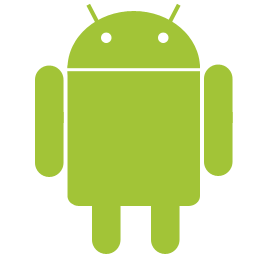
The Complete and Total Guide to starting Eclipse!

Compiled specifically for Mr. Kummer’s CP III and APCS

What is Eclipse and the Eclipse Foundation? ([eclipse.org/org](http://www.eclipse.org/org))

Eclipse is an open source community, whose projects are focused on building an open development platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing software across the lifecycle. The Eclipse Foundation is a not-for-profit, member supported corporation that hosts the [Eclipse projects](http://www.eclipse.org/projects) and helps cultivate both an open source community and an ecosystem of complementary products and services.

The Eclipse Project was originally created by IBM in November 2001 and supported by a consortium of software vendors. The Eclipse Foundation was created in January 2004 as an independent not-for-profit corporation to act as the steward of the Eclipse community. The independent not-for-profit corporation was created to allow a vendor neutral and open, transparent community to be established around Eclipse. Today, the Eclipse community consists of individuals and organizations from a cross section of the software industry.

The Eclipse Foundation is funded by annual dues from our [members](http://www.eclipse.org/membership) and governed by a[Board of Directors](http://www.eclipse.org/org/foundation/directors.php). Strategic Developers and Strategic Consumers hold seats on this Board, as do representatives elected by Add-in Providers and Open Source committers. The Foundation employs a full-time [professional staff](http://www.eclipse.org/org/foundation/staff.php) to provide services to the community but does not employ the open source developers, called committers, which actually work on the Eclipse projects. Eclipse committers are typically employed by organizations or are independent developers that volunteer their time to work on an open source project.

In general, the Eclipse Foundation provides four services to the Eclipse community: 1) [IT Infrastructure](http://www.eclipse.org/org/#IT), 2) [IP Management,](http://www.eclipse.org/org/#IP%20Management)3) [Development Process](http://www.eclipse.org/org/#Development), and 4) [Ecosystem Development](http://www.eclipse.org/org/#Ecosystem). Full-time staff are associated with each of these areas and work with the greater Eclipse community to assist in meeting the needs of the stakeholders.

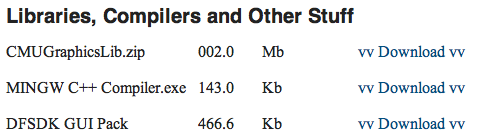
Get Eclipse 

Eclipse is an open source IDE, but it is also a java application making it useable on any of the three major commercial operating systems, Win, Mac, Linux. Eclipse comes in many different builds, specified for different developers. A complete list of their builds can be found [here](http://www.eclipse.org/downloads).

One amazing benefit to using Eclipse as opposed to Microsoft Visual Studio is that Eclipse comes in a Classic package that allows developers to code in multiple languages all in the same IDE. A downside to Classic however is that you must add the languages manually as plugins. Eclipse in general also doesn’t have it’s own compiler for some languages like C++ and these must be added manually as well, something Visual Studio comes standard with.

Anyways, back to getting Eclipse.

1. **Go to** [**Eclipse.org\downloads**](http://www.eclipse.org/downloads)
2. **Select your package.** Java comes standard on Classic. CPIII students can download Classic and add C++ as a plugin or just choose the C++ package.
3. [**Download the newest Java Virtual Machine.**](http://www.java.com) Even if you’re using the C++ build, Eclipse is a JAVA application and needs a virtual machine in order to operate. Note: Stay away from x64 builds of the JVM/JDK. 64 bit users should run x86 (32 bit) because the x64 JVM’s are unstable at the time of this writing. Apple users, you already have JAVA.
4. **Unzip Eclipse and move to your file system’s root directory.** Create a link on your desktop.
5. **Add C++.** If you downloaded Classic and are taking CPIII, you will need to add the C++ perspective and compiler to add to your IDE.
   1. For **Windows**: go to [stuartsoft.com/downloads](http://www.stuartsoft.com/downloads.htm) to retrieve the correct MINGW C++ compiler. Run the installer and check the following items...

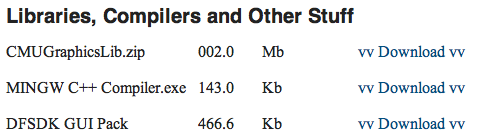


* 1. For **Mac**: Eclipse can leach off the Xcode Objective C compiler for C++ perspective. Log in or sign up free as an apple developer and download the newest Xcode build from [developer.apple.com](http://developer.apple.com). If you’re really lazy, Mr. Kummer should have an Xcode DVD you can borrow.
  2. Adding the C++ perspective is universal. **NOTICE: IF YOU DOWNLOADED THE C++ PACKAGE, SKIP THESE NEXT STEPS!!!** Open Eclipse and close all perspectives. Go to Help>Install New Software. Click add and in the dialog box enter “CDT” for name and <http://download.eclipse.org/tools/cdt/releases/helios> as the location. Check both items in the list press ok, agree to the lisence terms, bla, bla, bla.
  3. Once CDT plugin is installed, restart Eclipse and you should have a fully functional Java/C++ Eclipse environment. :D

Plugins and Local Libraries 

To increase the functionality of Eclipse, you can install what are called, “Plugins”. These allow you to use other Eclipse or 3rd party software within your Eclipse environment. This is absolutely necessary for phone development like the Android SDK. Plugins can be added/deleted under Help>Install New Software. This presents you with a dialog box that can be used to edit and add eclipse plugins.

Local libraries are another thing you’ll want to use in Eclipse. For CP III student’s, you’ll be using the Carnegie Mellon University Graphics Library in C++. These next steps will show you how to add CMU Graphics Lib to your projects.

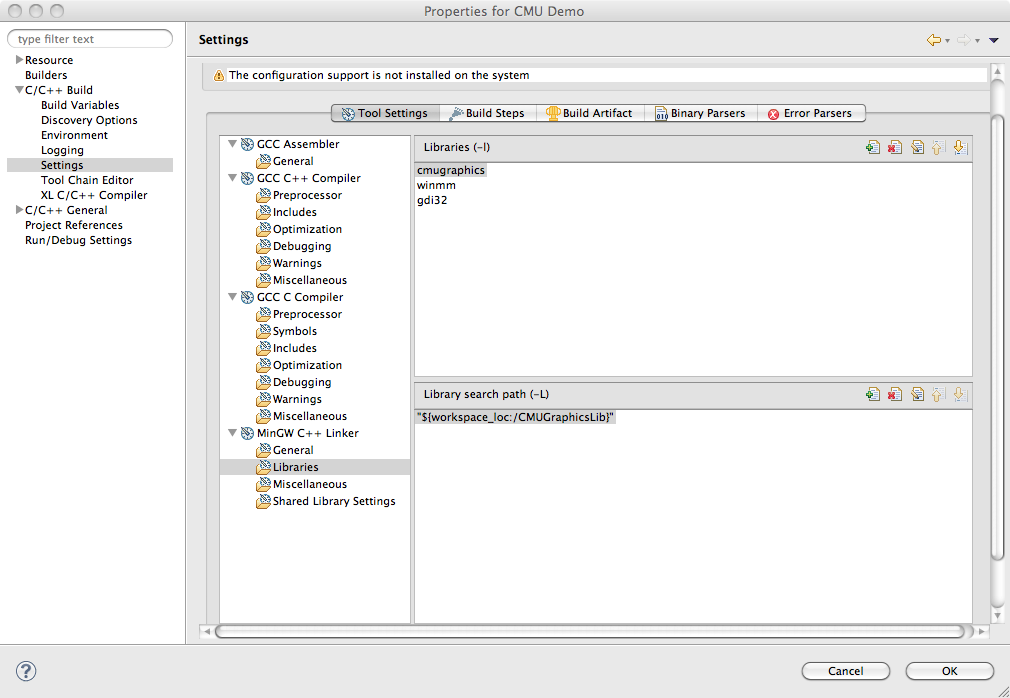
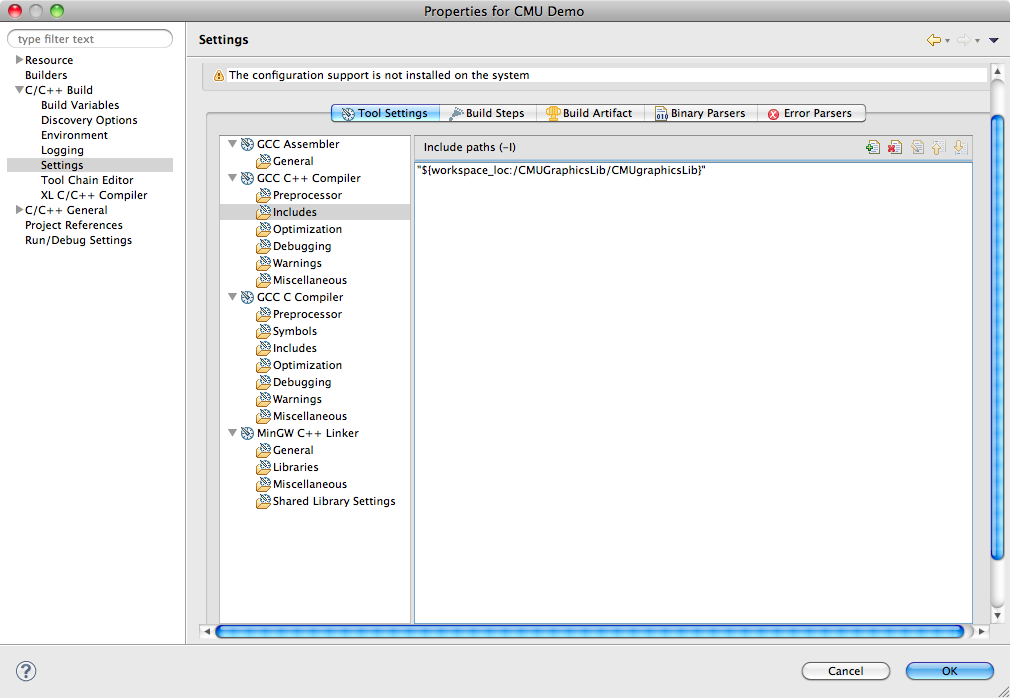
1. Go to [stuartsoft.com/downloads](http://www.stuartsoft.com/downloads.htm) and download the CMU Graphics Library.
2. Unzip the folder to your desktop.
3. Open Eclipse and drag the Library folder into your workspace.
4. Next, create a new C++ project.
5. Right click and select preferences
6. Go to C/C++ Build > GCC C++ Compiler > Includes. Click the add button and find the library folder inside the CMU folder you just added to your library."${workspace\_loc:/CMUGraphicsLib/CMUgraphicsLib}"
7. Next, go to MinGW C++ Compiler > Libraries and add the following to the library list

“cmugraphics”

“winmm”

“gdi32”

1. Under Library Search Path add your new library folder. "${workspace\_loc:/CMUGraphicsLib}"



1. You’re done! CMU Graphics Library should now be fully linked to your project. Simply add #include “cmugraphics.h” to your source for complete access to all the gui API’s.

AP Computer Science students will be using the Android SDK for development after AP testing. These next steps will show you how to add CMU Graphics Lib to your projects.

Hello World Apps

Starting a new language can be difficult, but as tradition dictates, when learning a new language, you start by saying, “Hello World”.

C++

JAVA

Common Problems

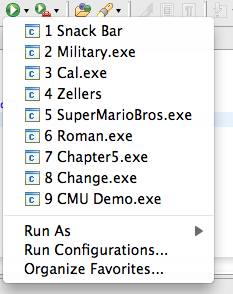
Although Eclipse is very powerful, it is also very picky. Here are a number of common problems we’ve experienced and how to solve them. If you’re having trouble with one of the items listed, try our methods first before bothering Mr. Kummer.

Make: \*\*\*

Eclipse was unable to either a) Find your compiler b) use the build settings specified in your project folder. If you haven’t changed any settings in your build configuration, it’s most likely the compiler. Sometimes re-installing the compiler may help. Also see “Unknown Option”.

Launch Failed. No binaries.

Eclipse cannot find your project binary (aka. the executable app for debugging) To solve this problem, first try building before running again. If you still have the same error, make sure Eclipse can find your app by clicking on the arrow next to the run button and selecting “run configurations...” From there you can give Eclipse the path of your app starting at your project path. Example, “debug\sandbox.exe”



Unknown Option: -o*ProjectName*

Caused by mismatch with compiler. You can change to the correct compiler to your platform by opening project preferences, clicking C/C++ Build > Tool Chain Editor. There you can select the correct tool chain for your platform. Win: MinGW GCC, Mac: MacOSX GCC. If the problem persists, click C/C++ Build > Settings and examine your build artifact. Artifact Extensions should be as follows: Win: “exe”, Mac: “”. If problems STILL persist, make sure your Binary Parsers are set correctly. Win: check PE Windows Parser, Mac: check Mach-O 64 Parser. If there’s still no luck, see the above solution on Binaries.

Cannot Open Output File *Project*: Permission Denied

This error is caused by rebuilding the debug app while it’s still running. Before rebuilding an app, you should always make sure the console says terminated at the top. If your app won’t build because of this error you have two options:

1. Win: Open task manager, select processes and kill your app. Mac/Linux: Force Quit manager may or may not work.
2. If all else fails, restart Eclipse.