Installation Procedure

These notes explain how to compile programs written in ANSI C with OpenGL and GLUT using the Dev-C++ compiler.

Bloodshed Dev-C++ is a free C++ compiler and development environment for Windows operating systems. Like most C++ compilers, it also can be used to compile ANSI C. By installing the GLUT header and library files, it can be used to write programs that use OpenGL. This is needed to run programs for Edward Angel's textbook, *Interactive Computer Graphics* 5th edition and possibly other computer graphics texts.

These notes do not explain how to compile OpenGL with C++ . The 6th edition of Angel's book uses C++ which will not work with these notes.

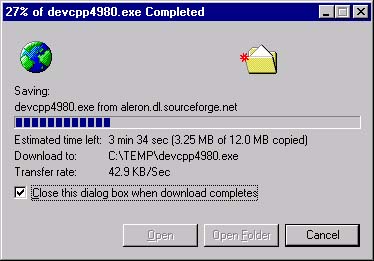
These instructions have been tested on a small variety of Windows 2000 and Windows XP systems. These systems come with the files needed for OpenGL, but not the files needed for GLUT.

Dev-C++ does not work well with Microsoft's Vista. The problem, and a possible fix, is discussed here: <http://aresio.blogspot.com/2007/06/vista-and-dev-cpp.html> but I have not tested this information.

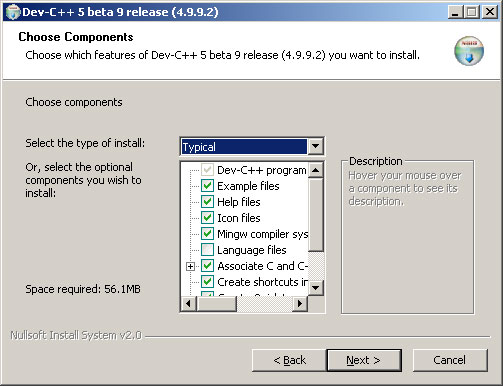
**I. Download Dev-C++ from** [**http://www.bloodshed.net/dev/devcpp.html**](http://www.bloodshed.net/dev/devcpp.html) **and install it.**

Details:

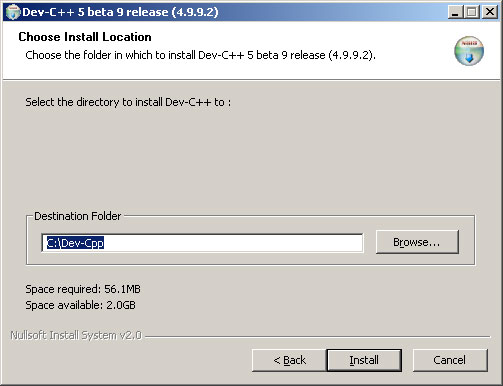
Get Dev-C++ 5.0 beta 9.2 (4.9.9.2) (9.0 MB) with Mingw/GCC 3.4.2 Although this is a "beta" version, it works perfectly fine. Click on SourceForge to go to a list of download sites and pick one. The file will be something like devcpp4.9.9.2\_setup.exe. Save this file in a place like C:\Temp.



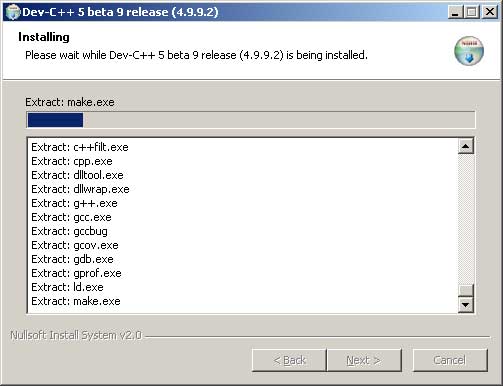
When the download is complete, click on the "open" button to start the installation process. (Or go to C:\Temp andDouble click on devcpp4.9.9.2\_setup.exe). You will see a few screens that ask you to pick a language (English) and to agree to the license terms. Choose a "typical" installation.



Accept the suggested destination for the installation:

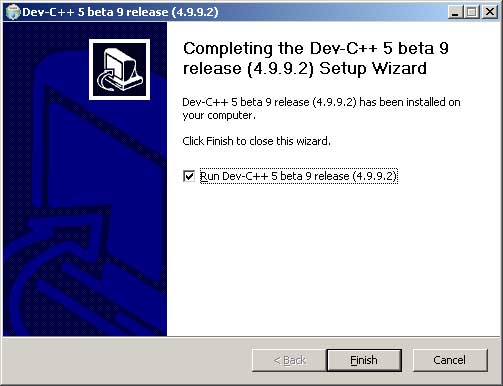


Many subdirectories and files are extracted to the destination:

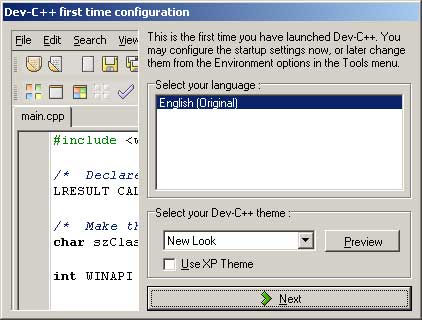


Answer "yes" when it asks if you wish to install Dev-cpp for all users. **Note:** if the installation fails, re-install and try "no" for this.

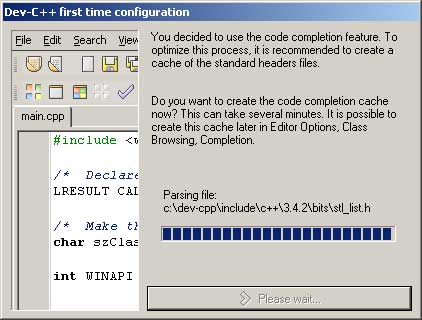
A screen says the installation is complete:



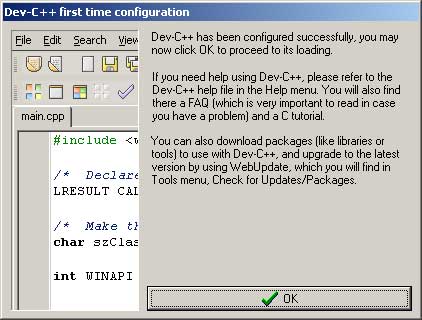
Keep the check mark in the box. Click on "Finish". A first-time configuration screen appears:



Pick "English" and "New Look". In the next several screens, hit "Yes" for its suggestions.



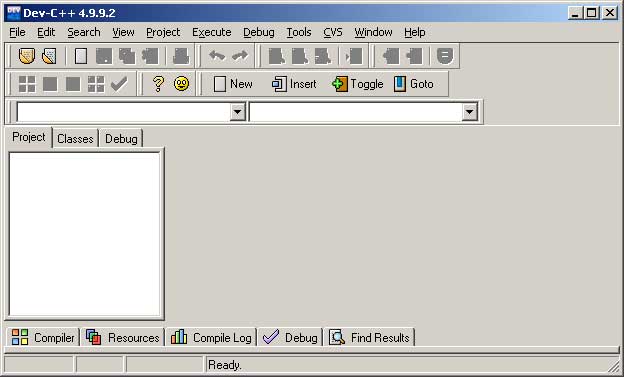
Eventually you are done. Click "OK".



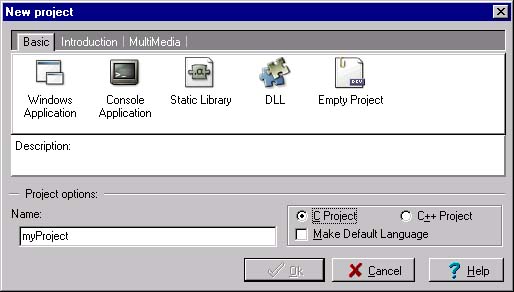
**II. DEV-C++ starts up. Try out the installation with a simple C program.**

Details:

The program starts up automatically.



Click File/New/Project. Pick a name for the project (such as "myProject"). Click "C Project". Click on "Empty Project". Click "OK".

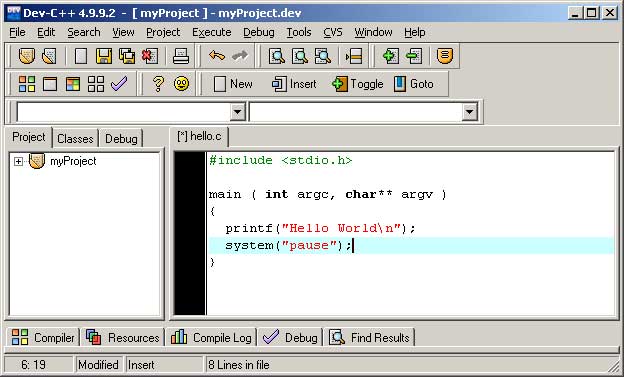


In "Create New Project", click "save" (later on you will probably want to create separate subdirectories for your various projects.).



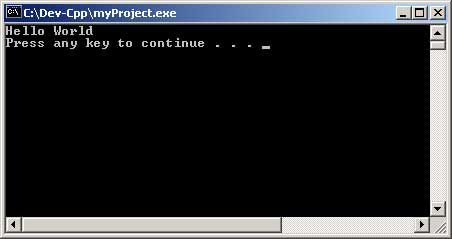
Click "File/New/Source File" and in "Add source file to current project" click "Yes". You now get a screen where you can edit the source file.

Type in a simple C program, as below. Now click "File/Save As" and save the file as "hello.c" (or other name.) Important: be sure that the file extension is **.c**. With any other extension (such as the suggested .cpp) you will have problems compiling.



Now click "Execute/Compile and Run"

The program will (hopefully) compile, run, and write its output to a DOS window. If you have the system("pause") statement in your program, the output will stay in the window until you hit a key. Another way to run the program (after it has been compiled) is to start a DOS window outside of the Dev-Cpp system, then navigate to the subdirectory that holds your project, and type hello.exe.



At this point, the compiler and development environment has been installed. You should find Dev-C++ listed under "Programs" on the "Start" menu and will now be able to write, compile, and run C (and C++) programs. You will have include files, libraries, and dll's for OpenGL (and all other standard packages) but not GLUT. GLUT manages the windows and other user interface components needed for OpenGL programming, and needs to be separately installed.

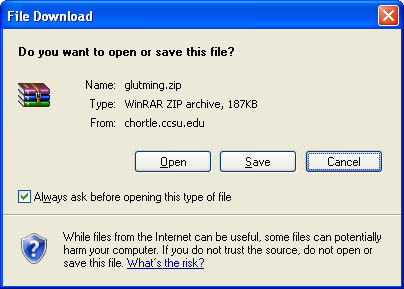
If you do not need GLUT , you can quit now.

**III. Download and install GLUT**

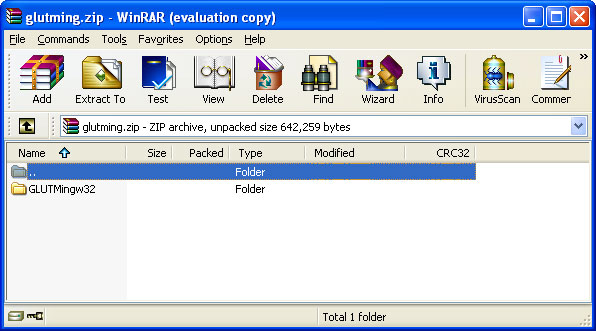
To run OpenGL with GLUT (which is what the programs in Angel's book use), you need to get three files and place each file in its proper directory. All the files you need (and more) are contained in one zip file.

Details:

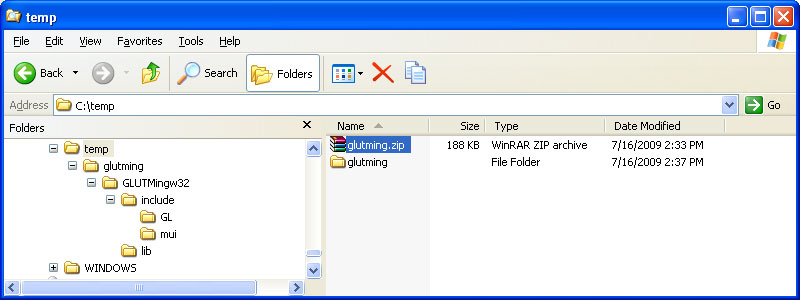
Download GLUT files from <http://chortle.ccsu.edu/Bloodshed/glutming.zip> Download the file glutming.zip Save the zip file in some convenient location (perhaps C:\temp).



Double click on glutming.zip (or otherwise unzip it). You will see the files that are in the zip archive. (Your un-zipping program will probably be diferent than the one shown here, but should work about the same.)



Click on "Extract" to extract all the subdirectories and files. Pick some convenient directory to extract them to (perhaps C:\temp\glutming). You only need three files, but extract all of them anyway.

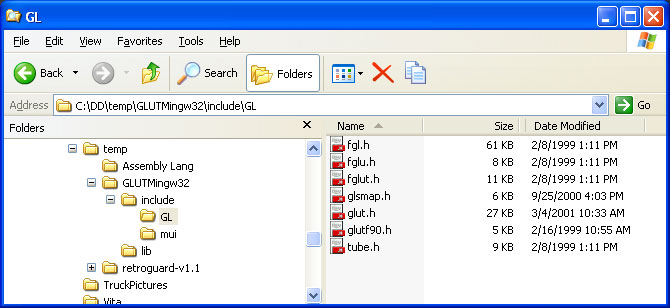


Only three of the files in the various subdirectories are needed. Each of the three files should be put in a subdirectory with other files of its type. Use Explorer to move the files to where they are needed.

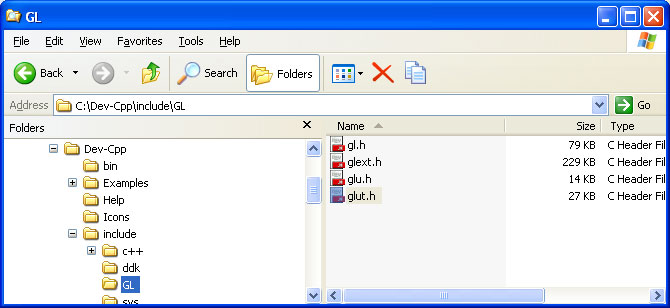
**Note:** If you only see some of these files listed in Explorer, click on "View/Options/View" and then select the radio button "Show all Files".

**glut.h** -- copy this file to C:\Dev-Cpp\include\GL

Copy from your "unzipped" subdirectories (wherever they are):



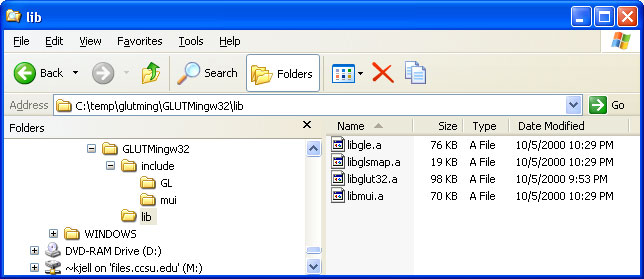
To here:



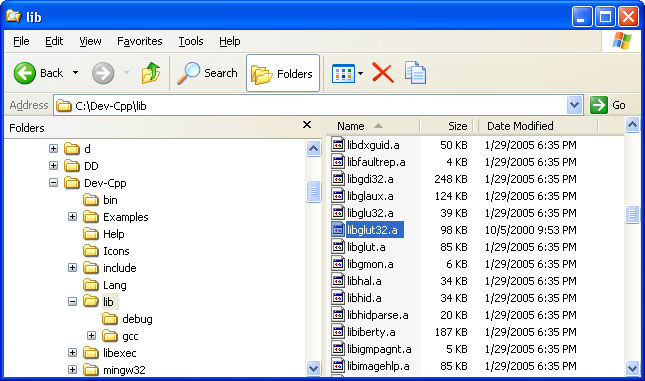
**libglut32.a** -- copy this file from your unzipped directories to C:\Dev-Cpp\lib

There may be a newer version of this file there, already. **Replace that version** with the one you unzipped (if you keep the newer version your programs will not link correctly.)

Copy from your "unzipped" subdirectories:

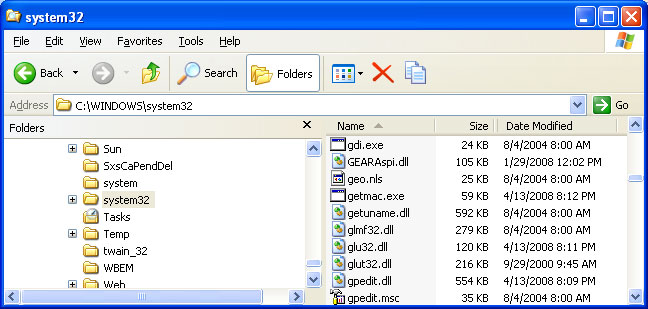


To here:



**glut32.dll** -- move this file to C:\WINNT\System32, or similar location.

The location for this file depends on your operating system. The directory where it goes is the directory that holds the dynamic load libraries (\*.dll). An easy way to find where it should go is to look for glu32.dll (use "Search" from the start menu).



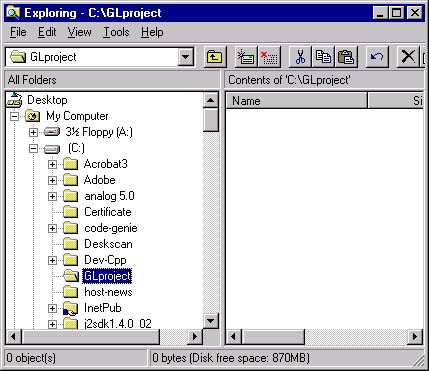
The directory to use should also have the files glu32.dll and opengl32.dll. These should have come with your operating system.

**IV. Test Dev-cpp with GLUT**

The essential step in compiling and running a C program that contains OpenGL and GLUT functions is to tell the linker where the libraries are. This is done by clicking Project/Project Options/Parameters/Add Library or Options and then navigating to the libraries you need to include: **libopengl32.a**, **libglu32.a**, and **libglut32.a**. The libraries should be added in that order.

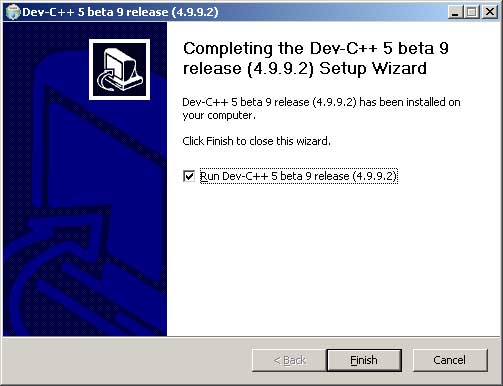
Details:

a. Create a subdirectory for a project. Do this first, before you start Dev-Cpp. Create a new subdirectory with "Explorer" by clicking "File/New/Folder".



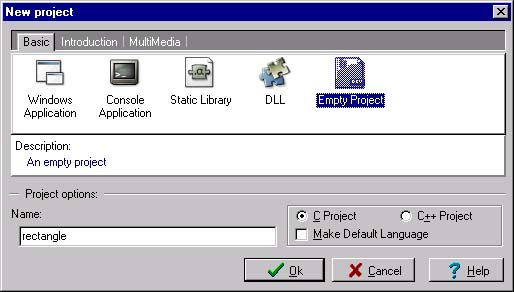
For example, create a folder C:\GLproject.

b. Start Dev-cpp:

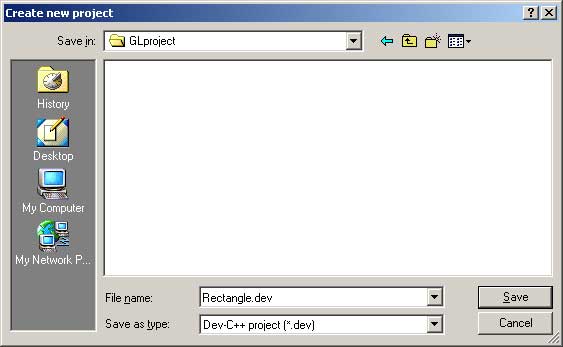


c. Start a new project by clicking File/New/Project. In the panel that pops up, name the project something like "rectangle", click on "empty project" and "C": Click OK.

Note: For compiling with OpenGL you must create a project. You need to have a project (not just a single C file) in order to link in the OpenGL libraries.

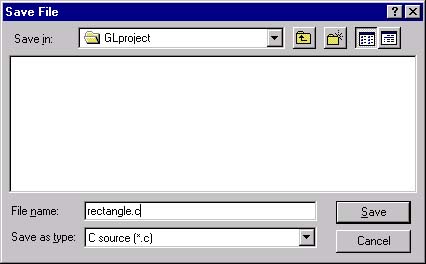


d. In the next panel, navigate to your folder C:\GLproject, and click "Save".



e. In Dev-C++, click "File/New/Source File" and then in the next panel "Add to Project" click "yes". Click "File/Save As" and then give the file a name. Navigate to your project subdirectory to save the file in it. Name the file something like "rectangle.c"

Be sure that the file names ends with ".c" anything else will cause big problems.



f. Click and drag your mouse over the following program so that it is highlighted, then click "Edit/Copy" from the browser's menu bar.

#include <GL/glut.h>

const int A = 500; /\* length of a side of the monitor window \*/

const float B = 500; /\* length of a side of the clipping rectangle \*/

const float C = 200; /\* length of a side of the square the program draws \*/

void myinit(void)

{

glClearColor(0.7, 0.7, 0.7, 0.0); /\* gray background \*/

glMatrixMode(GL\_PROJECTION); /\* In World coordinates: \*/

glLoadIdentity(); /\* position the "clipping rectangle" \*/

gluOrtho2D( -B/2, B/2, -B/2, B/2);/\* at -B/2, its right edge at +B/2, its bottom \*/

glMatrixMode(GL\_MODELVIEW); /\* edge at -B/2 and its top edge at +B/2 \*/

}

void display( void )

{

glClear(GL\_COLOR\_BUFFER\_BIT); /\* clear the window \*/

glMatrixMode(GL\_MODELVIEW); /\* The following coordinates are expressed \*/

glLoadIdentity(); /\* in terms of World coordinates \*/

glBegin(GL\_POLYGON) ; /\* draw a filled polygon \*/

glColor3f ( 1.0, 0.3, 0.2); /\* draw in light red \*/

glVertex2f( -C/2, -C/2 ); /\* (x,y) \*/

glVertex2f( C/2, -C/2 ); /\* (x,y) \*/

glVertex2f( C/2, C/2 ); /\* (x,y) \*/

glVertex2f( -C/2, C/2 ); /\* (x,y) \*/

glEnd();

glFlush(); /\* send all commands \*/

}

void main(int argc, char\*\* argv)

{

glutInit(&argc,argv);

glutInitWindowSize( A, A ); /\* A x A pixel screen window \*/

glutInitDisplayMode( GLUT\_RGB | GLUT\_SINGLE);

glutCreateWindow("My Rectangle"); /\* window title \*/

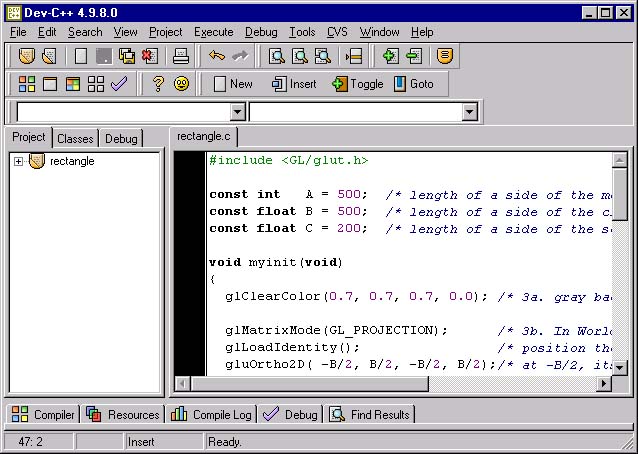
glutDisplayFunc(display); /\* tell OpenGL main loop what \*/

myinit(); /\* set attributes \*/

glutMainLoop(); /\* pass control to the main loop \*/

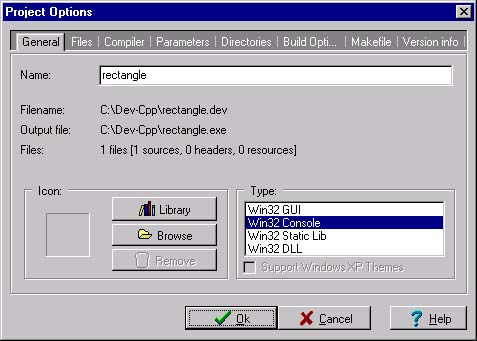
}

g. Now click in the editing window of Dev-cpp and then click "Edit/Paste" in its menu bar. The program will appear in the editing window.



h. Click "File/Save". The file in your project directory should now contain an OpenGL program.

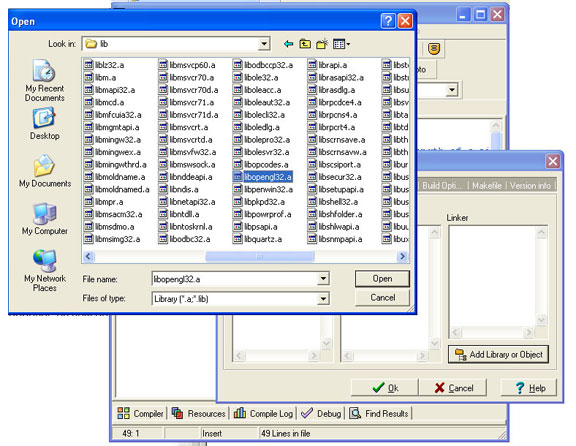
i. Tell Dev-cpp what libraries need to be linked. Click "Project/Project Options".



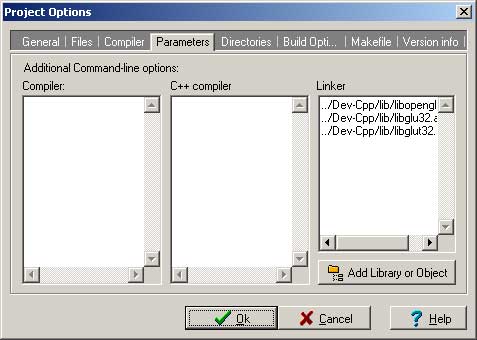
j. Now click "Parameters". Click the "Add Library or Object" button and navigate to the libraries that should be added, found under C:\Dev-cpp\lib

* ../lib/libopengl32.a
* ../lib/libglu32.a
* ../lib/libglut32.a

Add them in that order (only). Notice that the slashes will appear in Unix style "/" rather than DOS-style "\".



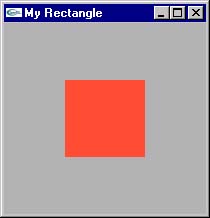
When you are done adding the three libaries, you should see:



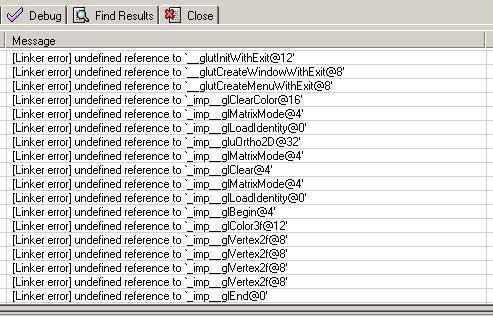
The exact pattern of "../../.." you see depends on how deep in the directory structure your source file lies.

Click "OK".

k. Click "Execute/Compile and Run". The program should compile, link, and run:



If things don't work (very common) click on the "Compile Log" tab for some confusing error messages. If you see something like the following, it means that you made a mistake in adding the libraries to the project:



Try to fix the list of libraries, or perhaps start over from scratch.

You now are finished, or have given up.