Managing A Project, Reading

As with any application that has so many parts as does Visual Studio, it takes a lot of practice to find your way around. There are terms for things and you have to know what they are in order to make any progress at all! This reading explains the terms you need to know -- *refer back to this often* as you develop GUI apps in VS.

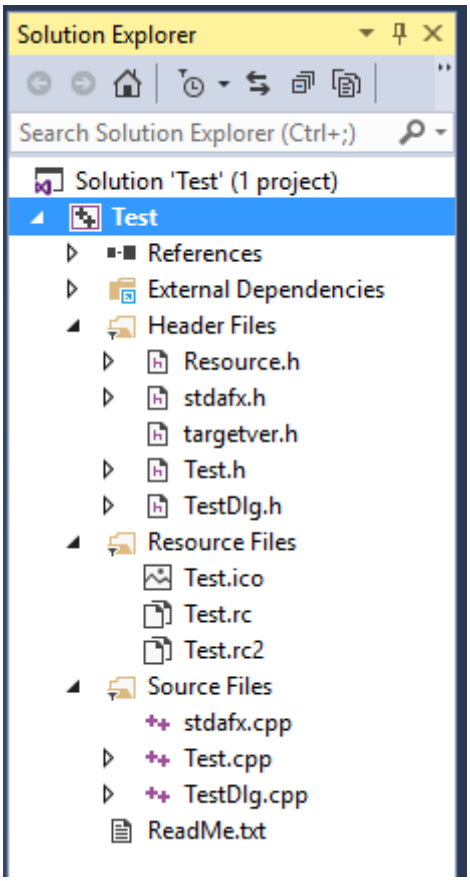
Projects

In VS, an app is a "project". Projects are stored in a folder named after the project. The various CPP and H files are in that folder and its sub-folders. So are a number of *"resource" files* that are also part of your program. For example, there's a resource file that stores the design of your window and its controls.

There are also files that the compiler and VS use in order to build your app -- and the app's EXE file is there too.

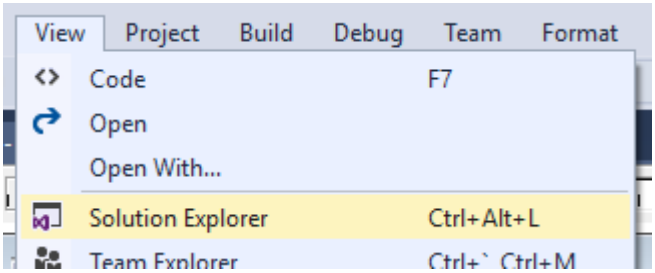
Go ahead -- build a Hello World app for yourself. Find its project folder and explore for a bit.

Solution Explorer Window

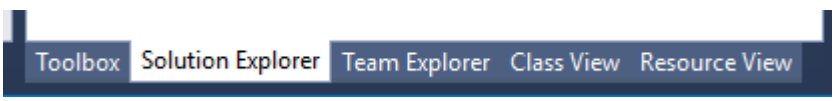


There's a window called the "Solution Explorer" which is a version of Windows File Explorer that shows the organization of files in your prc cannot get very far without this window, so the first thing you need to how to find it!

If you cannot see it, use the View menu to find and open it:



Double-click any file to edit it, but when you do, the solution explorer away. Look for this set of tabs at the bottom of the solution explorer, a can use it to get back:

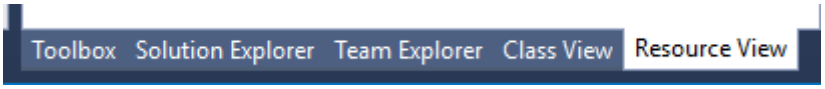


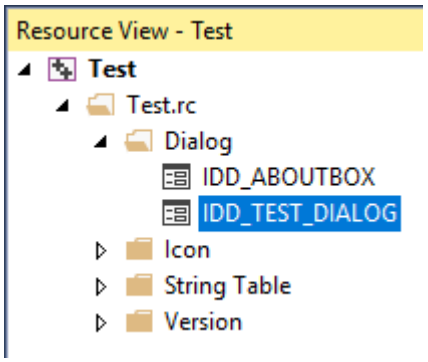
Make very sure you can always find the solution explorer, because it's how you'll control your app development!

Resource View Window

Besides direct code editing, there's the *visual* part of "Visual" Studio. It's the ***visual layout editor*** with "TODO Place dialog controls here" in the middle of it. That's where you'll design the *user* interface for your app. What you do there with your mouse and keyboard translates into code that code in the "resource files".

The Resource View looks like the Solution Explorer. Get there by double-clicking your project's .RC file, or click in the tabs at the bottom of the window:





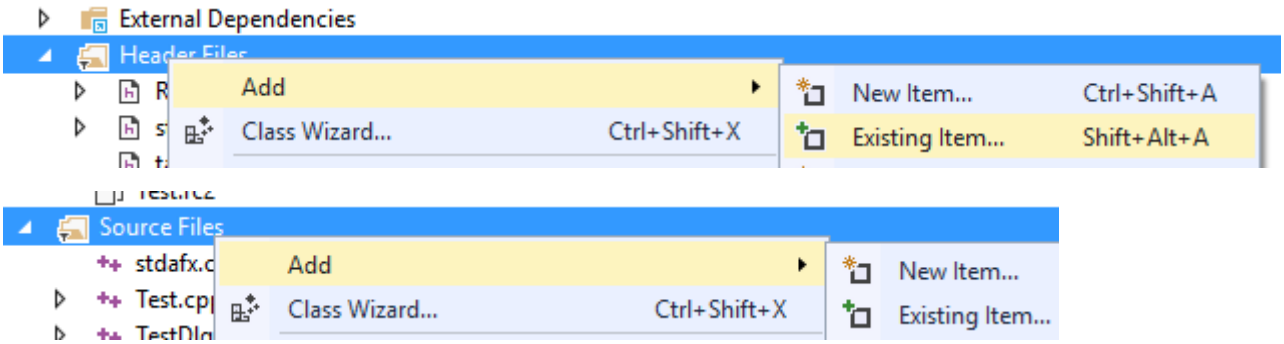
Double-click IDD_..._DIALOG (named to match your project's name) to get back to the visual layout editor. Note that there is also an IDD_ABOUTBOX that you may wish to edit, to personalize your app with its name, version number, and your copyright! More on personalization later...

"***Dialog***" refers to a kind of window that's a rectangular panel with controls on it. It's the kind of apps that we're learning to write. You design and edit your app's dialog window using the VS visual layout editor.

Make very sure your can always get back to the visual layout editor! Remember that the View menu gets you to your Solution Explorer, and that gets you to the resource view.

Header Files And Source Files

Without writing any yourself, VS creates several H and CPP files for your project automatically. If you have files of your own to add to your project, you can either create them or import them using this menu -- ***right-click*** over Header Files or Source Files in the Solution Explorer:



To import existing H or CPP files, *first* copy them to the folder in

your project that has the CPP and H files created by VS. It's the one with the `res` folder and **stdafx.h**. Then use the "Existing Item..." option to add them.

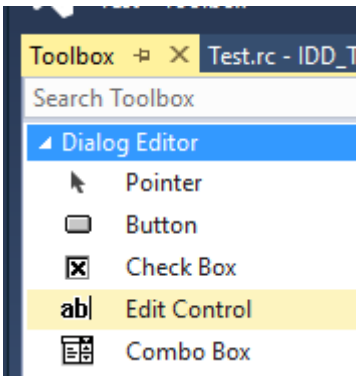
It is very important that H files *not* be listed as "source files"! H files are *not* stand-alone compilable -- CPP's *are*.

Source files are the CPPs of your project. The one named after your project and ending in **...Dlg.cpp** is the one where you'll do most of your coding. You'll be able to get to the **...Dlg.cpp** files easily when you start adding visual "controls" to you app's "dialog window", because you'll be able to double-click any control to go directly to the member function that "handles" it. Note that "Dlg" is an abbreviation for "Dialog"...

If you do ***add a CPP***, make sure its first #include is `#include "stdafx.h"` -- if you forget this, you'll see *"unexpected end of file while looking for precompiled header directive"* when you compile.

Toolbox and Controls

Use the "Toolbox" to place buttons and labels and menus and such in your app's main dialog window. The "Dialog Editor" section should be open by default, but if it's not, open it yourself.



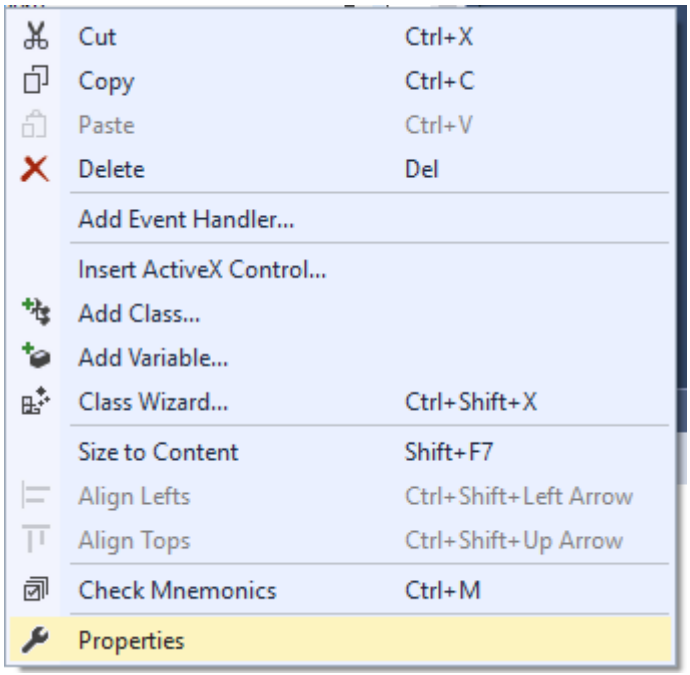
The tools we'll learn to use in this course are:

- Button
- Check Box
- Edit Control (text I/O)
- Combo Box (menus)
- Radio Button
- Static Text (labels)
- Picture Control (embedded images)
- Tab Control

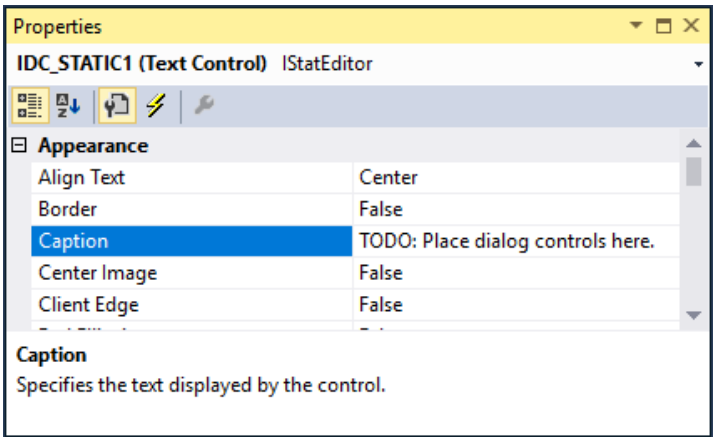
Controls are stored by your app in a bag of diverse object types, just like the shape objects in our Shapes app. When you access a particular object from the bag, you'll be doing so with a `reinterpret_cast` pointer, just as in the Shapes app!

Properties Window

Controls are objects in a bag of diverse object types. As such they have attributes specific to their type, but in the MFC they are called "***properties***". Since the visual layout editor writes code for you, the way you assign initial values to control objects' properties (instead of using a constructor or initializer list) is through its *"properties window"*.



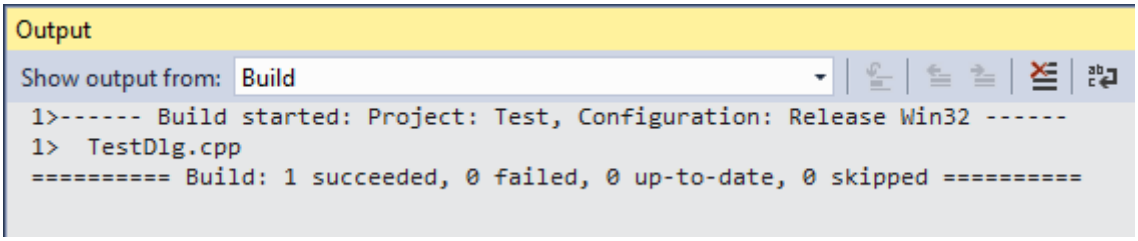
Right-click over any control to see a menu of options -- the only option we'll be using from that menu is the last one -- "Properties". Its content depends on the control selected (as you'd expect, since the bag contains all kinds of controls). Here's the one for the "TODO" label control -- note that you can edit its text:



Output Window And Compiling

To compile any individual CPP in your list of Source Files, double-click it, then use the Build menu's Compile option. If you don't see the option at the bottom of the Build menu's options, then you've not selected a CPP file...

The results of compilation appear in the Output Window. If you cannot find it, use the View menu's Output option.



Starting Over

It's fairly easy to lose control of the placement of all these windows in VS. So if that happens, use the Window menu's Reset Window Layout option.