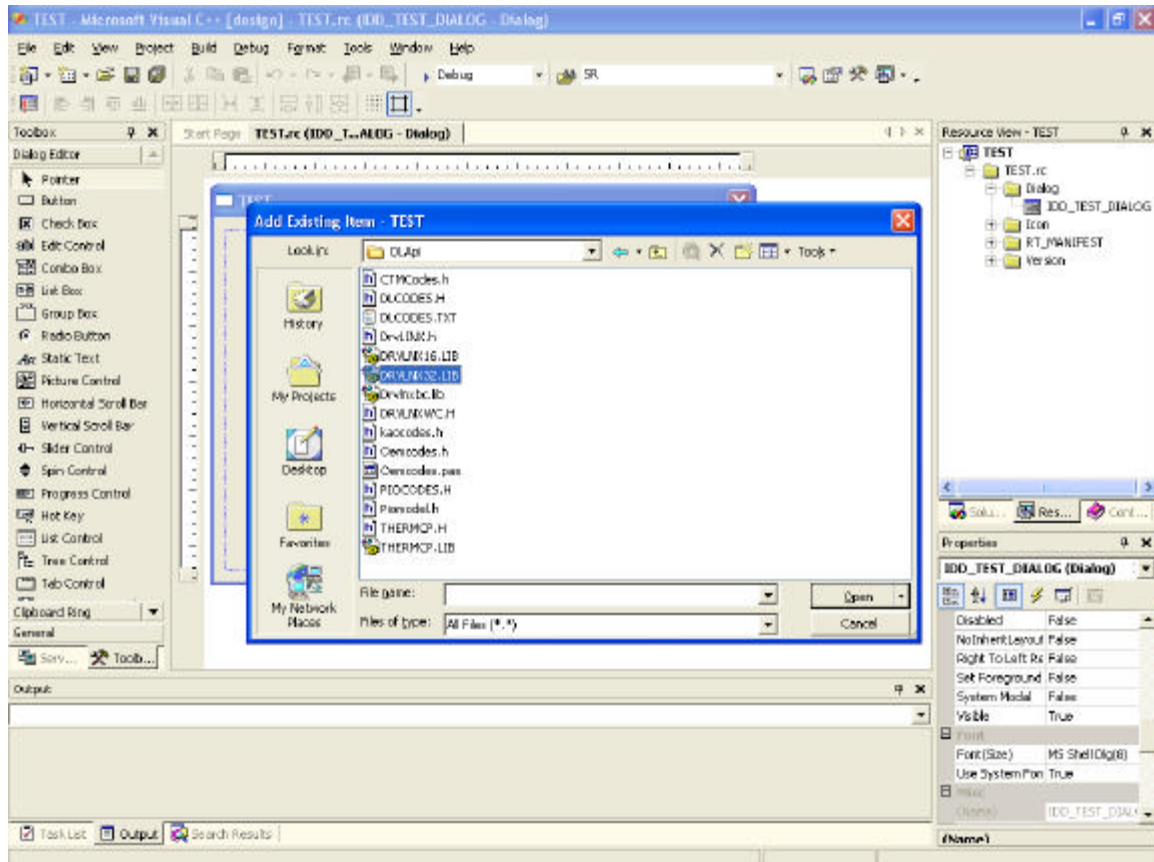


Go to the following location:

Root directory: \DrvLINX4\DLApi\DRVLINX32.LIB and click open.

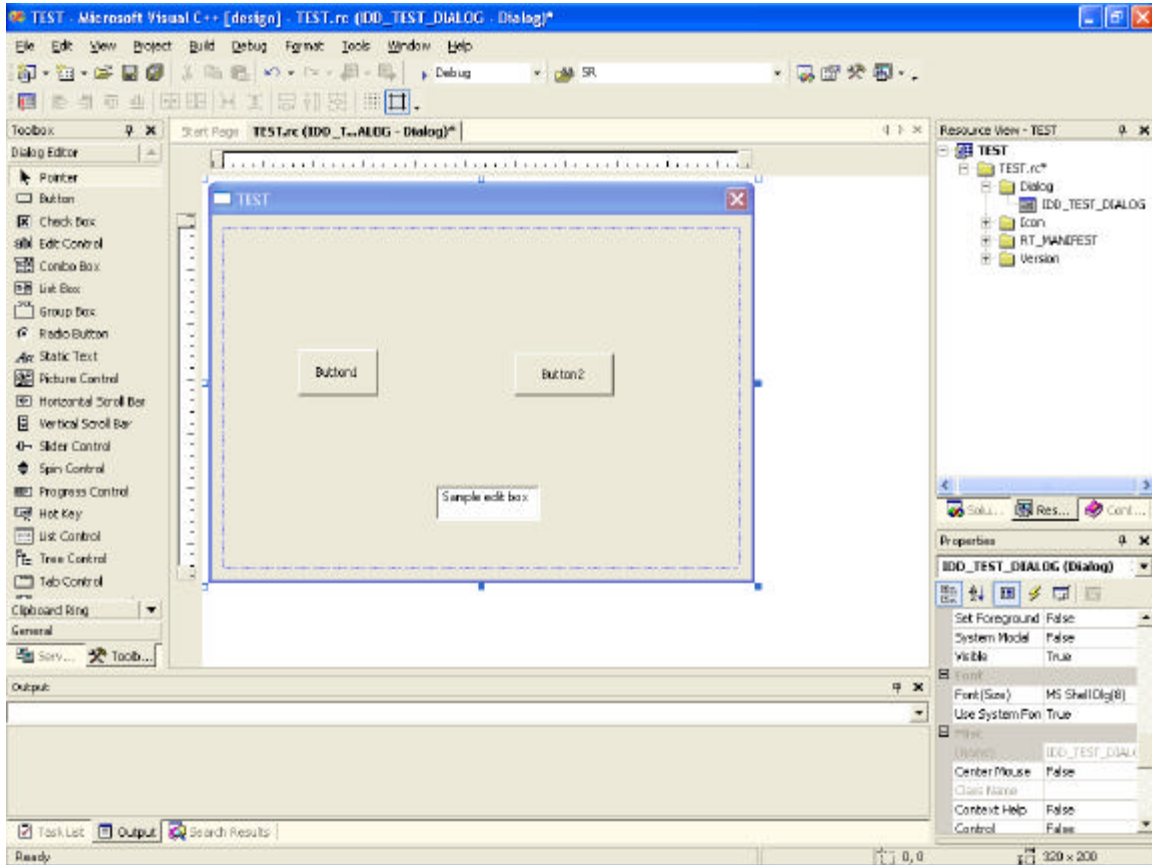
(The root directory will be the directory where DriverLINX was installed)

If you fail to add this library file to the project, then you will get unresolved externals when you compile/link the project.

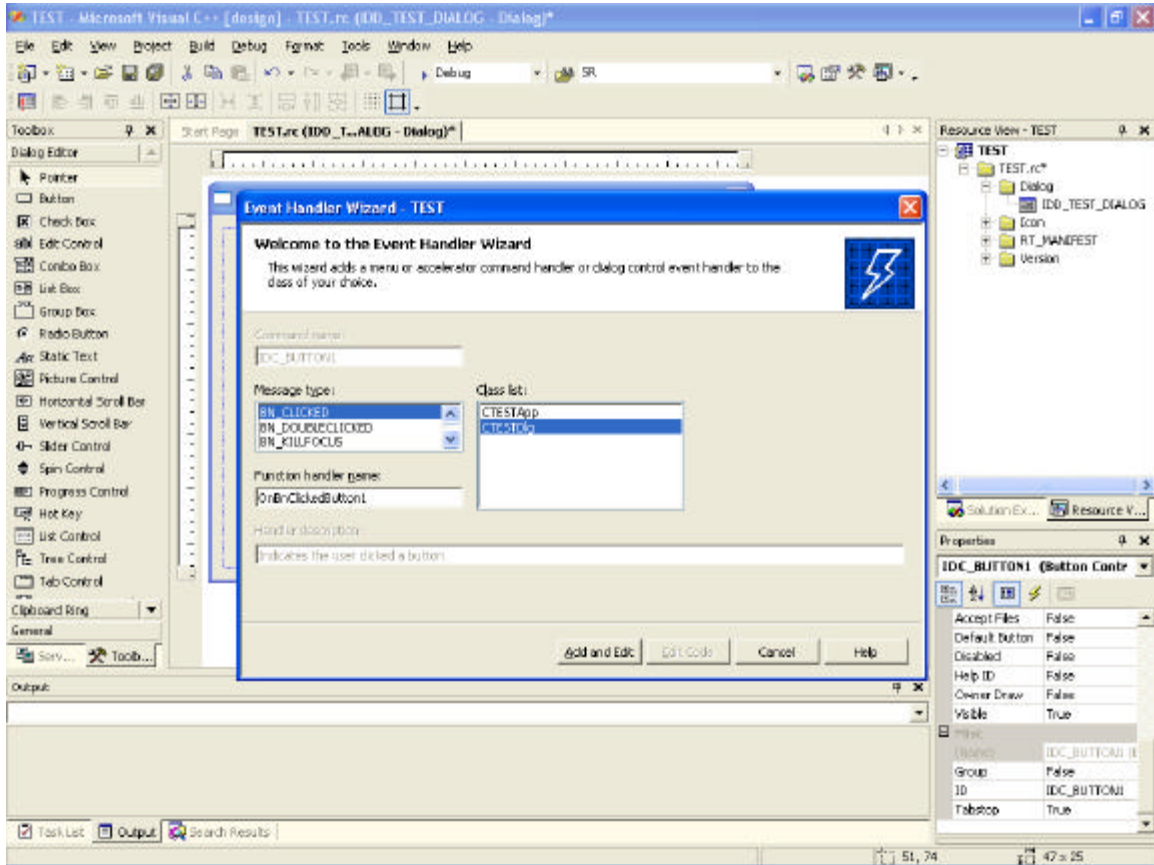




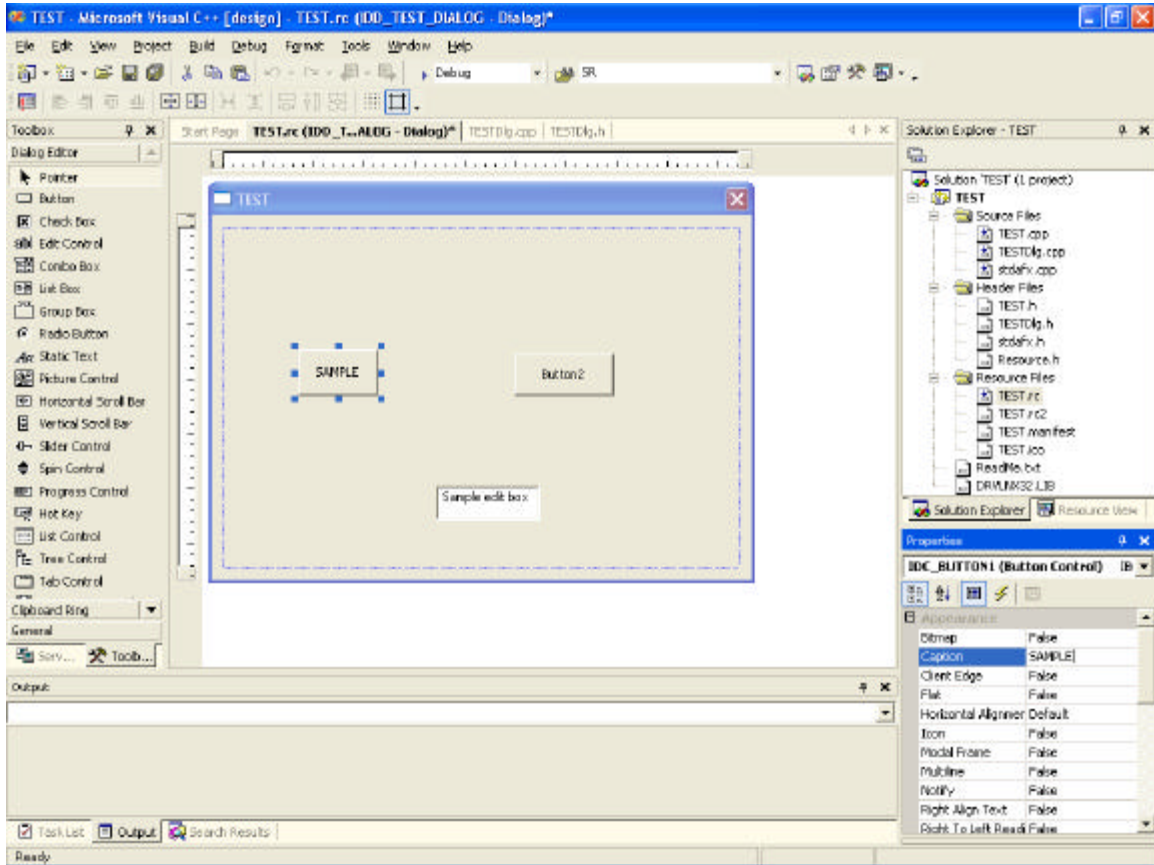
Now you can add buttons, text boxes, etc. in the same manner that you may already be familiar with from earlier versions of Visual C++. Let's finish cleaning up the dialog before we customize it: highlight the OK button and choose delete. Do the same for the Cancel button and the TODO label. Now add two buttons and an edit control box.



To add an ID for a Button do the following: Choose button, click the right mouse button and select Add Event Handler. Select: Message type as **BN\_CLICKED**, Class list as **CTESTDlg** and rename Function handler name as desired. Click **Add and Edit**. Repeat for second button.



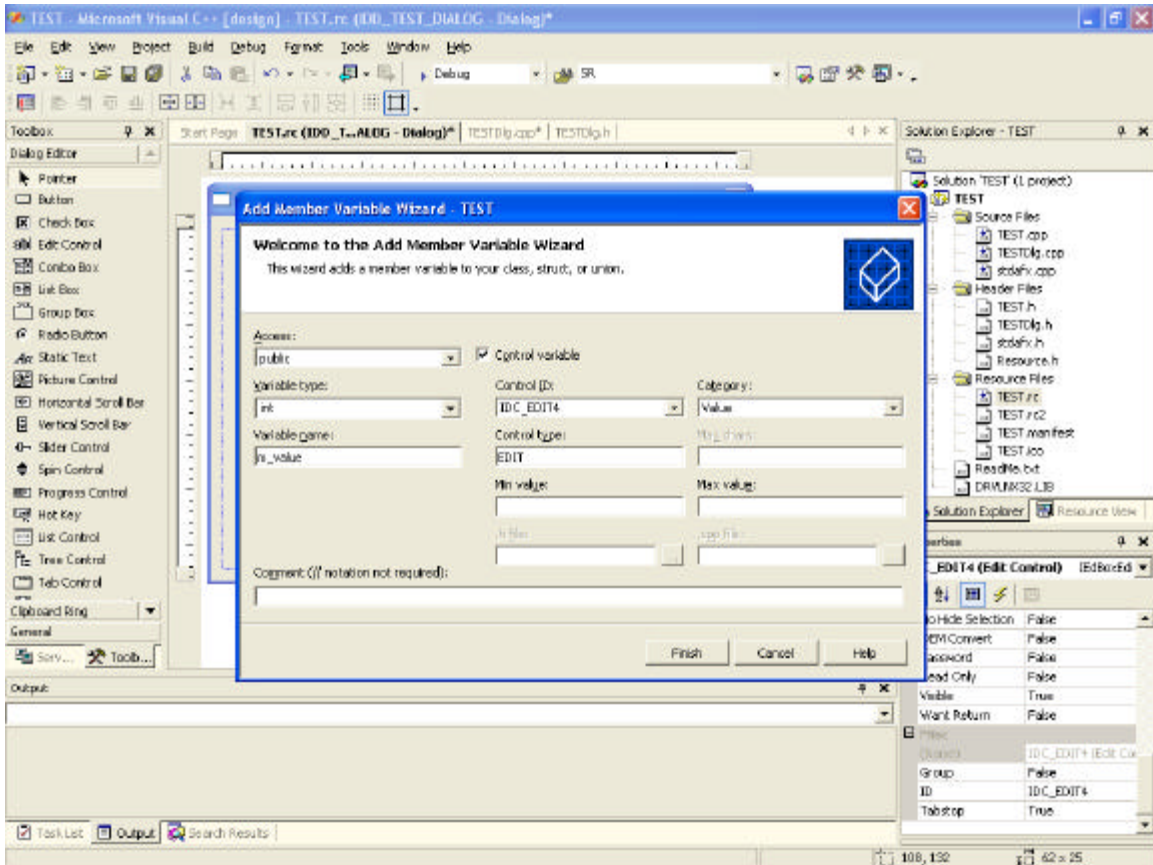
To change the caption/label of a button, single click on button to be modified. Go to program menu bar and select **View** then choose **Properties Window**. Highlight Button1 by Caption and type **SAMPLE** and press Enter. Repeat for Button2 but use **QUIT** instead of **SAMPLE**.



To add a variable to Edit Control box, click on edit box. Click the right mouse button, choose **Add variable** and select the following:

- Category = **Value**
- Variable Type = **int**
- Variable name = **m\_value**

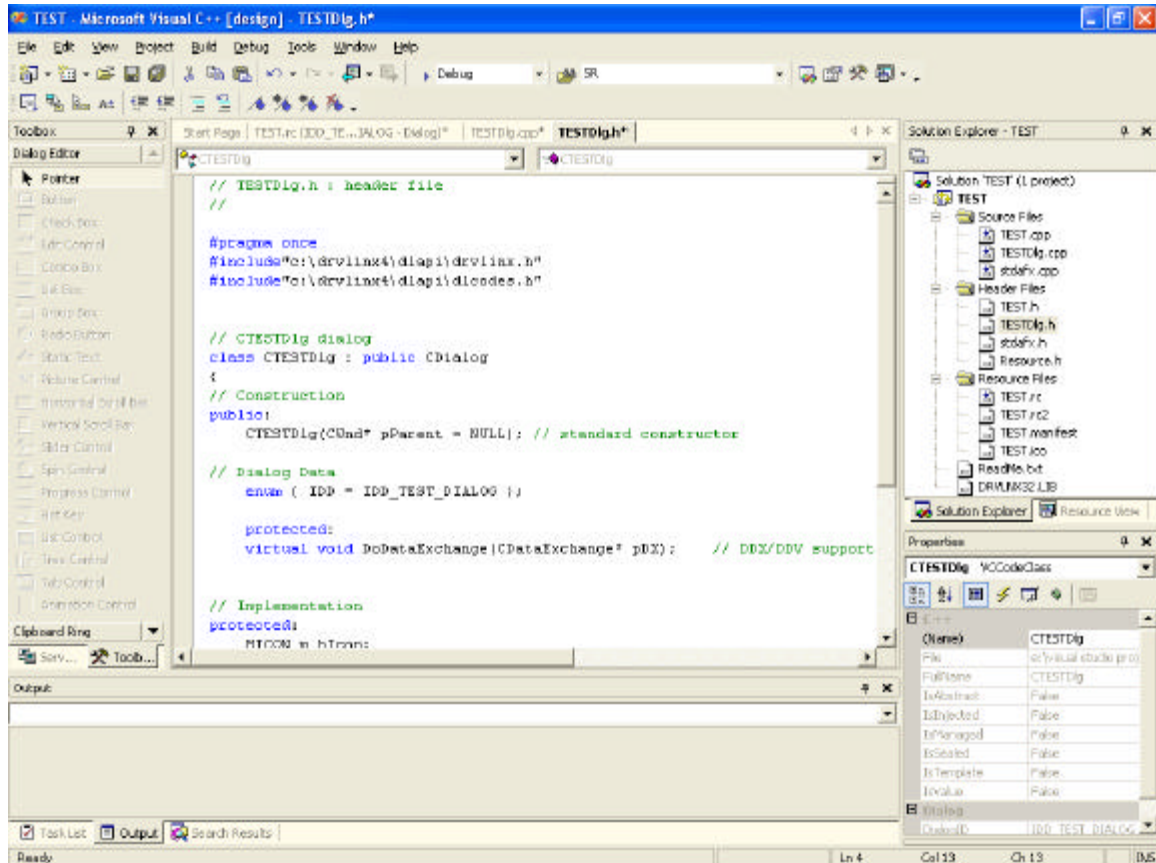
Click Finish.



To complete this test program that will read a digital an input port, the following code must be added:

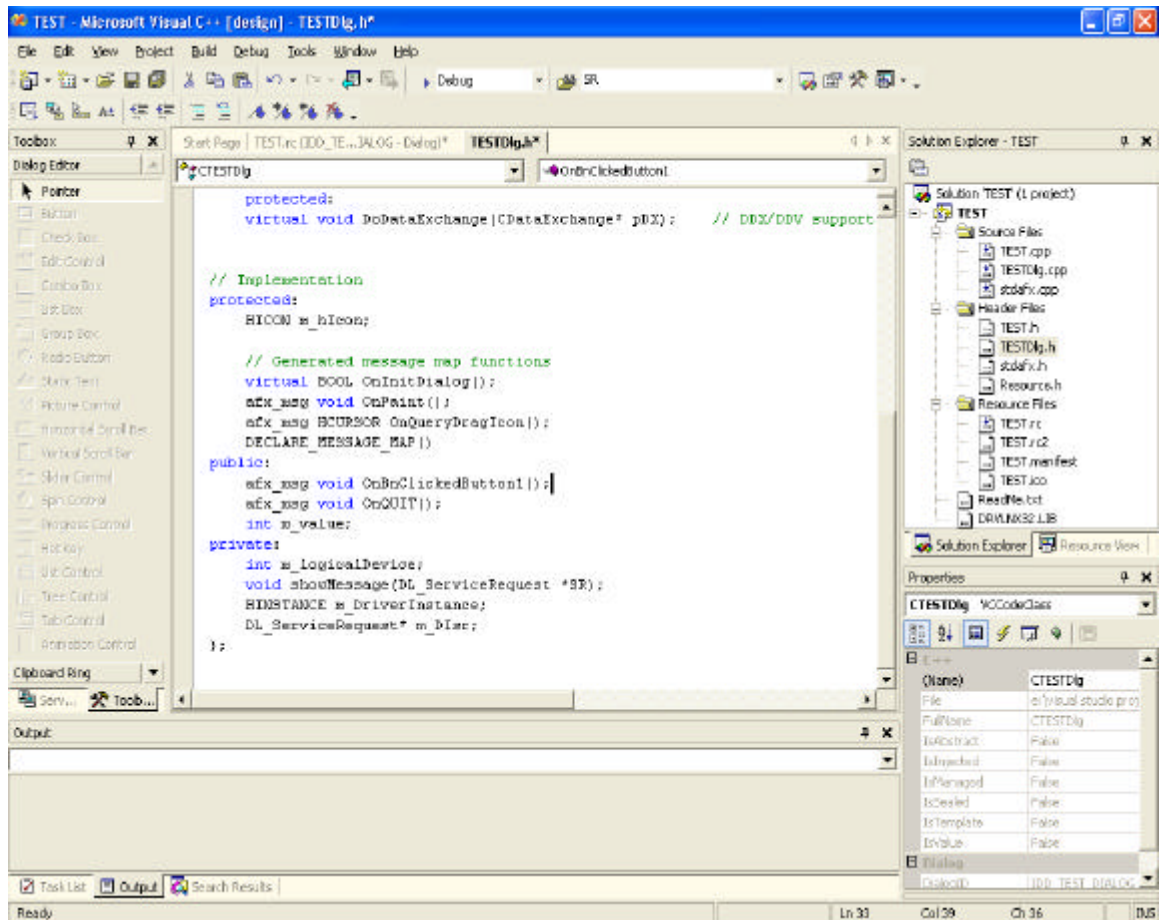
**UNDER TESTDlg.h** : header file at the top:

```
#include "c:\drvlinx4\dlapi\drvlinx.h"  
#include "c:\drvlinx4\dlapi\dlcodes.h"
```



UNDER TESTDlg.h : header file at the bottom:

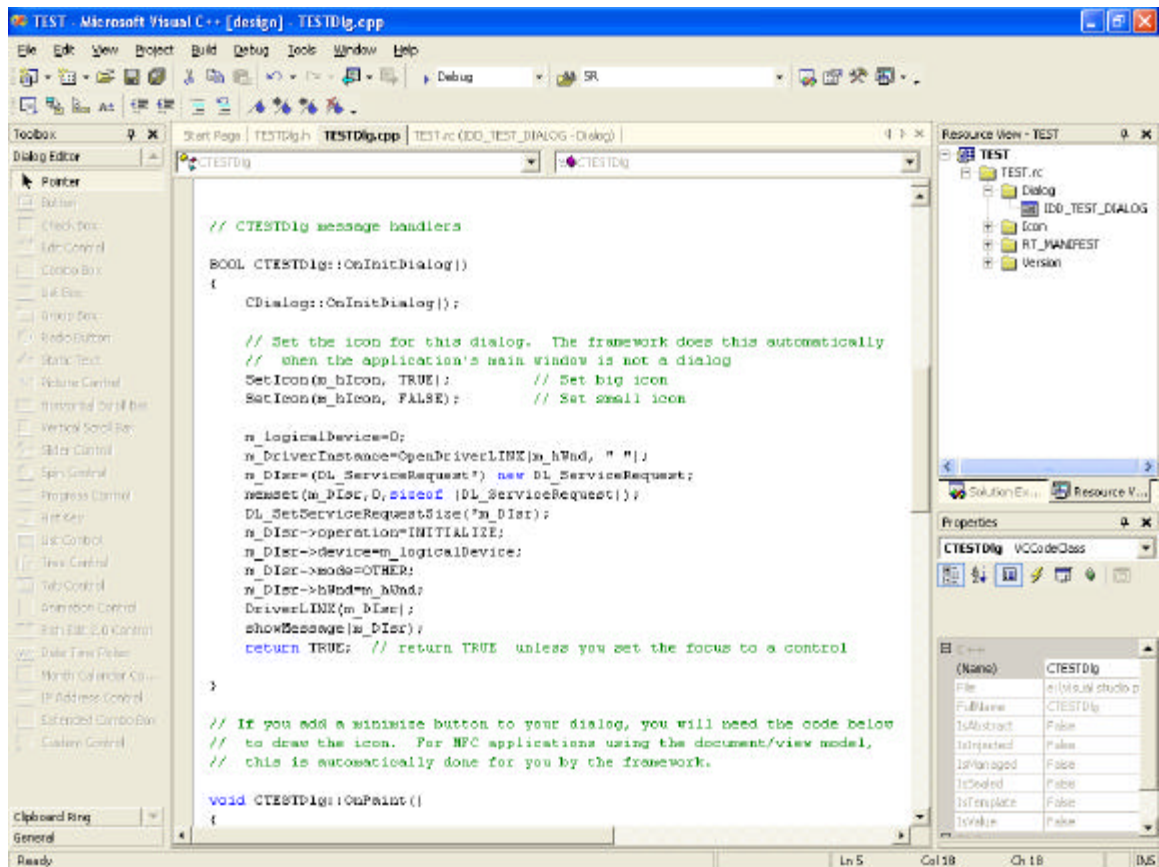
```
int m_inputvalue;  
private:  
int m_logicalDevice;  
void showMessage(DL_ServiceRequest *SR);  
HINSTANCE m_DriverInstance;  
DL_ServiceRequest* m_DISR;
```





```
UNDER CDialog::OnInitDialog();
```

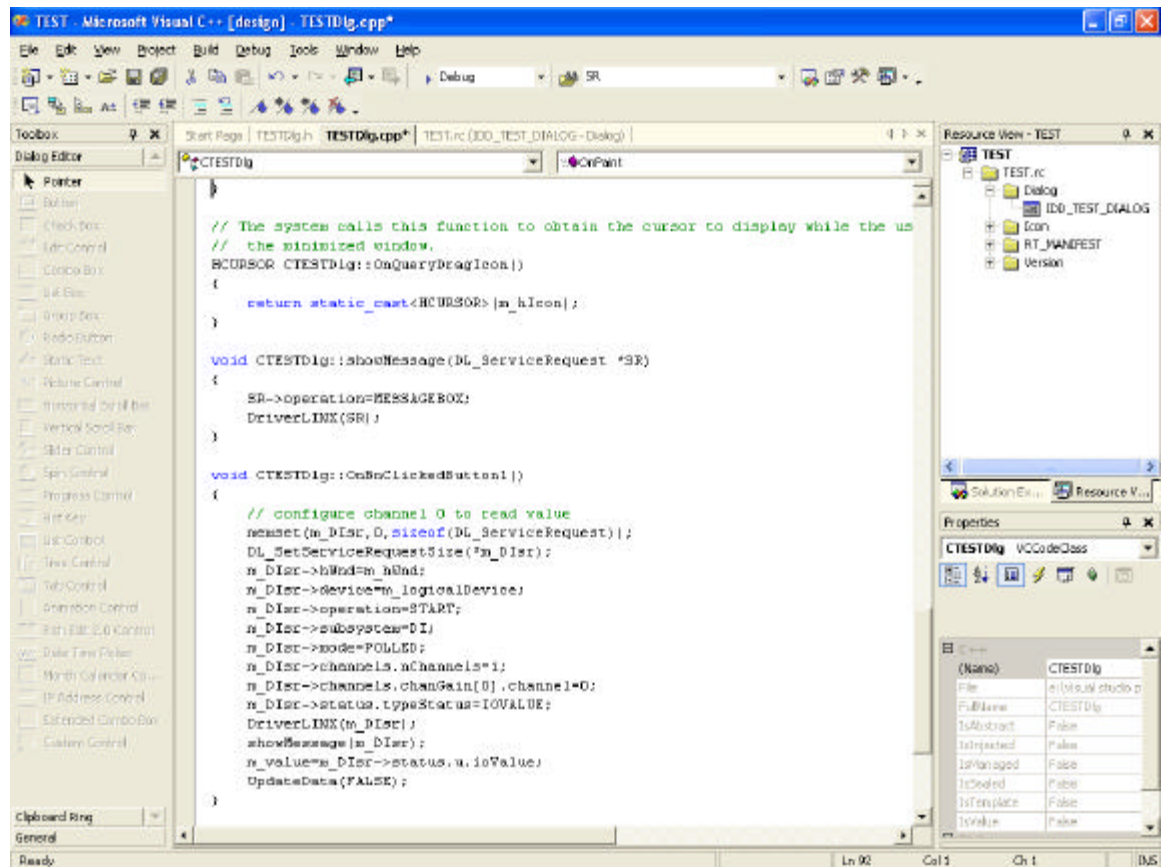
```
m_logicalDevice=0;  
m_DriverInstance=OpenDriverLINX(m_hWnd, " ");  
m_DIsr=(DL_ServiceRequest*) new DL_ServiceRequest;  
memset(m_DIsr,0,sizeof (DL_ServiceRequest));  
DL_SetServiceRequestSize(*m_DIsr);  
m_DIsr->operation=INITIALIZE;  
m_DIsr->device=m_logicalDevice;  
m_DIsr->mode=OTHER;  
m_DIsr->subsystem = DEVICE;  
m_DIsr->hWnd=m_hWnd;  
DriverLINX(m_DIsr);  
showMessage(m_DIsr);
```





**BEFORE** `void CTESTDlg::OnBnClickedButton1()` add the code below for an error checking function. NOTE: if your project is not named TEST, then the dialog class name will be different than CTESTDlg; modify accordingly.

```
void CTESTDlg::showMessage(DL_ServiceRequest *SR)
{
    SR->operation=MESSAGEBOX;
    DriverLINX(SR);
}
```



**UNDER** void CTESTDlg::OnBnClickedButton1()

```
// configure channel 0 to read value
memset(m_DIsr,0,sizeof(DL_ServiceRequest));
DL_SetServiceRequestSize(*m_DIsr);
m_DIsr->hWnd=m_hWnd;
m_DIsr->device=m_logicalDevice;
m_DIsr->operation=START;
m_DIsr->subsystem=DI;
m_DIsr->mode=POLLED;
m_DIsr->channels.nChannels=1;
m_DIsr->channels.chanGain[0].channel=0;
m_DIsr->status.typeStatus=IOVALUE;
DriverLINX(m_DIsr);
showMessage(m_DIsr);
m_value=m_DIsr->status.u.ioValue;
UpdateData(FALSE);
```

**UNDER** void CTESTDlg::OnQUIT()

```
delete m_DIsr;
m_DIsr=NULL;
CloseDriverLINX(m_DriverInstance);
m_DriverInstance=NULL;
CTESTDlg::OnOK();
```

