CS420/520 Lab 2
Adding a Dialog to WinMerge

- This tutorial requires WinMerge from https://winmerge.svn.sourceforge.net/svnroot/winmerge It uses version 7230, Please use the WinMerge solution you built following the instructions in the other Word document titled "Event Driven Programming 1 - How to Build WinMerge.doc". You will use the WinMerge source code I uploaded to Blackboard.

1. Open WinMerge in Visual Studio 2008 and confirm it can be built

2. We are going to add a pop up dialog. Select View -> Resource View.

3. The Resource View will open, but it could be on either side of VS. In the Resource View file tree, Select Merge -> Merge.rc -> Dialog.

4. Right click on Dialog and select Add Resource…

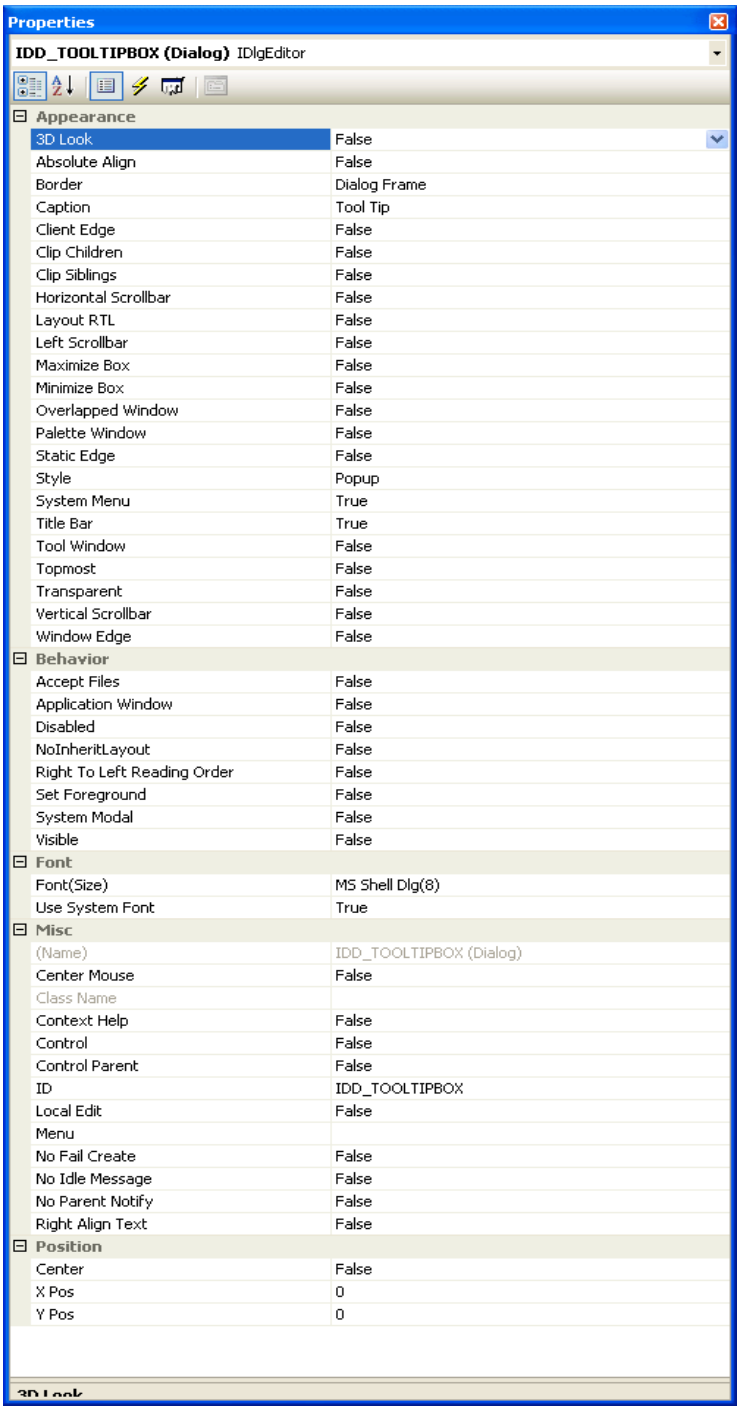
5. Expand the Dialog Menu with the plus sign, double click IDD_PROPPAGE_MEDIUM.
   *Note: The other dialogs have different features, you may want to explore them later, but if you add them now, VS can add things to the project that can be difficult to remove. You may want to experiment in an empty MFC project.

6. A new window appears you can change the size to whatever you like.

7. There is a button in the upper left of the Dialog Editor Toolbar, with a green arrow and dialog window, called Test Dialog, press it and it will show you the window. Press ESC to exit the window.

8. Right click on the window and select properties, the Properties pane will open, but it may be on the left or right, you can move it and resize it.

9. Set the properties as follows:



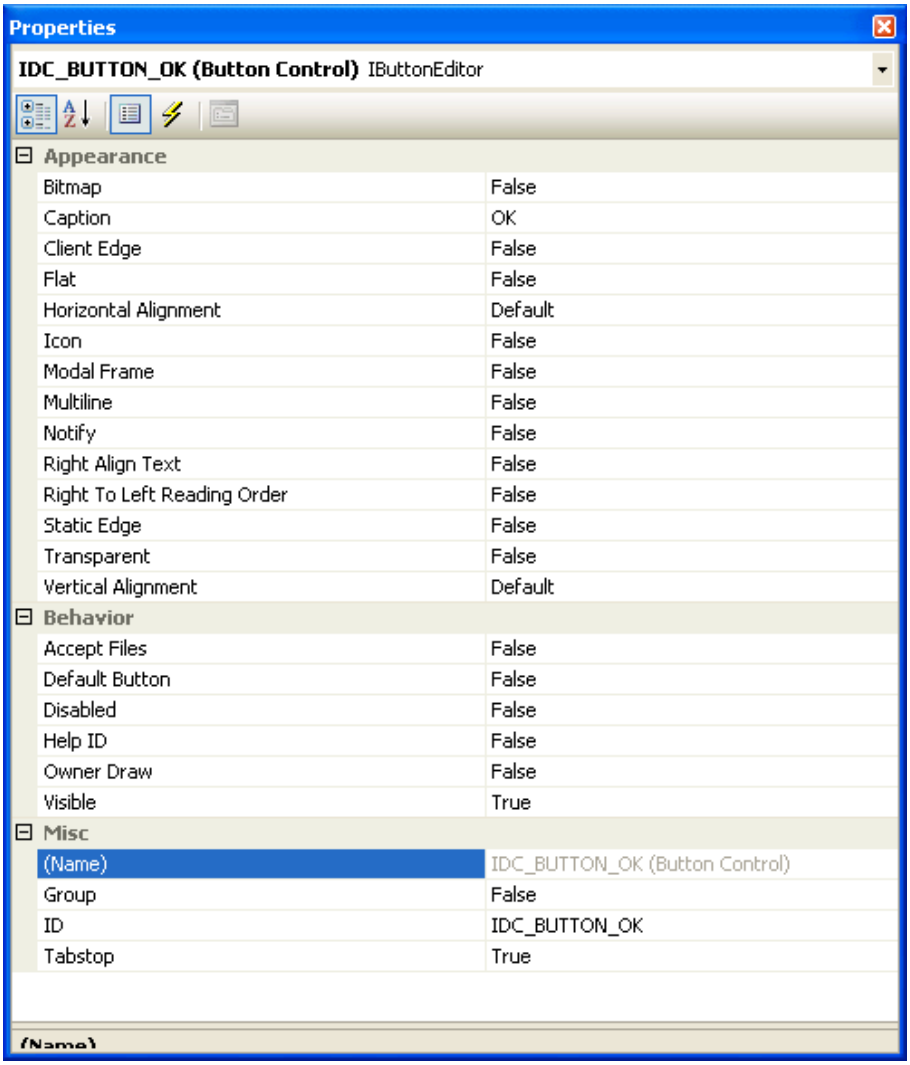
10. Again you can change these to make your window do different things, but the project is very sensitive and you may get errors. So only can one thing at a time should be supported for you to investigate what happens until you fully understand the meaning of the different Properties.

11. Right click on your window and select Add Class… a new wizard will open.

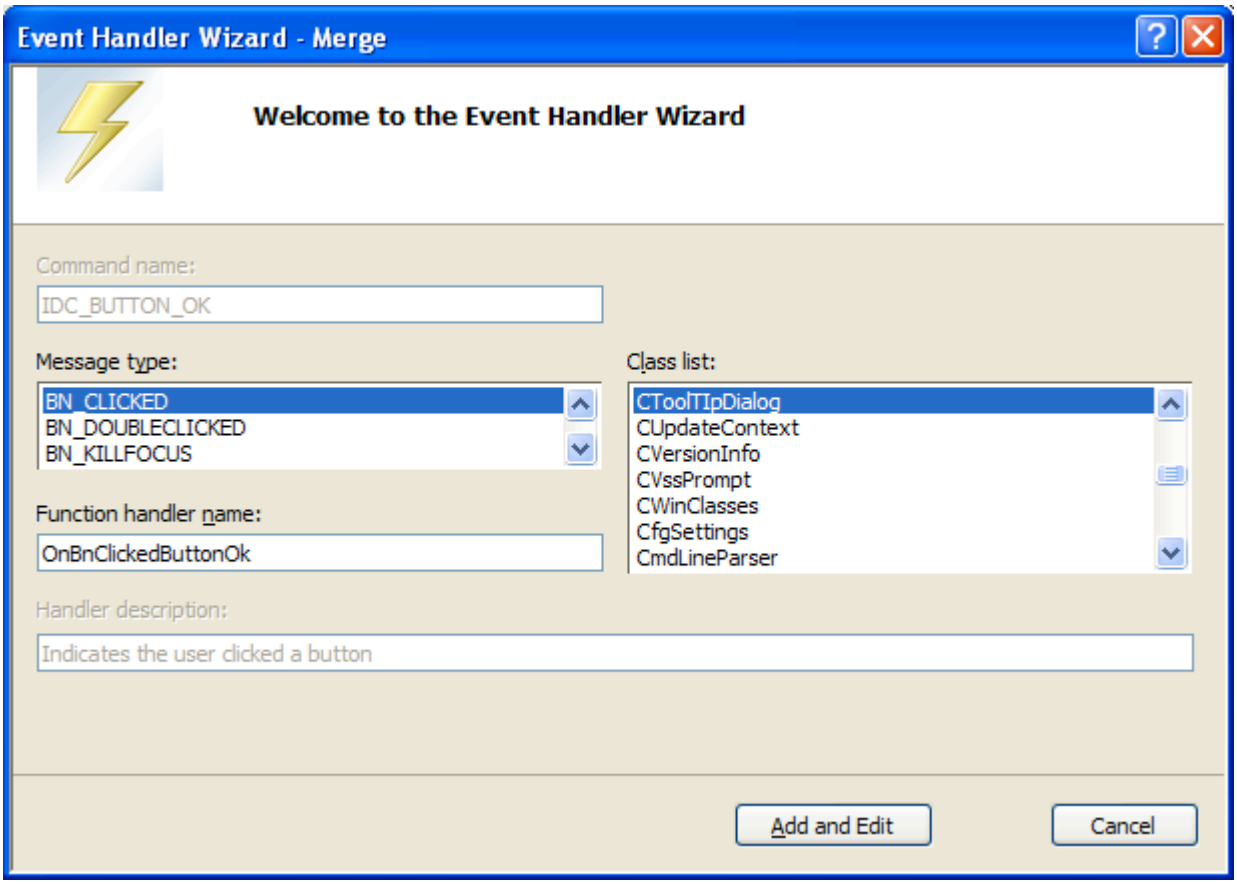
12. Under Class name type CToolTipDialog, notice the .h and .cpp file names are filled in, but without the C. Click on Finish.

13. On the left or right side of VS find a tab called Toolbox, click on it. There are many different objects we can add to our window. Left click on a Button and drag and drop it in to your window.

14. Set the Properties as follows:



15. Right click on the Button select Add Event Handler… a wizard will open. Make sure the values are as below and hit Add and Edit.



16. The wizard added a method to your class as follows:

```
BEGIN_MESSAGE_MAP(CToolTipDialog, CDialog)
  ON_BN_CLICKED(IDC_BUTTON_OK, &CToolTipDialog::OnBnClickedButtonOk)
END_MESSAGE_MAP()

// CToolTipDialog message handlers

void CToolTipDialog::OnBnClickedButtonOk()
{
      // TODO: Add your control notification handler code here
}
```

It also added the declaration to the .h file. The all caps part is the event handler. It tells the window, to pay attention to the button and call the `OnBnClickedButtonOk()` method if it's clicked.

17. Add the method call OnOK(); to the OnBnClickedButtonOK() method. This is an inherited method that will close the window.

18. The wizard added some things that we don't want to the .h and .cpp files, so make sure yours matches what's below. All we are doing is removing the association to the main window and a dynamic method.

ToolTipDialog.h

```
#pragma once

#ifndef _TOOLTIPDLG_H_
#define _TOOLTIPDLG_H_

#include "resource.h"

// CToolTipDialog dialog

class CToolTipDialog : public CDialog
{

public:
	CToolTipDialog();   // standard constructor
	virtual ~CToolTipDialog();

// Dialog Data
	enum { IDD = IDD_TOOLTIPBOX };

protected:
	virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support

	DECLARE_MESSAGE_MAP()
public:
	afx_msg void OnBnClickedButtonOk();
};

	#endif
```

ToolTipDialog.cpp

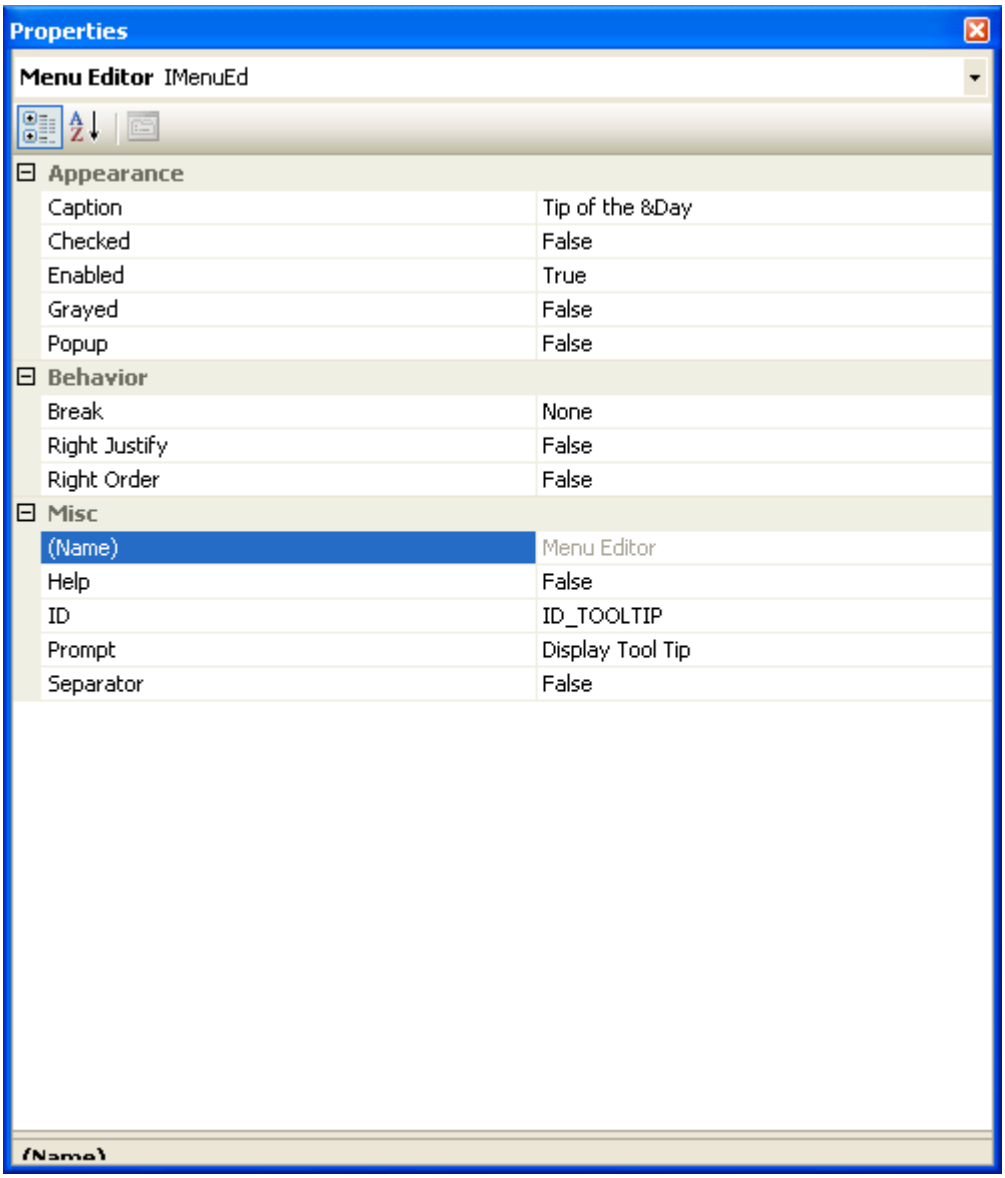
```
// ToolTipDialog.cpp : implementation file
//
#include "stdafx.h"
#include "Merge.h"
#include "ToolTipDialog.h"

// CToolTipDialog dialog

CToolTipDialog::CToolTipDialog()
	: CDialog(CToolTipDialog::IDD)
{
}

CToolTipDialog::~CToolTipDialog()
{
}
```

```
void CToolTipDialog::DoDataExchange(CDataExchange* pDX)
{
	CDialog::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CToolTipDialog, CDialog)
	ON_BN_CLICKED(IDC_BUTTON_OK, &CToolTipDialog::OnBnClickedButtonOk)
END_MESSAGE_MAP()

// CToolTipDialog message handlers

void CToolTipDialog::OnBnClickedButtonOk()
{
	// TODO: Add your control notification handler code here
	OnOK();
}
```

19. We now have a window, but we can't open it. To add an option to the menu bar at the top of WinMerge, go to the Resource View -> Merge -> Merge.rc -> Menu and double click on IDR_MAINFRAME.

20. At the top of the new menu, you'll see the menu bar from WinMerge. Navigate to Help and where it says "Type Here" add "Tip of the &Day" and hit enter. Notice what the & does.

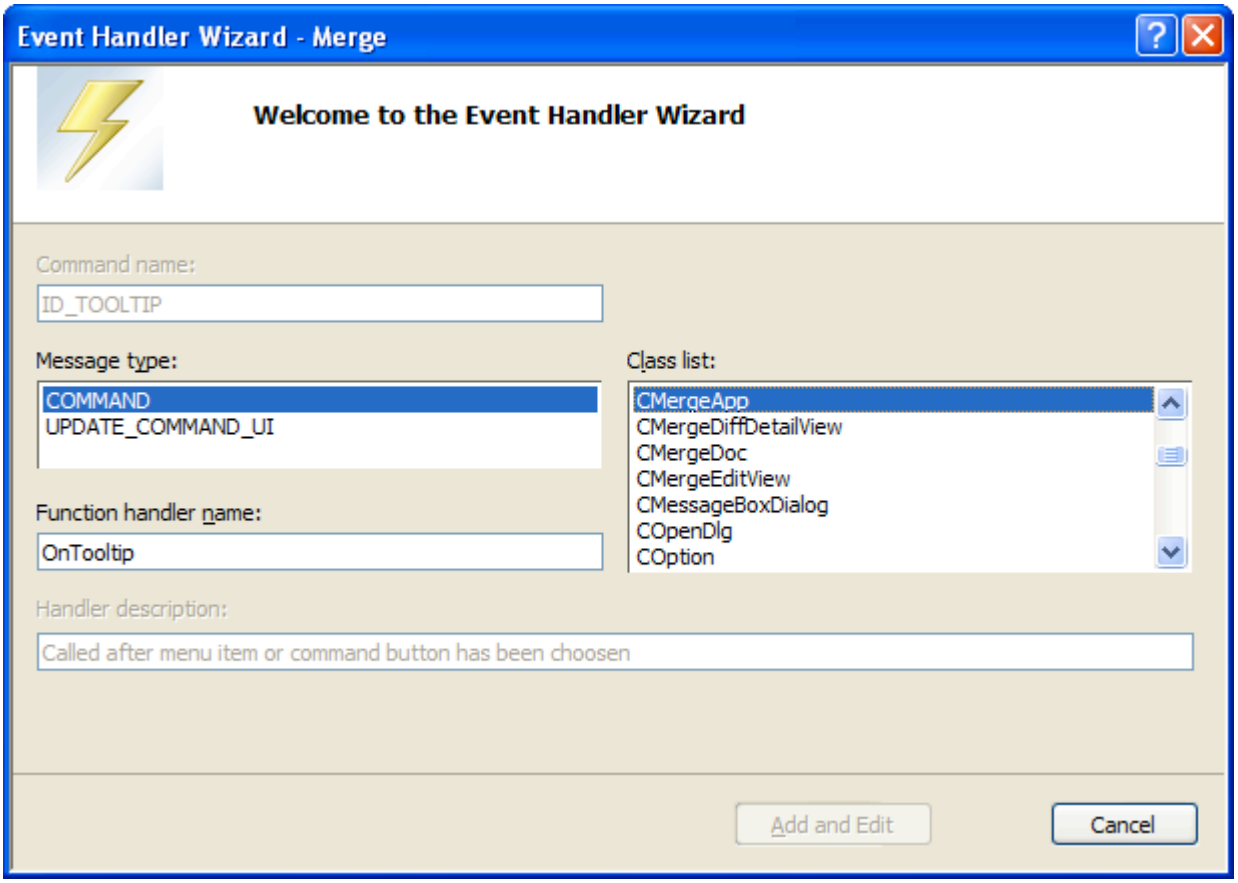
21. Set the properties as follows:



22. Notice the Prompt Property; it is what will appear in the status bar at the bottom of WinMerge, if we let the cursor hover over the Tip of the Day selection.

23. Right click on the Tip of the Day text you just entered, select Add Event Handler…

24. In the event handler window scroll down the class list and select CMergeApp. The rest should fill in by itself. **Don't click Add and Edit!** Make sure it looks like:



*Note: How can one figure out which class to add the handler to? That is discovered during Impact analysis. The concept of impact analysis and techniques will be covered in class, soon.

25. **Don't click Add and Edit**, this time we'll add the event handler manually to better see what it does, but you could use the wizard to add this handler.

26. To add the code manually, open the Solution Explorer, the view you're used to, and open Merge -> Source Files -> Merge.cpp. On line 139 we can see the event handler for the About Dialog Box. The Tool Tip box should work similar, so we'll use the same command.

27. On line 140 add ON_COMMAND(ID_TOOLTIP, OnTooltip) this handles the event, but now we need to add the OnTooltip method.

28. To add it near the About Box's OnAppAbout function, right click on the methodand select Go to Definition.

29. Just after the OnAppAbout method, add a similar method for the Tool Tip window.

```
void CMergeApp::OnTooltip()
{
	CToolTipDialog toolTip;
	toolTip.DoModal();
}
```

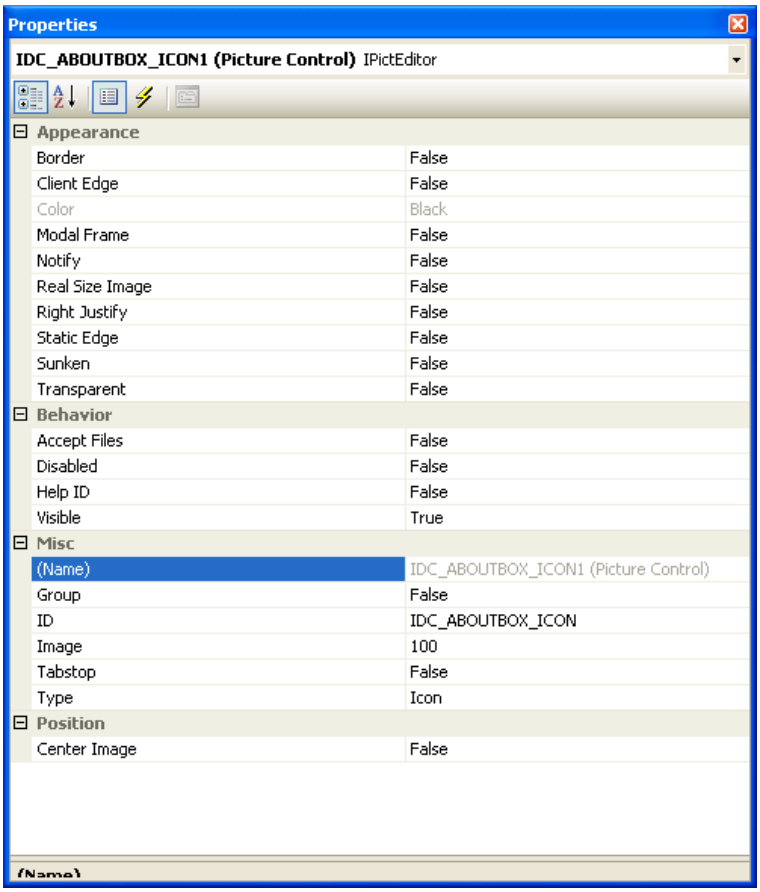
30. We also need to add it to the Merge.h file, so right click on the OnAppAbout method name and select Go to declaration.

31. Just below the highlighted line add: `afx_msg void OnTooltip();`

32. The only thing left is to include CToolTipDialog in Merge. Add `#include "ToolTipDialog.h"` with the rest of the header files.

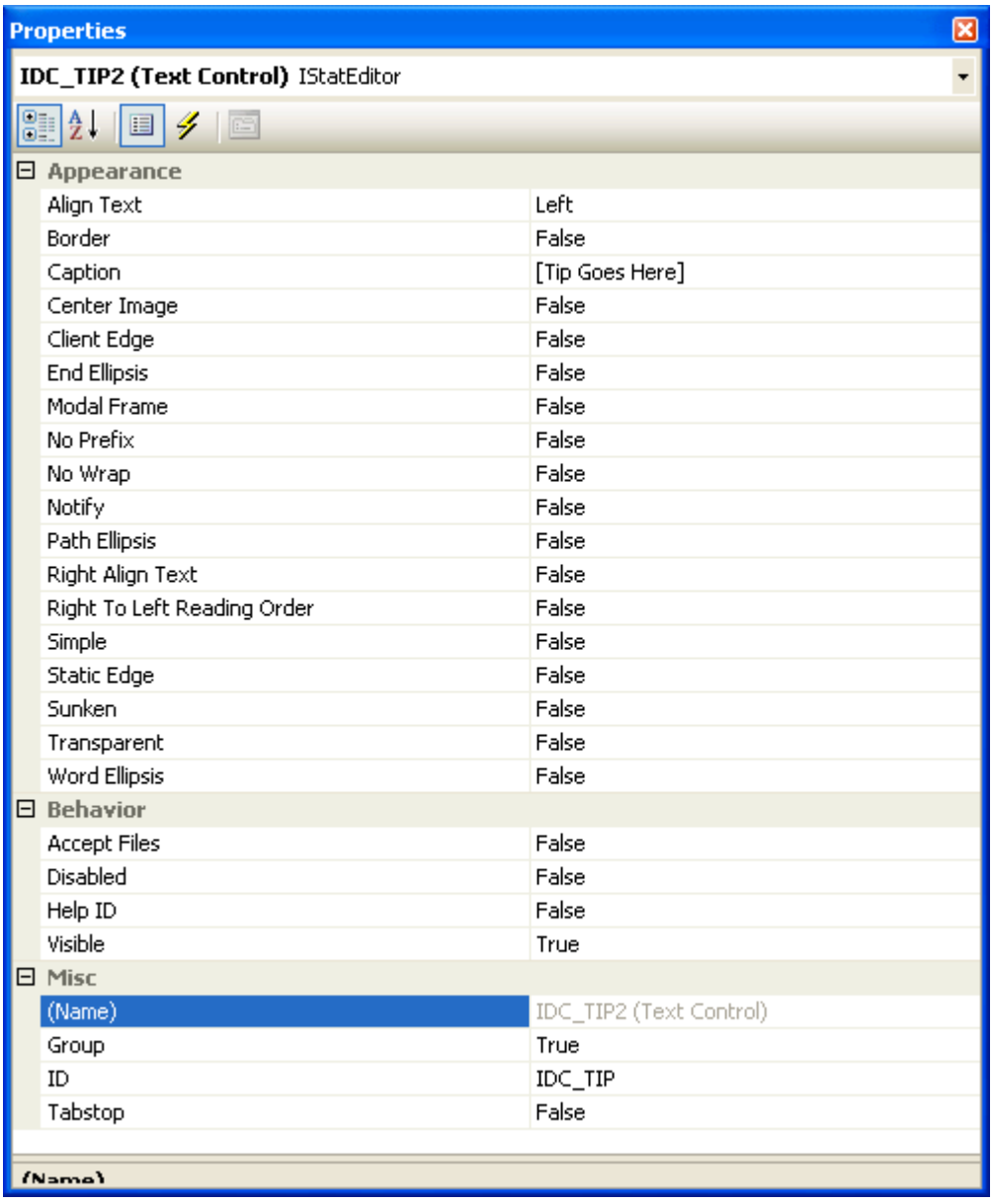
33. Compile and run your program. In WinMerge go to Help -> Tip of the Day and your window should pop up with an OK button that closes it.
34. Our window is pretty boring right now. To add the WinMerge icon, go back to the window layout. The tab is called Merge.rc (IDD_TOOLTIPBOX) and you can find the IDD_TOOLTIP BOX in the Resource view we opened in step 2 if you closed it.

35. From the Toolbox drag and drop a Picture Control in to your window. Then change its properties to match these:



36. It should now look like a piece of paper with a gray square. Don't worry; it will load the icon so that it will always use the current one.

37. There is a Static Text Box in the Tool Tip window that reads, “TODO: layout property”. Change its properties to:



38. To complete the Tool Tip Dialog, Add 4 more static text boxes from the Toolbox. Their ID’s and captions should be

ID	**Caption**
IDC_STATIC	WinMerge Tip of the Day
IDC_STATIC	Q.
IDC_STATIC	A.

IDC_TIPANSWER	[Answer goes here]

You'll notice the first 3 have the same ID, that's fine if the text never changes. Also, I'm not telling you how to layout your window. That is up to you.

39. In ToolTipDialog.h add two attributes of type CString and a method declaration as follows:

```
CString     m_tipQuestion;
CString     m_tipAnswer;
void getTipFromFile();
```

*Note: CString is a class similar to string that is used to get and send text from objects in MFC. To find out more go to: http://msdn.microsoft.com/en-us/library/aa300688(v=VS.60).aspx

40. To change the text in the static boxes do a data exchange. Add the following method calls to the DoDataExchange method in ToolTipDialog.cpp.

```
DDX_Text(pDX, IDC_TIP, m_tipQuestion);
DDX_Text(pDX, IDC_TIPANSWER, m_tipAnswer);
```

41. To set the attributes to a tip, add the following method:

```
void CToolTipDialog::getTipFromFile(){

      std::ifstream inFile;          // Input file variable
      inFile.open("..//Docs//Users//FAQ.txt");  // Open the input file.
      if (!inFile){                 // if to check for file error.
            m_tipQuestion = "Couldn't open faq file";
            m_tipAnswer = "";
            return;
      }
      char questions[10][401];
      char answers[10][401];

      for (int i = 0; i < 10; i++){
            inFile.getline(questions[i], 400, '\n');
            inFile.getline(answers[i], 400, '\n');
      }

      m_tipQuestion = questions[0];
      m_tipAnswer = answers[0];

      inFile.close();         // Close input files.
}
```

42. You will also have to include the fstream library, to use ifstream.

43. We need to override a method from the base class, CDialog, so that we can run our method when the window opens. Add the following to ToolTipDialog:

```
BOOL CToolTipDialog::OnInitDialog()
```

```
{
            CDialog::OnInitDialog();


   getTipFromFile();

   UpdateData(FALSE);

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

44. Then add its declaration to ToolTipDialog.h. `virtual BOOL OnInitDialog();`

45. Compile and run it again. It should display a Tip of the day now. You can get to it by using the menu bar or by selecting ALT -> H -> D on your keyboard. By putting the & before the 'D ' in Day when naming the option, MFC automatically added the hot key.

46. The window is almost done now. The only thing left to do is to make the tips randomly appear when the window is opened. That part of the assignment is left to you. It should be relatively easy. There are 10 tips and answers. They are already loaded in the arrays "questions" and "answers" all you need to do is randomly pick one of the ten questions and answers.

47. After that, clean your project put the whole project in a zip file called **YourLastName_YourFirstInitial_Lab2.zip** and submit it through Blackboard by **Tuesday January 29, 2013 at 6:00 PM.**

Step	Task	Points	How to get credit
1 to 45	Adding a window to WinMerge	50	The window must open and close, have a button, the WinMerge Icon, have 3 static text boxes and 2 static text boxes that read from a file.
46	Adding the random tips	40	All of ten tips and answers must randomly appear one at a time in the 2 static text boxes.
47	Submitting a cleaned project with the proper naming convention.	10	Submit cleaned project with proper naming convention.
	Extra Credit	10	Make the window appear at startup, instead of the splash screen.
*Note: Any projects that do not compile will receive 0 points			