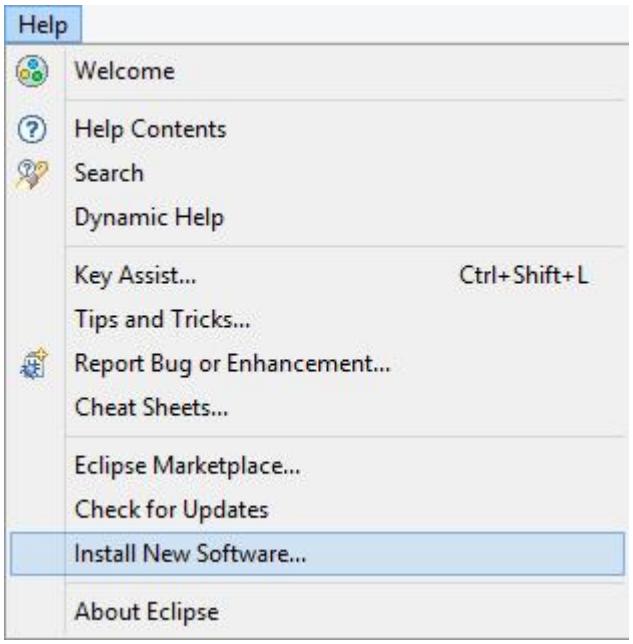
Step 1 Install Eclipse

If Eclipse is not already installed on the machine, then get the latest version of the Eclipse IDE for C/C++ Developers from the Eclipse downloads page (http://www.eclipse.org/downloads/).

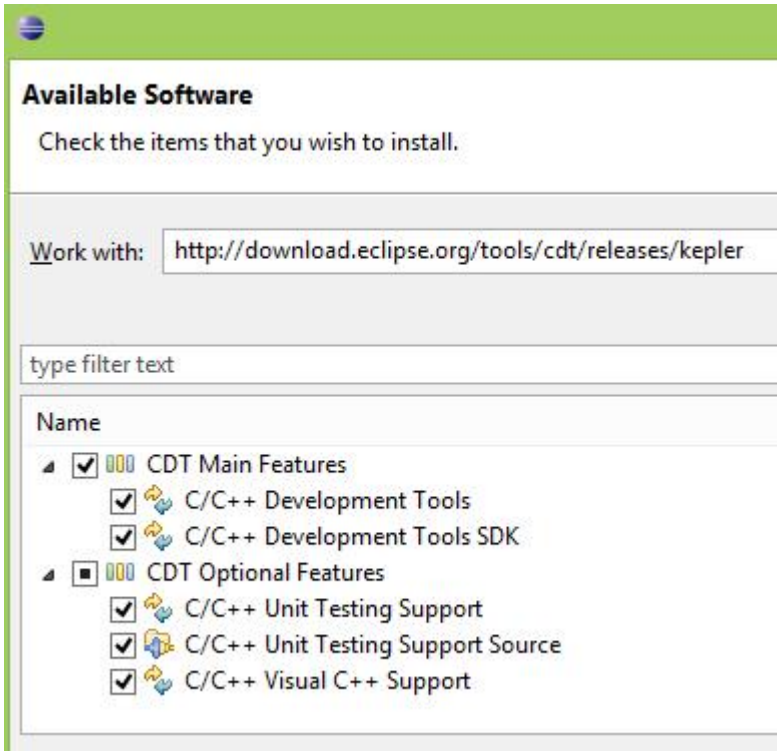
If Eclipse is already installed but only for Java, download the C++ plug-in following these instructions.

a. Open Eclipse and click on Help->Install New Software



b. In the Work with: box, type in http://download.eclipse.org/tools/cdt/releases/kepler. After a few moments, the Name box will populate. Select the following components:

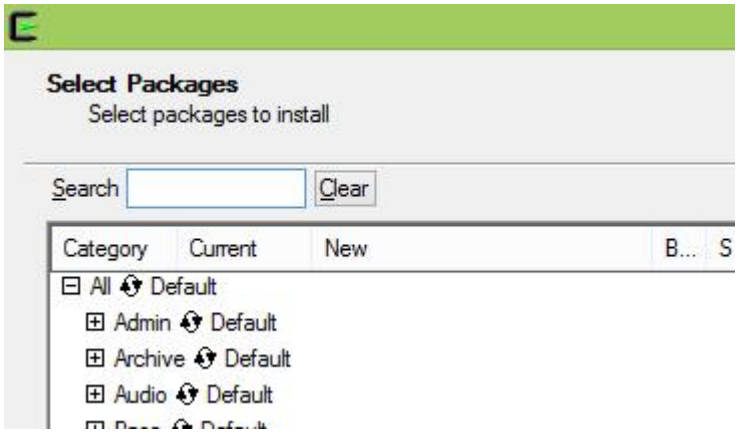
- CDT Main Features -> C/C++ Development Tools
- CDT Main Features -> C/C++ Development Tools SDK
- CDT Optional Features -> C/C++ Unit Testing Support
- CDT Optional Features -> C/C++ Unit Testing Support Source
- CDT Optional Features -> C/C++ Visual C++ Support



c. Click on Next, agree to the statements, and then Finish.

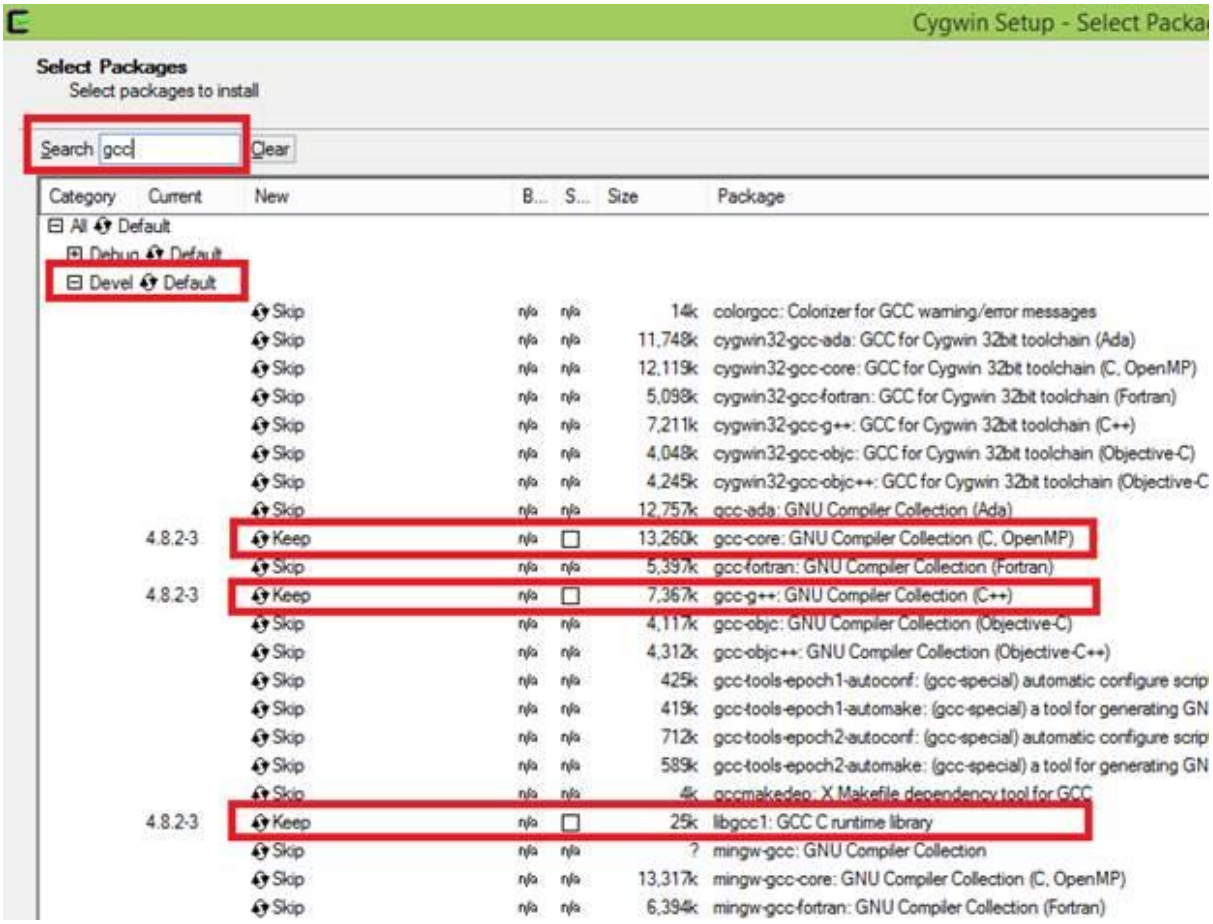
Step 2 Download Cygwin

Install Cygwin by clicking on the setup-x86_64.exe link on the Cygwin install page (http://www.cygwin.com/install.html). After running, click Next through the defaults until you get to the Select Packages window.

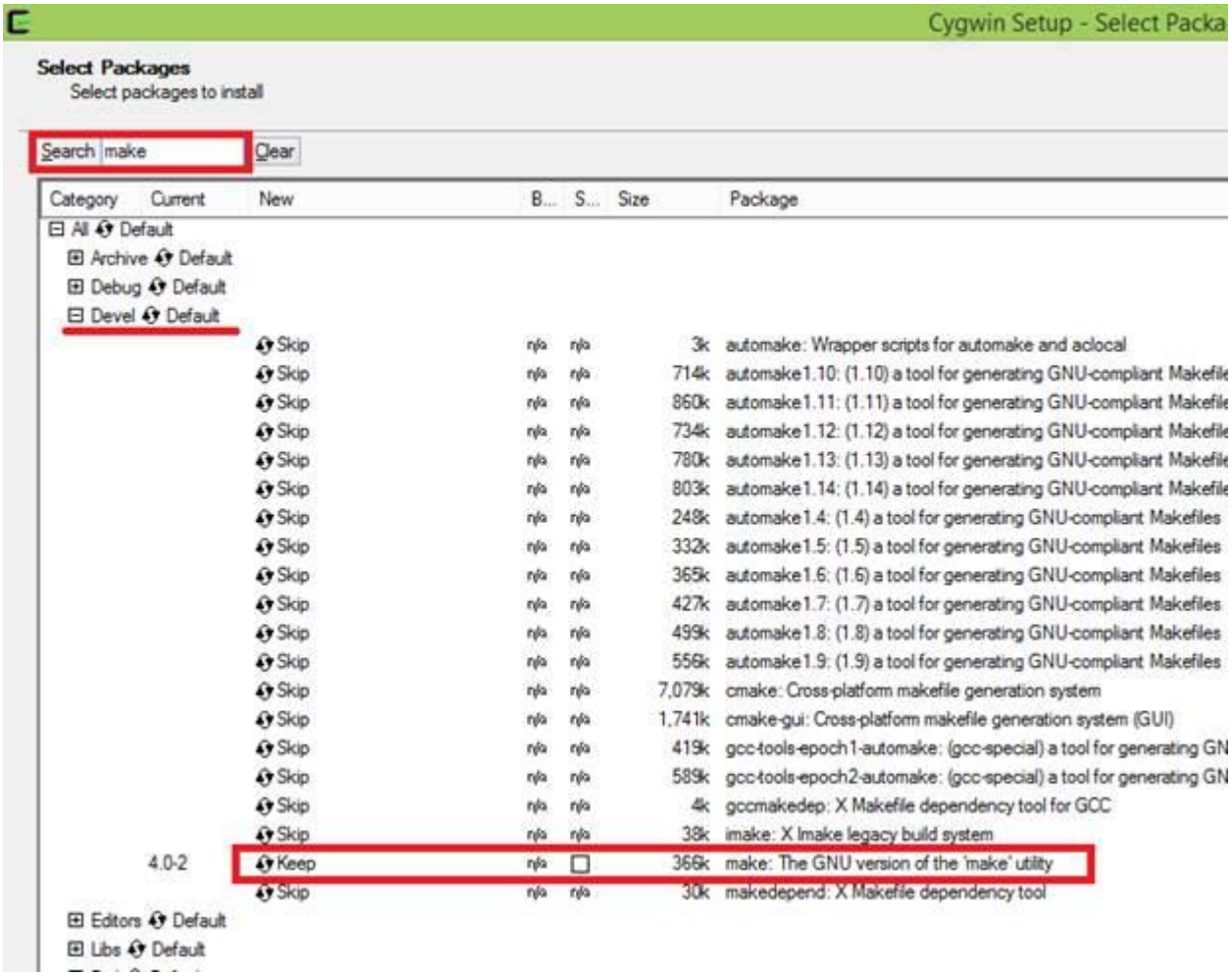


You will need to search for and install two packages: gcc and make.

The first search term is gcc. Search for gcc and then open the Devel folder. Mark the following packages for install by clicking on the word Skip (it will then change to the build number, which may by higher than the one depicted here): gcc-core, gcc-g++, and libgcc1.



The second search term is make. Here, we will only need the Devel package make.



Once these have been selected, click Next to install.

Step 3 Download and build Google Test project

Download the latest release of GoogleTest from https://code.google.com/p/googletest/downloads/list, and extract the zip file contents into a common directory. It is important that all users are able to access this directory.

(Note: the following commands use *make* -- the last revision of GoogleTest that uses *make* is https://github.com/google/googletest/releases/tag/release-1.8.1. For future revisions of GoogleTest use *cmake* instead.)

To build the Google Test project:

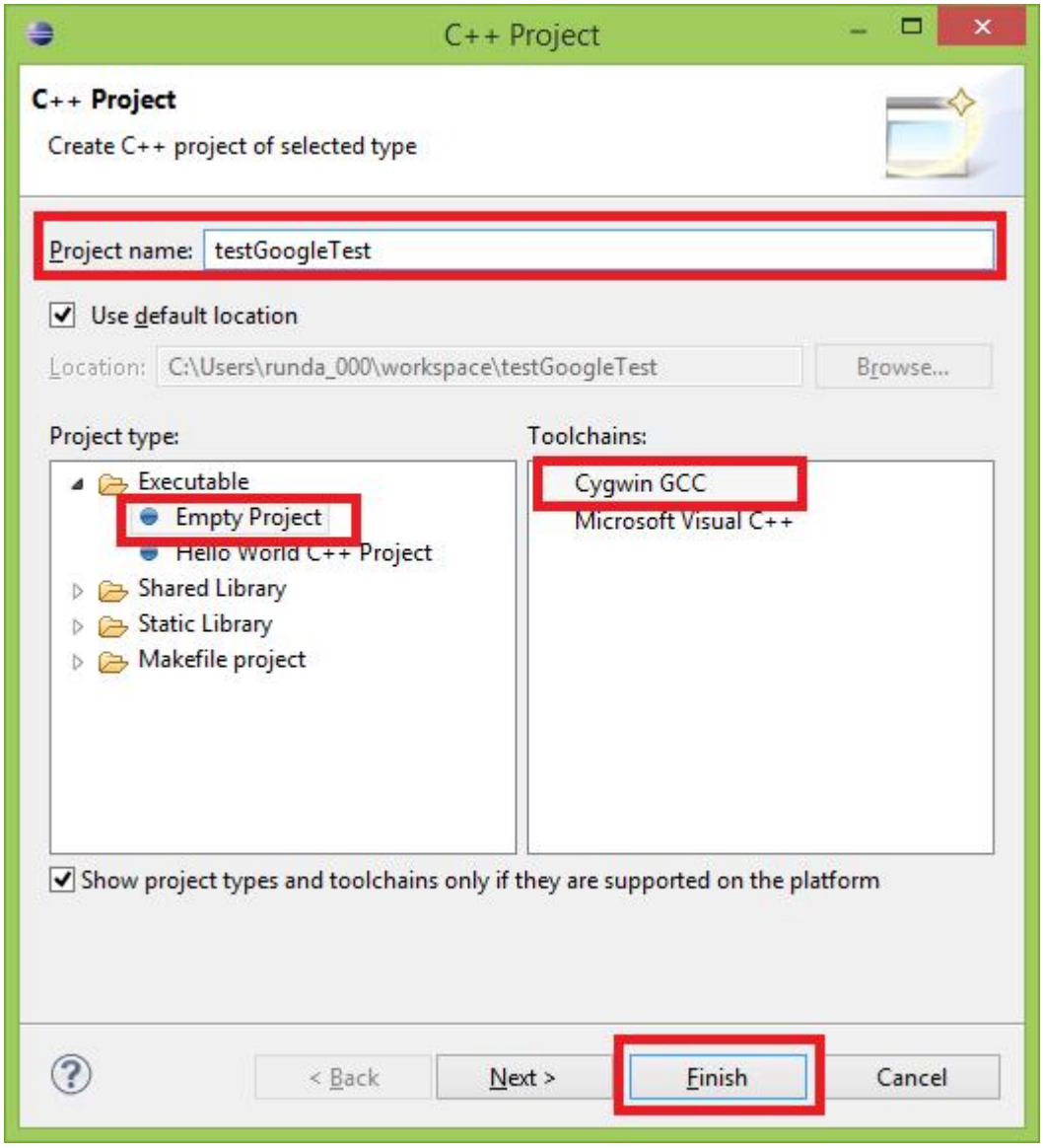
1. Open Cygwin (find the install directory for Cygwin and double-click on Cygwin.bat).
2. Change the current working directory to the unzipped GoogleTest make directory: *cd c:/<<yourpath>>/gtest-1.7.0/make/*
3. Build the project: *make*
4. Create an archived library out of the gtest-all.o file: *ar -rv libgtest.a gtest-all.o*

Step 4 Add the Cygwin bin directory to the computers PATH variable

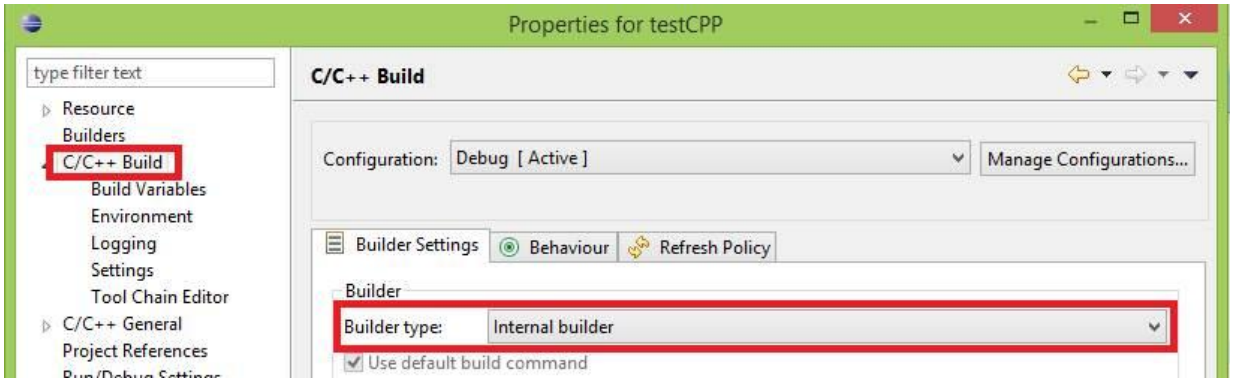
Follow the instructions on this page for your version of
Windows: http://www.java.com/en/download/help/path.xml, to add Cygwins bin directory to the computers PATH environment variable. (typically by adding ;C:\cygwin64\bin to the end of the current value).

Step 5 Create a new project that uses GoogleTest

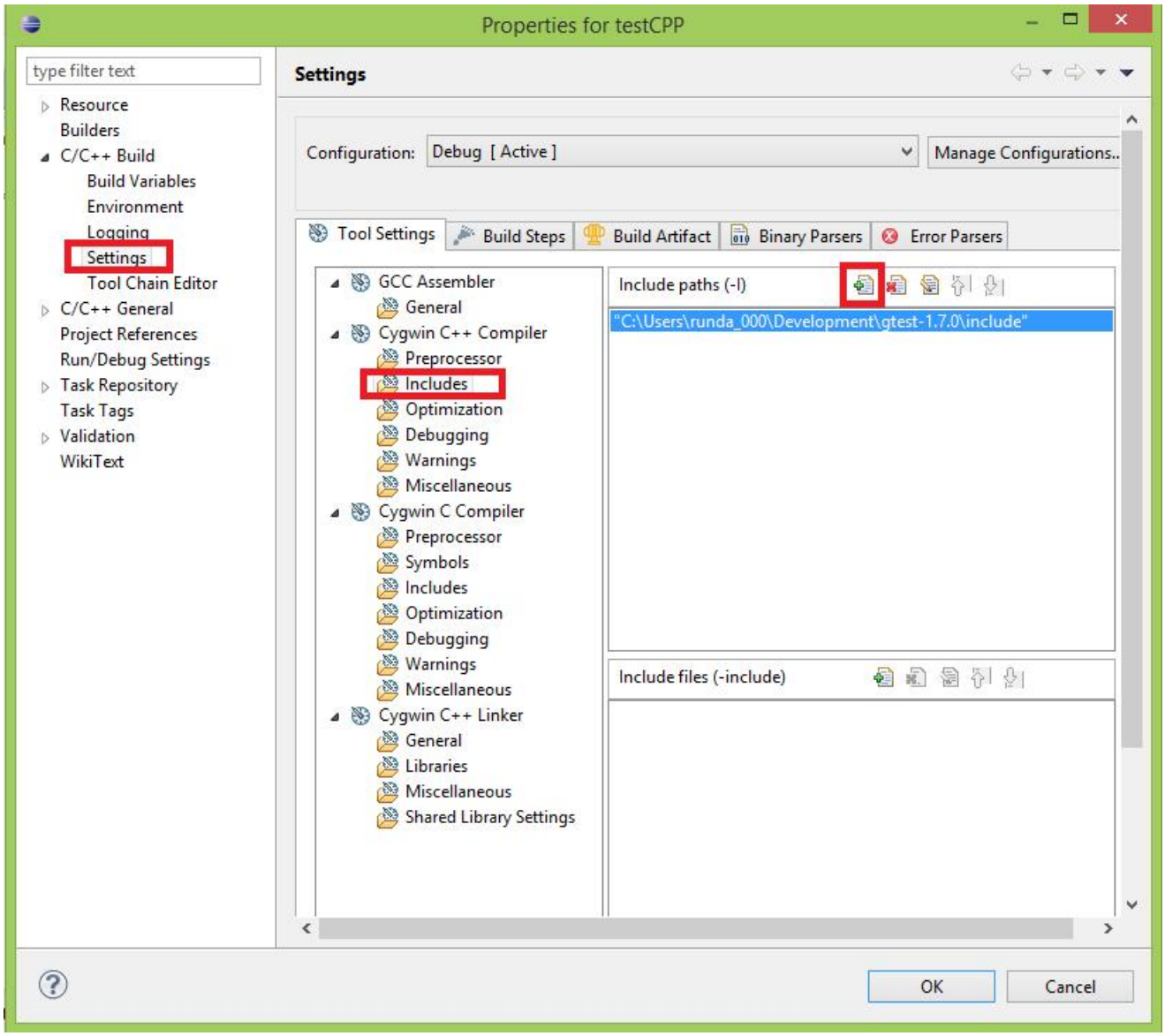
Start Eclipse and select File->New->C++ Project. Enter the values below and click Finish.



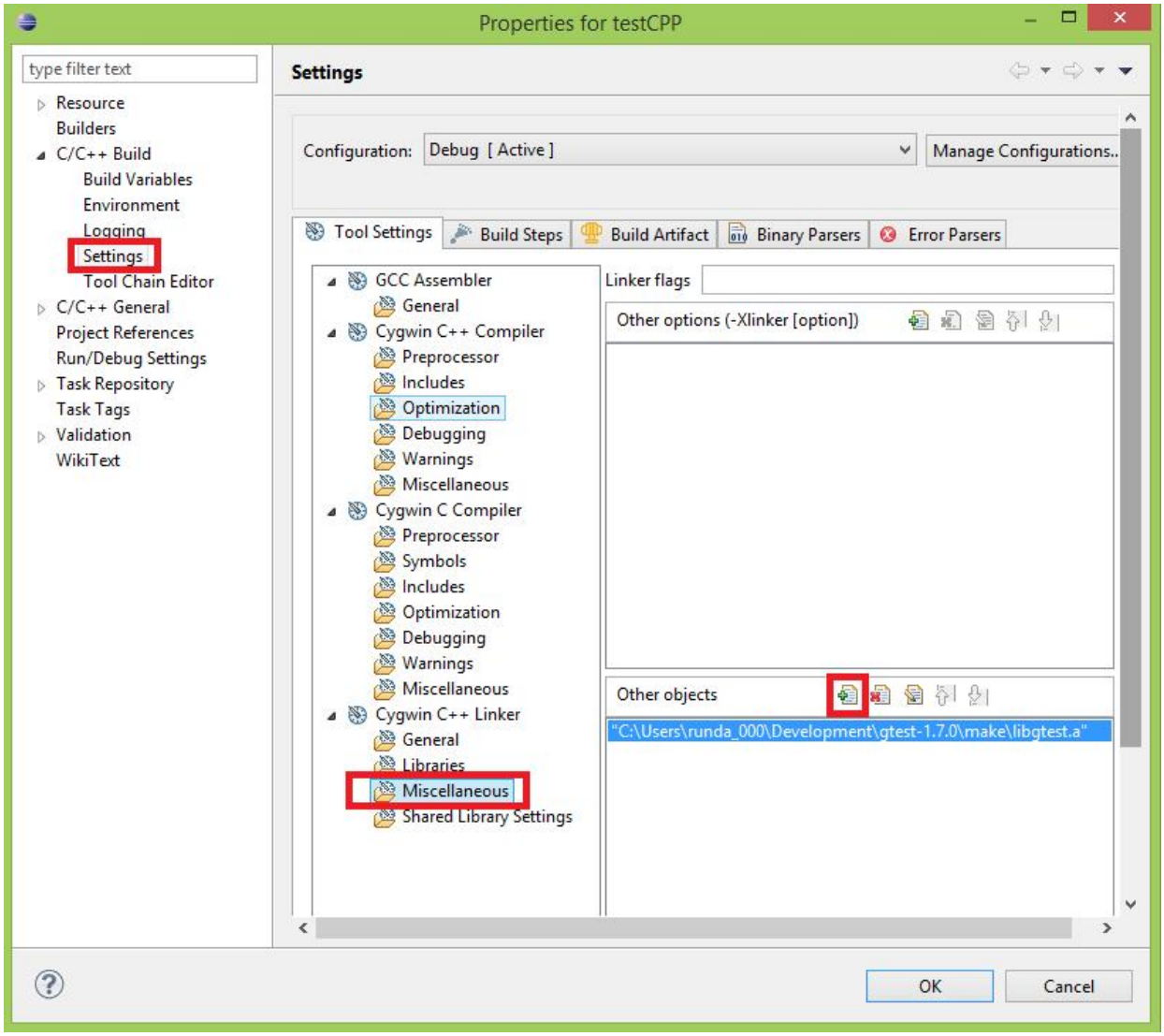
In the Project Explore, right-click the name of the project and select Properties. Under C/C++ Build, change the Builder type to Internal Builder.



Under C/C++ Build, select Settings, then click on the Includes folder under Cygwin C++ Compiler. Click on the Add button in the top box and then browse to the GoogleTest include folder.



Lastly, under the Cygwin C++ Linker folder, select Miscellaneous and then click the Add icon under Other objects. Find the libgtest.a file that you built back in step 3 (it should be in the make directory of the unzipped gtest folder).



Thats it! Now you're ready to try it out.

Step 6 Write some code that uses GoogleTest

- Add a source folder by clicking File->New->Source folder. Call it src.
- Add a header file by right-clicking on the src folder and select New->Header File. Call this file Counter.h.
- Add a source file by right-clicking on the src folder and select New->Source File. Call this file Counter.cpp.
- Add another source file and call it Counter_Tests.cpp.

Copy and paste the code below into the appropriate files:

Counter.h

```
class Counter { private:
        int mCounter;public:
        Counter() : mCounter(0) {}
        int Increment();
};
```

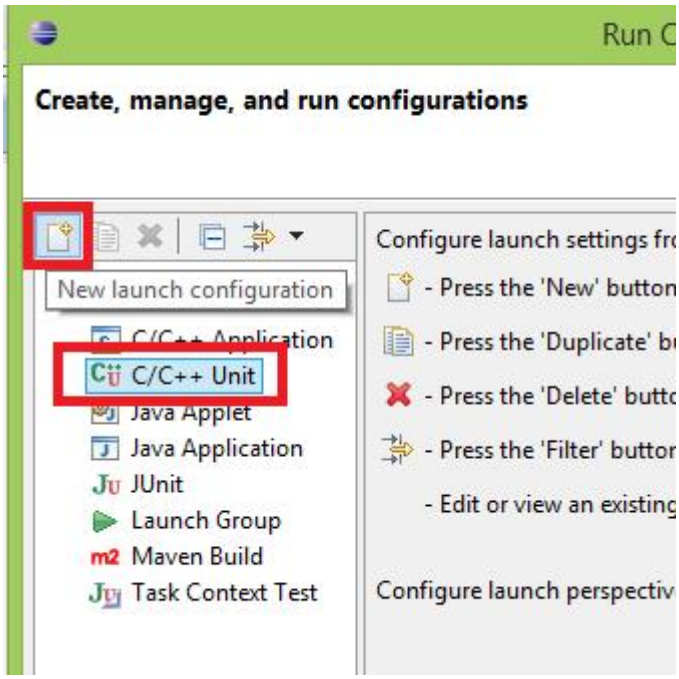
Counter.cpp

```
#include <stdio.h>#include "Counter.h"
int Counter::Increment() {
	return mCounter++;
}
```

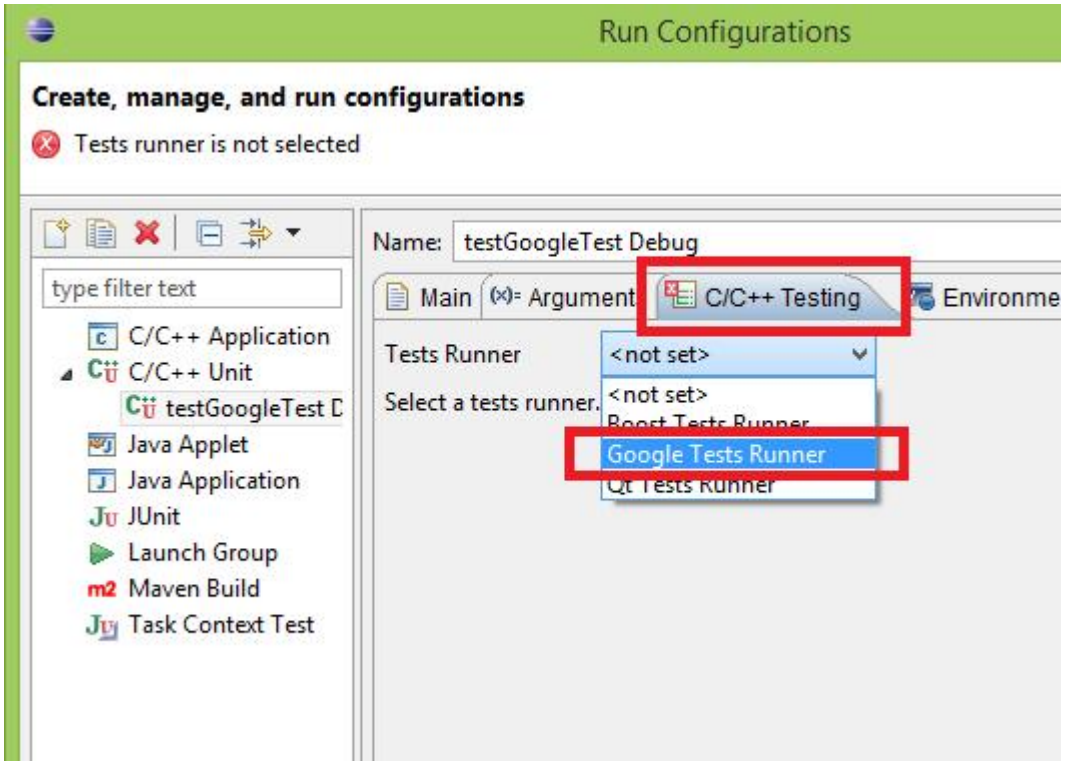
Counter_Tests.cpp

```
#include "gtest/gtest.h"#include "Counter.h"
TEST(Counter, Increment) {
	Counter c;
	EXPECT_EQ(0, c.Increment());
	EXPECT_EQ(1, c.Increment());
	EXPECT_EQ(2, c.Increment());
}
int main(int argc, char **argv) {
	::testing::InitGoogleTest(&argc, argv);
	return RUN_ALL_TESTS();
}
```

In the Project menu select Build All. Now, to connect up the GoogleTest unit testing framework, select Run Configurations from the Run menu. From this dialog, select C/C++ Unit and click the New button.



It should fill in this project name automatically under C/C++ Application, if not click on Search Project to select this project. Next, click on the C/C++ Testing tab. In the Tests Runner drop-down, choose Google Tests Runner, and then click Run to watch the magic!



Below is a snapshot of the result. After writing more code/tests, you can click on the button highlighted in red to quickly recompile and re-run all of the tests.