

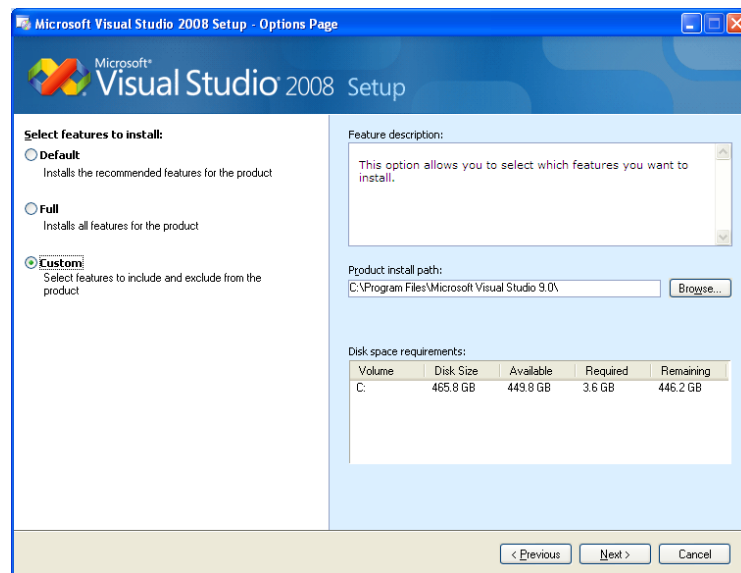
Using Coin3D with Microsoft Visual Studio for MECH 434/534 Assignments

During this course, you will be asked to complete 7 Homework assignments and 3 projects using Coin3D, 3D Graphics Developer Kit. Coin is an OpenGL based, retained mode 3D graphics rendering library. It is implemented in C++ and publicly available at <http://www.coin3d.org/>. In the rest of the document, we will provide you with step-by-step installation instructions for Microsoft Visual Studio 2008. Note that it is also possible to complete the assignments and projects with a different IDE (Integrated Development Environment) as long as your programs run correctly and adhere to standards. If you use a Mac or Linux machine, I would suggest you use the Qt Creator.

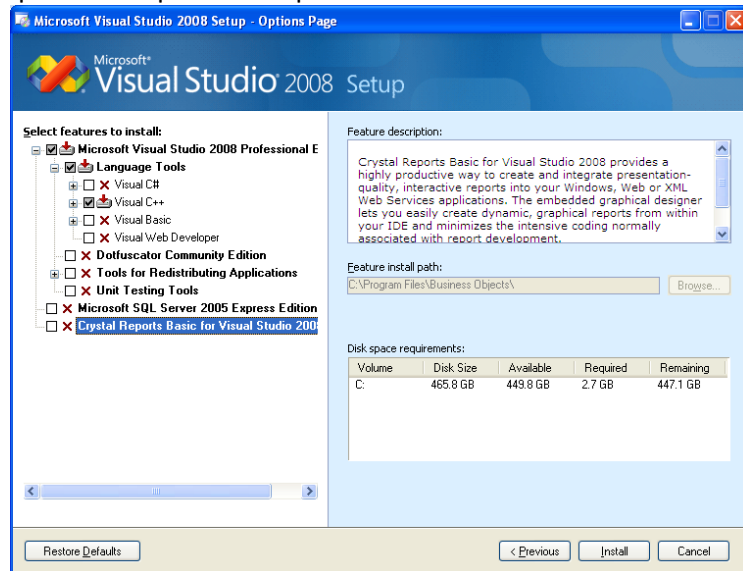
Installing Microsoft Visual Studio 2008 on your computer

If you have not already done so, install Microsoft Visual Studio 2008 (version 9) on your computers with the C++ option. To do this, you can drop by the CIT office, and they can install it for you. If you would like to install the IDE yourself please follow these steps:

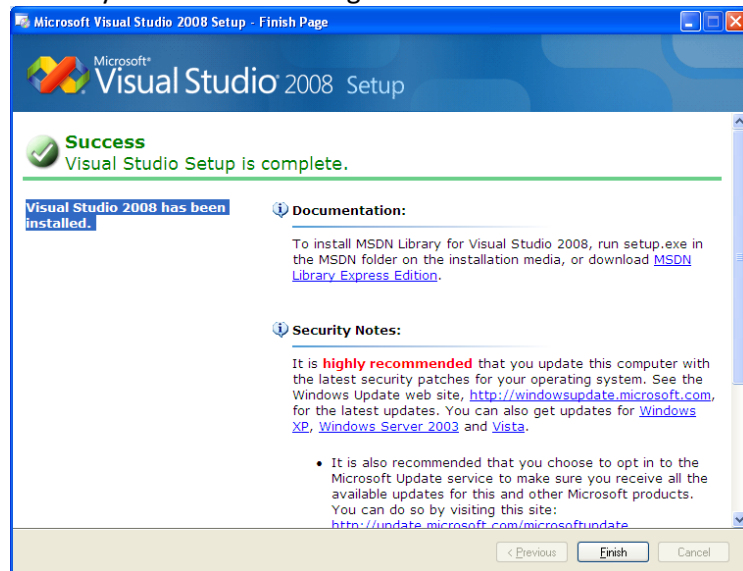
1. Run setup
2. Proceed until you reach the Options Page during setup and select Custom:



3. Uncheck all options except C++ and press Install:



4. You are done when you see the following screen:

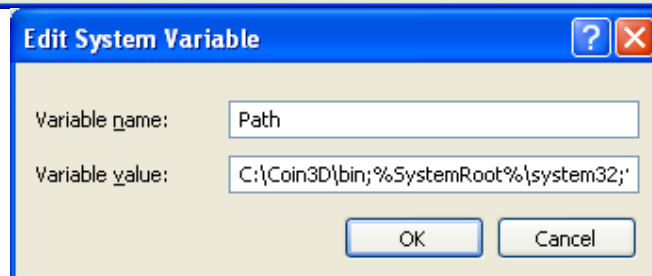
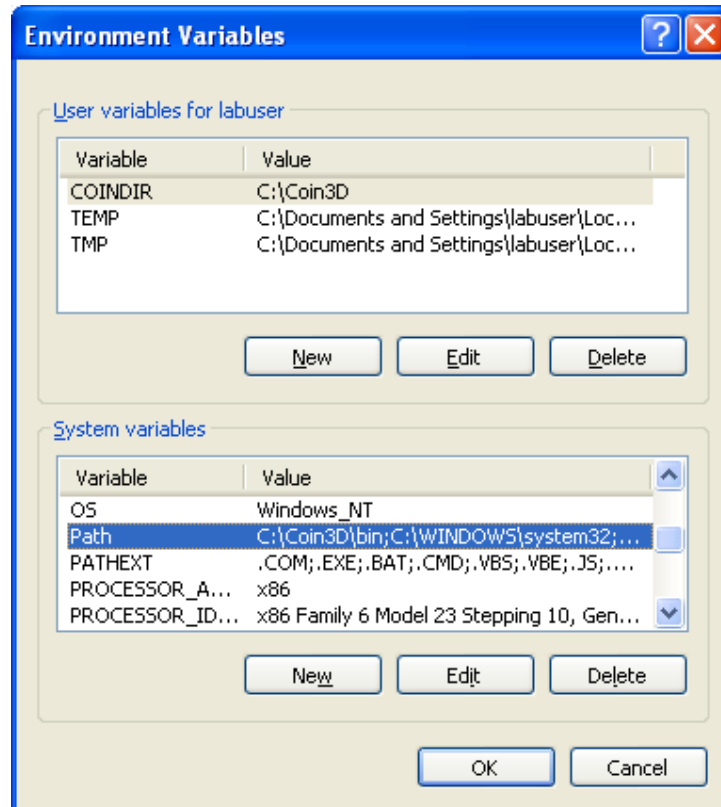


Installing Coin3D on your computer

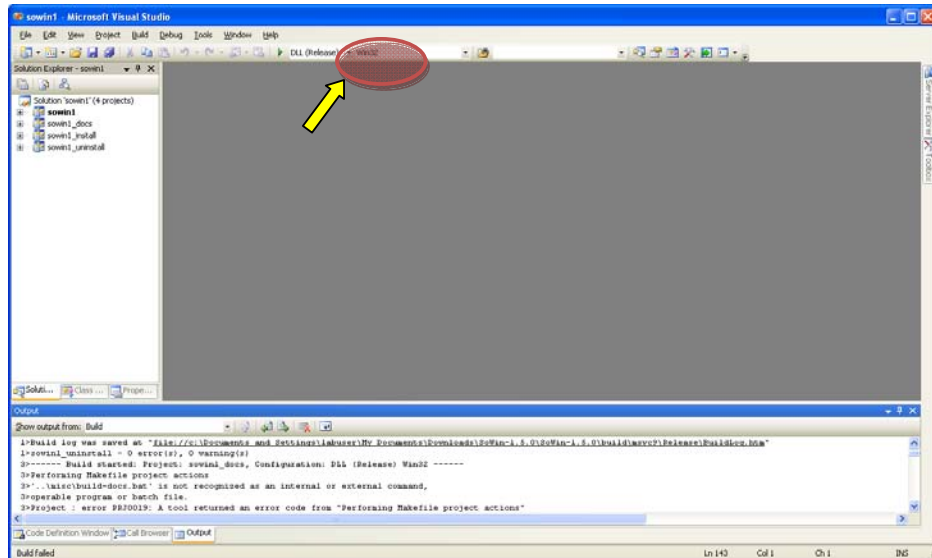
Having installed the Visual Studio, it is time to install Coin3D, the graphics library we will use for MECH 434/534. The latest version is publicly available in the web so you can download and install it on your own machines.

1. Go to <http://www.coin3d.org/lib/downloads>
2. Select the option "I'm Developing Open Source Software"

3. On the next page, select “Continue to open source downloads (for free software developers)”
4. Select the latest Coin release and the latest SoWin release from the list (You will need SoWin in order to write Windows API programs).
 - a. Download Visual C++ 9.0 binaries for the latest Coin release for Windows.
 - b. Download one of the source code archives for the latest SoWin release.
5. Install Coin library
 - a. Extract the archive file that contains the Coin release to a local directory you choose for your Coin installation. The path to the directory should not contain spaces. Something like C:\Coin3D will work. In the end, the coin directory should contain 4 subfolders: *bin*, *include*, *lib*, and *share*.
 - b. Define an environment variable named COINDIR to point to the directory you created. You can do this from Control Panel | System | Advanced | Environment Variables | New.
 - c. Also add the bin folder to your system path by editing the existing Path Variable:



- d. These operations might require you to restart your computer for the changes to take effect, hence restart your computer.
6. Compile and install SoWin
 - a. Extract the archive file that contains the SoWin release.
 - b. Open Microsoft Visual Studio Solution *sowin1.sln* residing within the folder *build\msvc9* by double clicking on it.
 - c. If your configuration is correct, the solution will be opened in Visual Studio.
 - d. If you are using a 64-bit system, select x64 for the solution platform (OPTIONAL FOR x64 users).

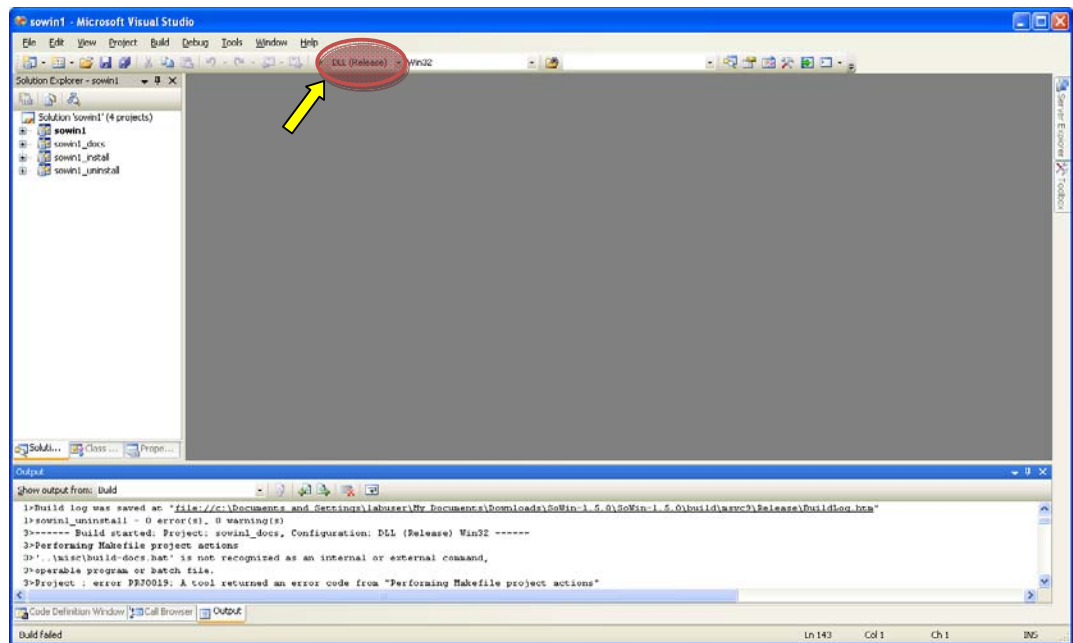


- e. Select Build → Build Solution from the menu or basically hit F7 to compile the code. When this operation is finished, in the output window (at the bottom portion of the IDE), you will see the following message:

```
===== Build: 3 succeeded, 1 failed, 0 up-to-date, 0 skipped =====
```

Don't worry about the error, it only indicates that the documentation could not be installed, which is not a major issue.

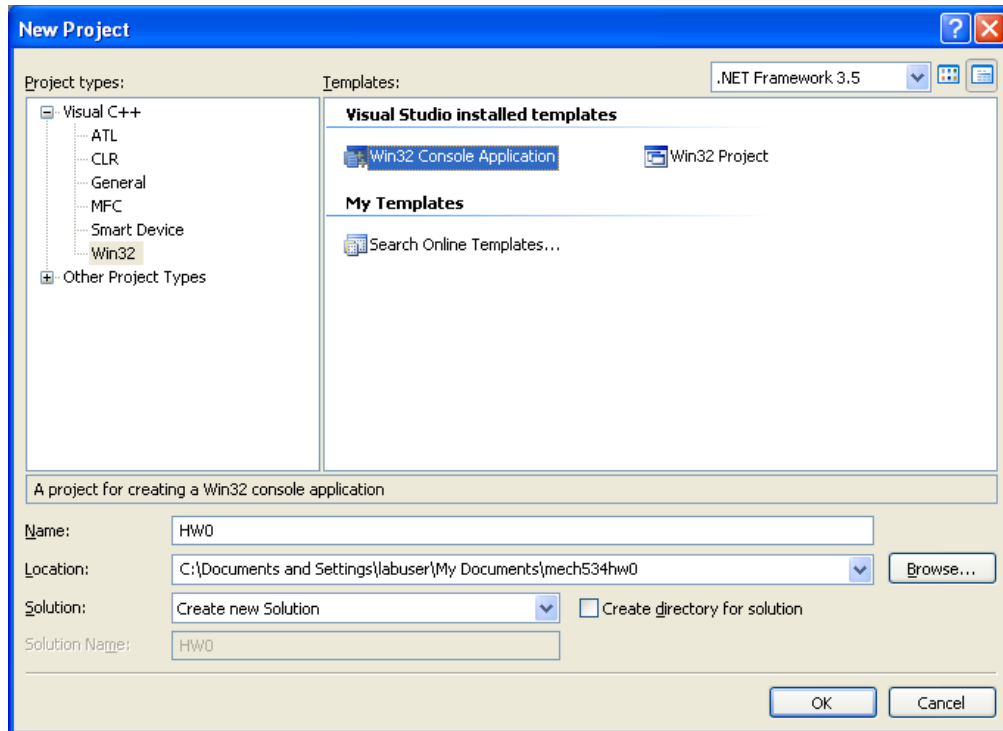
- f. Do the same build operation also for the Release configuration by selecting DLL (Release) from the configuration pull-down list.



- g. Check your Coin installation folder. It should now contain a new folder named *data*.
-

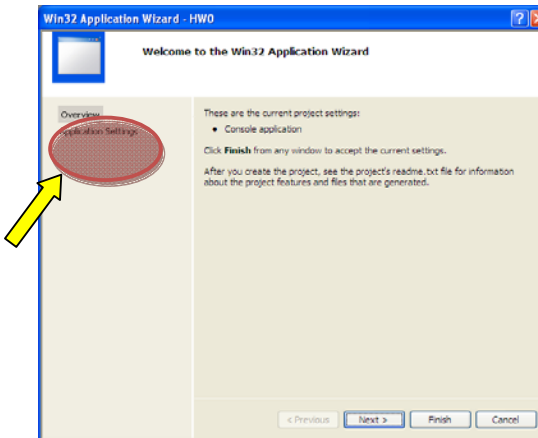
Testing your installation, setting up a project and compiling your first program

1. Run Visual Studio and select File → New → Project from the menu.
2. Select project properties in the dialog box by executing the following steps:

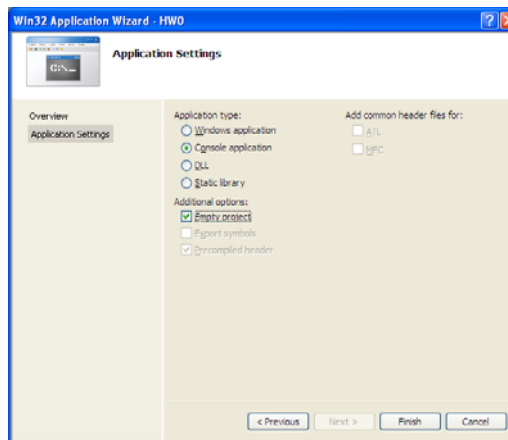


- a. Select **Win32** as the project type and **Win32 Console Application** as the template.
- b. Give a name for your project.
- c. Choose the location on your computer where the project files will be saved to.
- d. Uncheck *Create directory for solution* checkbox.
- e. Click OK.

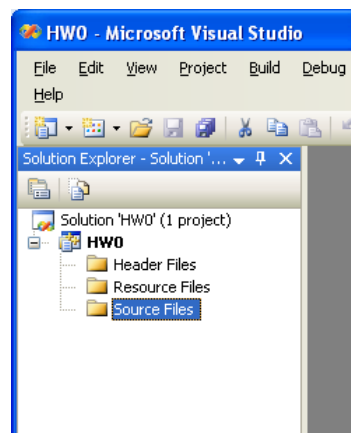
- f. You will see a confirmation screen. Click on **Application Settings** before hitting Next.



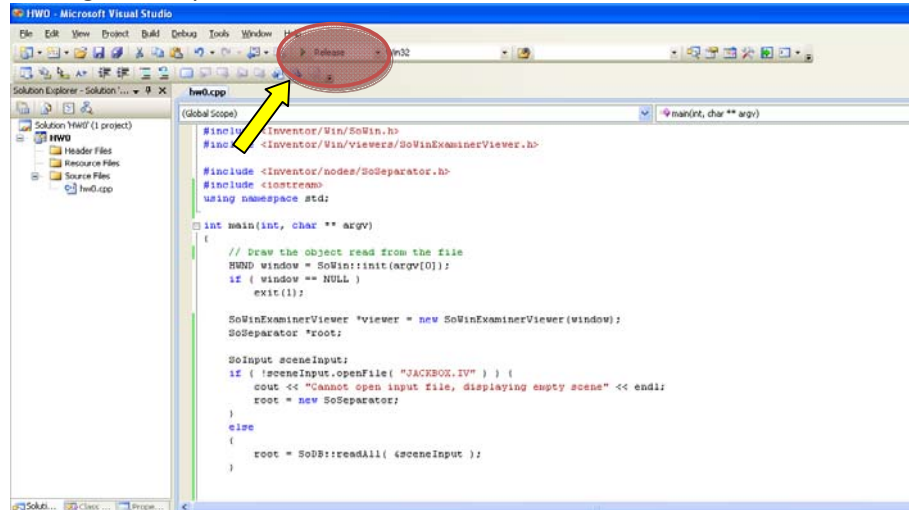
- g. Check *Empty project* and hit Finish.



3. Download main.cpp and jackbox.iv to your computer and paste them to the newly created folder.
4. Add main.cpp to your project either by
 - a. dragging and dropping main.cpp to the Source Files folder in Solution Explorer
 - b. right clicking Source Files, then selecting Add → Existing Item...

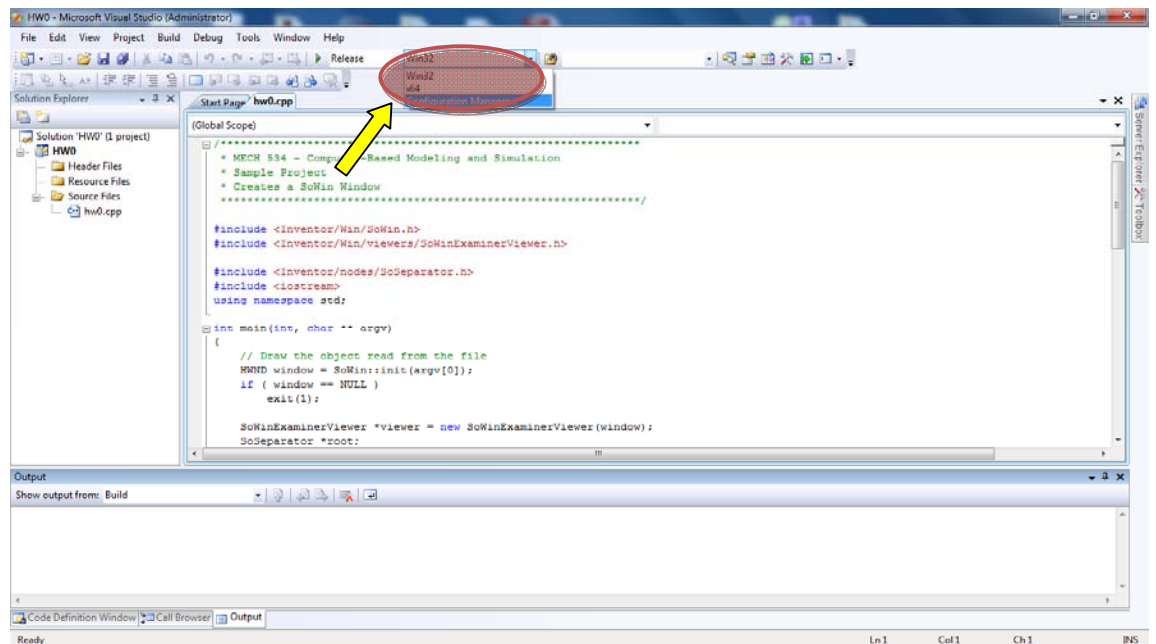


5. Select configuration profile as Release

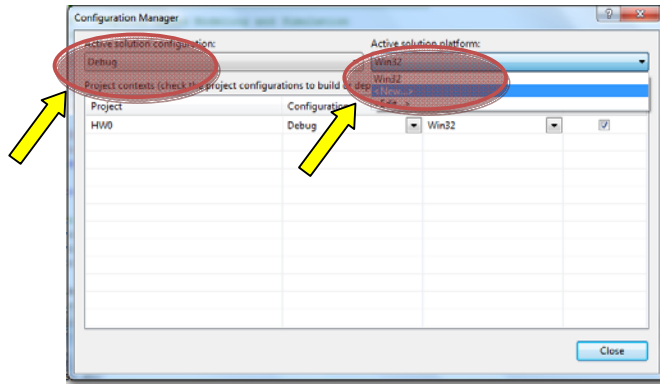


6. If you are using a 64-bit system, select x64 for the solution platform (OPTIONAL FOR x64 users):

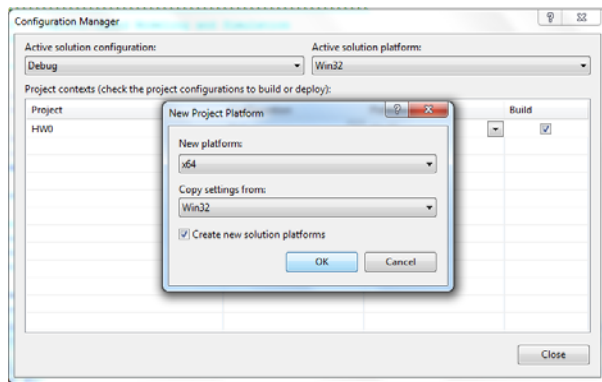
- a. Open Configuration Manager by using the Solution Platforms drop down menu.



- b. Create a new platform by selecting *Active solution configuration* as **Release** and selecting <New...> in *Active solution platform* drop down menus.



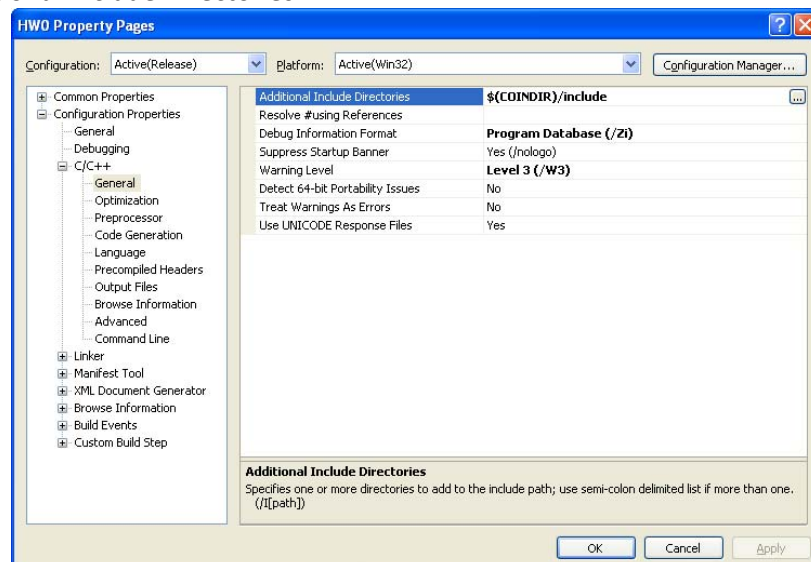
- c. Create a new platform by choosing **x64** in *New Platform* and by selecting Copy setting from **Win32**. Click OK and close Configuration Manager.



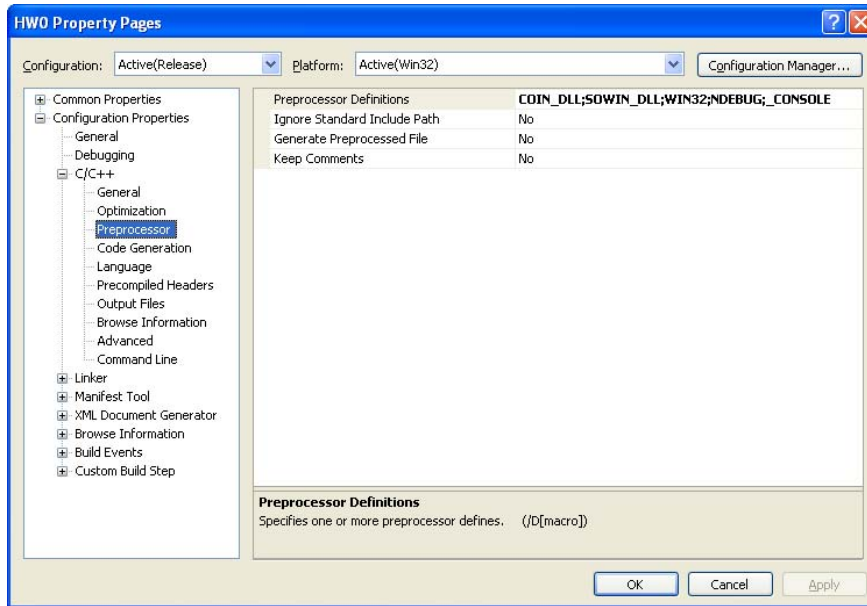
- d. Finally, select x64 as the solution platform in the drop down list.

7. Configure project properties

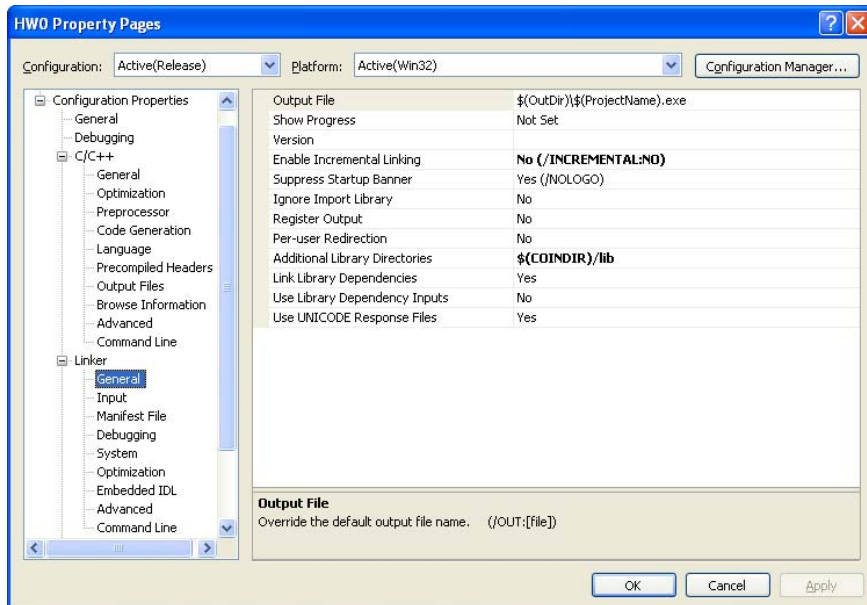
- Select Project → *<your_project_name> Properties...* from the menu
- In Configuration Properties | C++ | General, add **\$(COINDIR)/include** to *Additional Include Directories*:



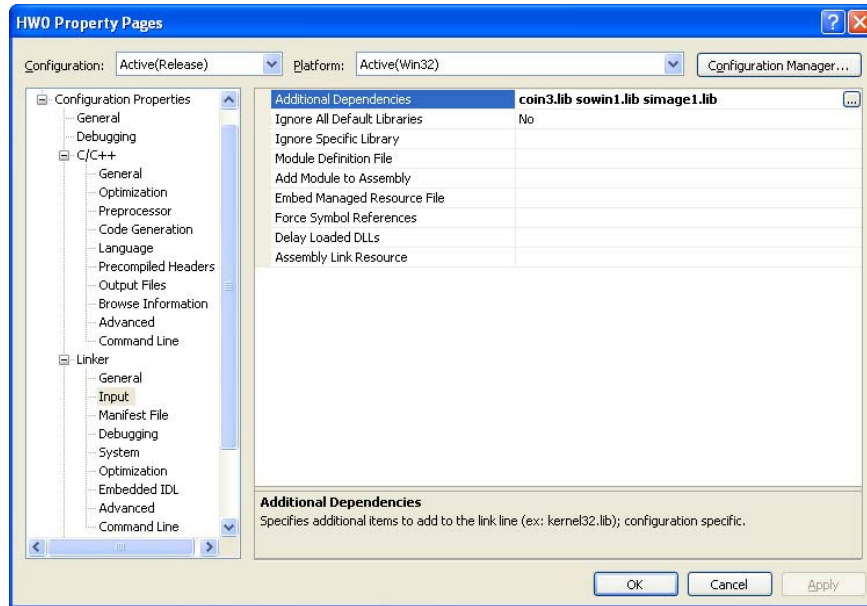
- c. In Configuration Properties | C++ | Preprocessor, add **COIN_DLL;SOWIN_DLL** to *Preprocessor Definitions*:



- d. In Configuration Properties | Linker | General, add **\$(COINDIR)/lib** to *Additional Library Directories*:



- e. In Configuration Properties | Linker| Input, set *Additional Dependencies* as **coin3.lib sowin1.lib simage1.lib**:

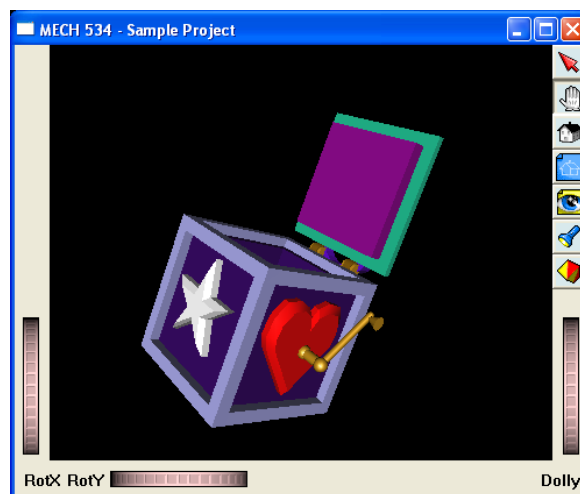


- f. Press OK.

8. Compile your program by selecting Build → Build Solution or simply by pressing F7.
9. If everything is configured correctly, you will see the following message in the output window:

```
===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
```

10. Run your program by selecting Debug → Start Without Debugging or simply by pressing Ctrl + F5.
11. If your installation is successful, a new SoWin window will be created:



12. You can reuse the same project files in your other assignments by renaming the projects and replacing the source files.