

## Programming Cloud Services for Android: Setup Guide

Welcome to the course! In order to minimize confusion and maximize any troubleshooting efforts, we highly recommend that students use the course's standard development environment. This consists of a standard installation of Eclipse 4.4 packaged with the Gradle plug-in. Eclipse is the de-facto IDE for Java programming, while Gradle provides a powerful dependency management and building environment that makes it easy to utilize the frameworks and libraries we will cover in the course.

The source code for the examples and assignments will be hosted on a Github repository. If you are not already familiar with git, we highly recommend that you learn how to use it since it's practically everywhere these days. Github and Vogella have some decent tutorials located at:

<https://try.github.io>

<http://www.vogella.com/tutorials/Git/article.html>

Also, this visual reference is very helpful:

<http://marklodato.github.io/visual-git-guide/index-en.html>

This setup guide will cover several topics:

- [Downloading and Installing Eclipse](#)
- [Installing the Gradle Plug-in](#)
- [Downloading source code from the Github repository](#)
- [Importing Gradle projects into Eclipse](#)
- [Running Gradle projects from Eclipse](#)
- [\(Optional\) Installing the JDK to fix Gradle](#)

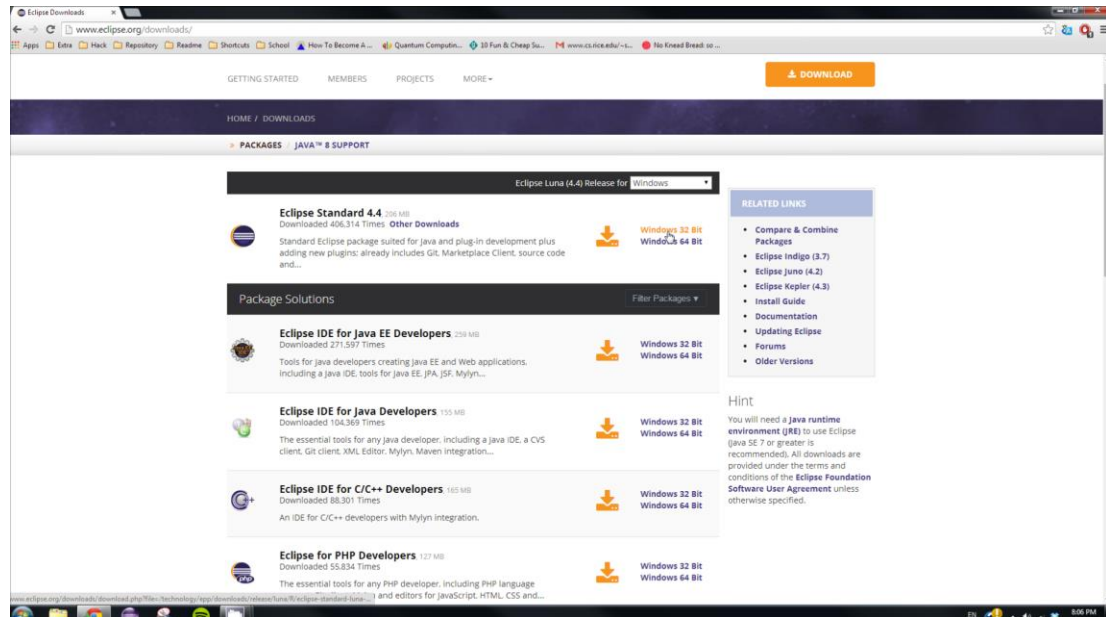
## Download and Installing Eclipse

(Windows)

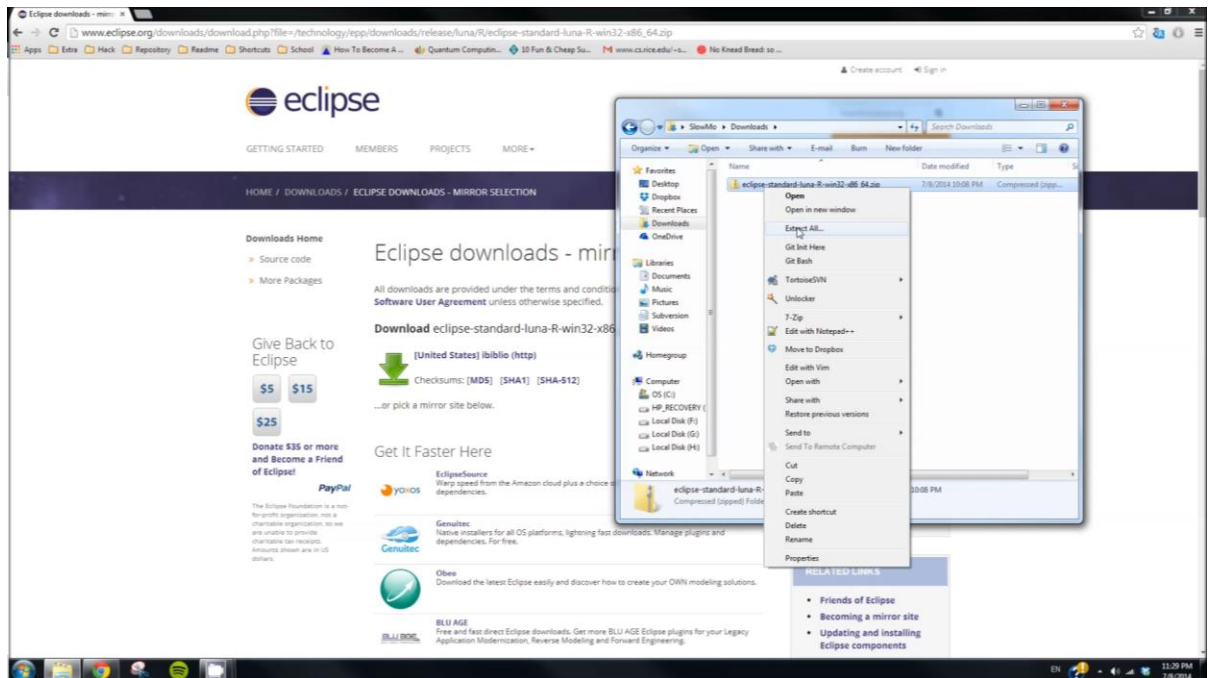
1. Open a web-browser and navigate to:

<http://www.eclipse.org/downloads>

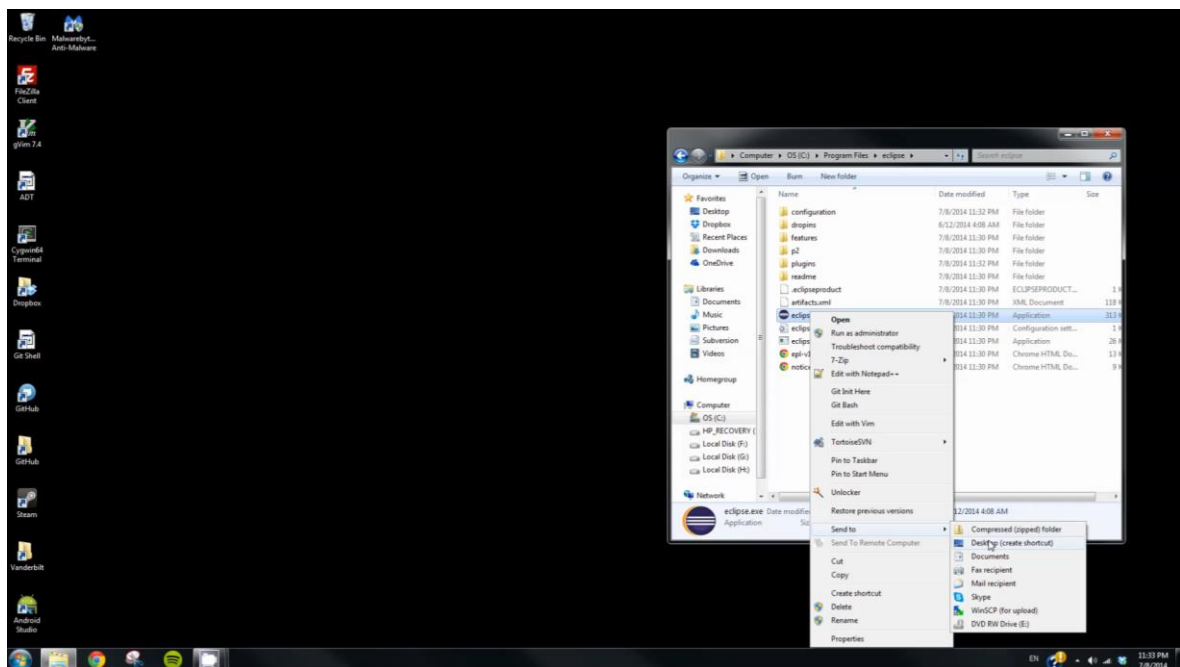
And download the Eclipse standard version 4.4. Select either 32-bit or 64-bit, depending on your OS version.



2. Extract the downloaded zip file. To extract, Right Click -> "Extract All". You should extract the files to wherever you want the Eclipse installation to live. For most Windows users, C:\Program Files should be sufficient.

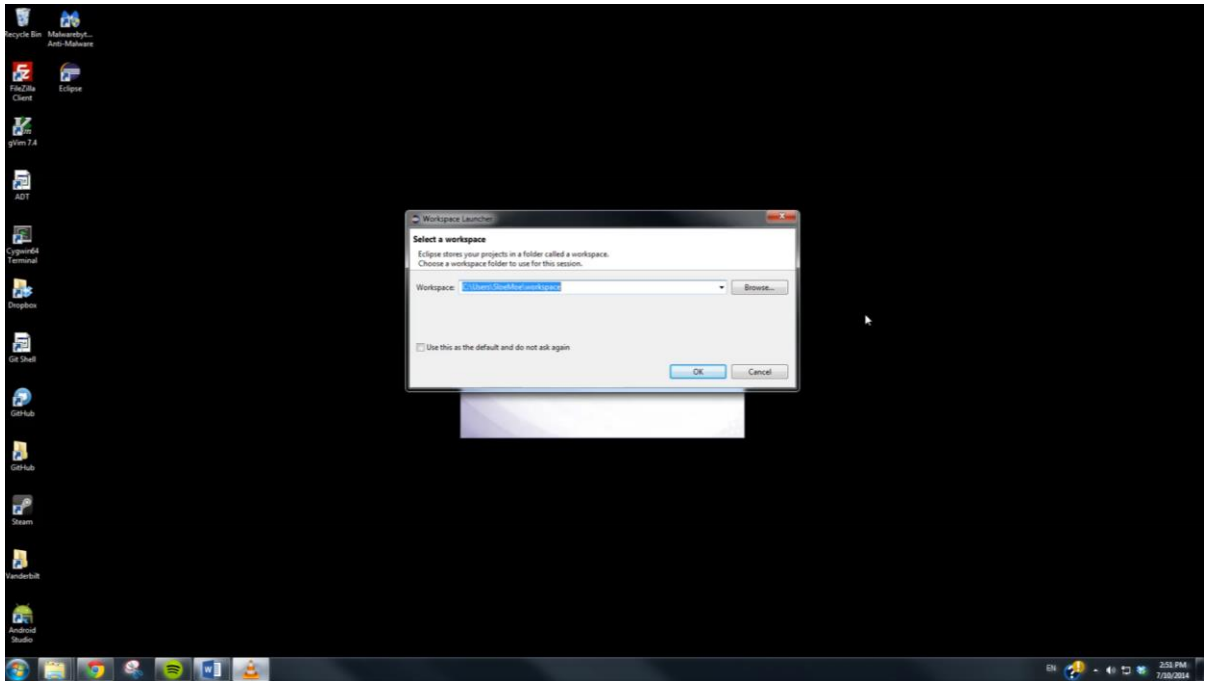


3. Start Eclipse. In the folder that you extracted the files to, there should be an executable called "eclipse.exe". Double-clicking that should start Eclipse. You may also want to create a shortcut on your Desktop by right-clicking eclipse.exe -> Send To -> Desktop.



4. Choose a workspace. This is where Eclipse will store metadata files and plug-in configuration data. The default location is normally fine. It is sometimes convenient to create different

workspaces for different groups of related projects, but that will not be necessary for this course.

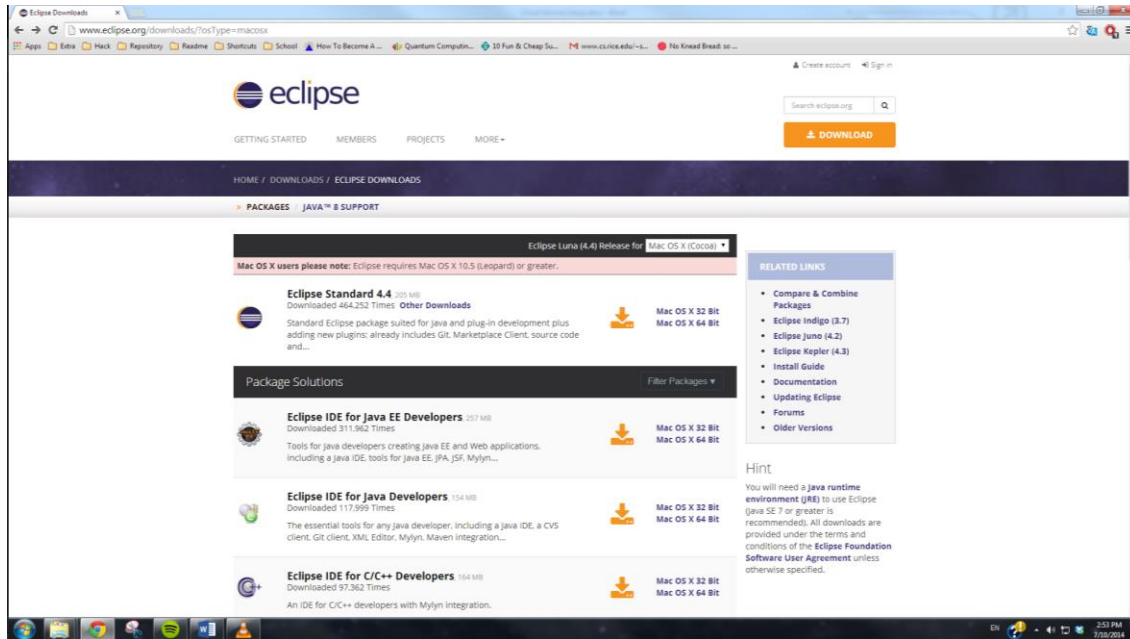


(Mac)

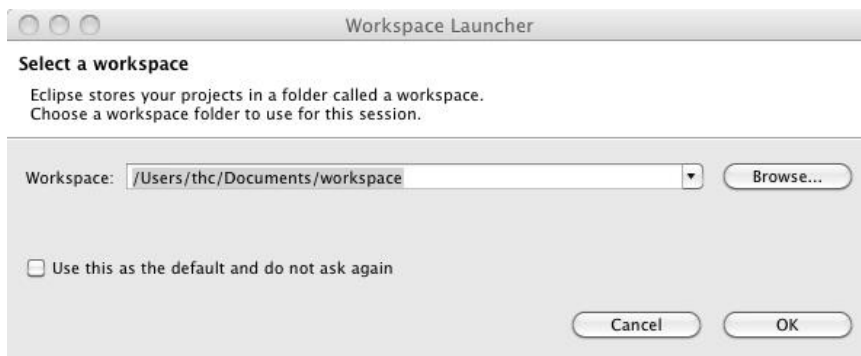
1. Open a web-browser and navigate to:

<http://www.eclipse.org/downloads>

And download the Eclipse standard version 4.4. Select either 32-bit or 64-bit, depending on your OS version.



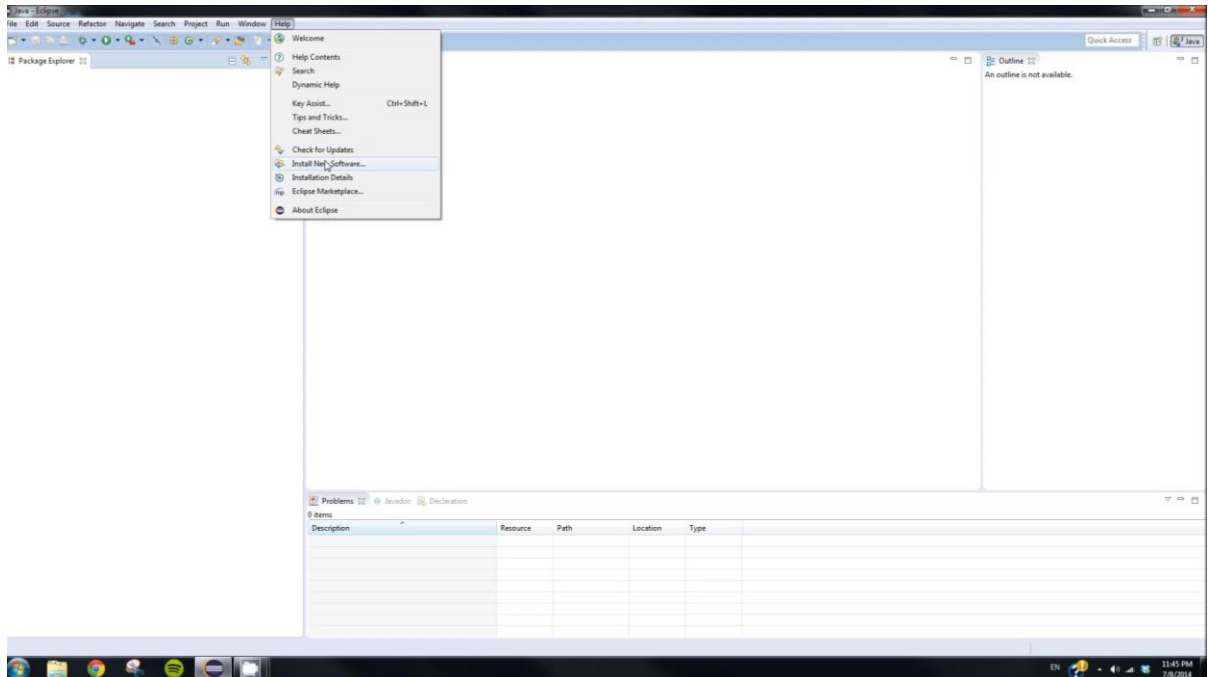
2. Open the .tar.gz Archive file. The download will go to your Downloads folder. You will normally have an icon for the Downloads folder on the right-hand side of the dock. Find eclipse-java-helios-SR2-macosx-cocoa.tar.gz in your Downloads folder, and drag it to the Desktop. Then double-click it. This will open the archive, revealing the "eclipse" folder.
3. Install Eclipse by dragging the "eclipse" folder into your Applications folder. The easiest way to do so is to open a new window in the Finder and click on Applications in the list you get on the left-hand side. Then drag the "eclipse" folder in with the other applications.
4. Add Eclipse to your dock for easy access. Double click the "eclipse" folder. You'll see an application named "Eclipse"; it has a purple icon with white horizontal stripes. Drag it into your dock. Now you will be able to launch Eclipse by clicking on the icon in the dock.
5. Choose a workspace. This is where Eclipse will store metadata files and plug-in configuration data. The default location is normally fine. It is sometimes convenient to create different workspaces for different groups of related projects, but that will not be necessary for this course.



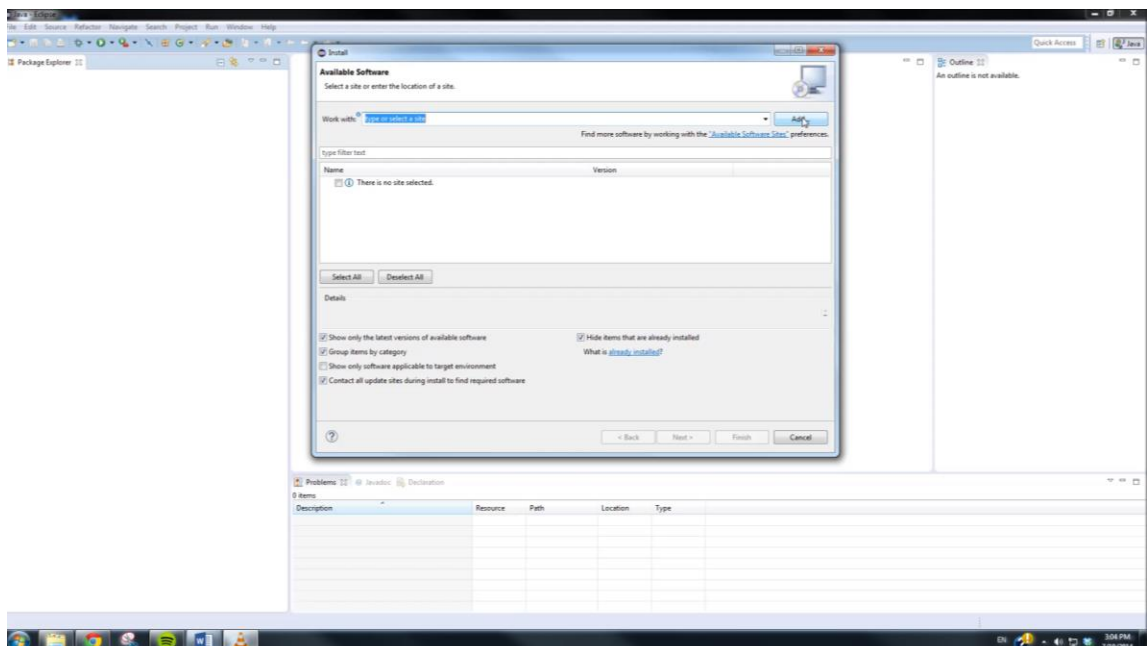
## Installing the Gradle Plug-in

(All OS)

1. At the top of the Eclipse window, select “Help” -> “Install New Software”

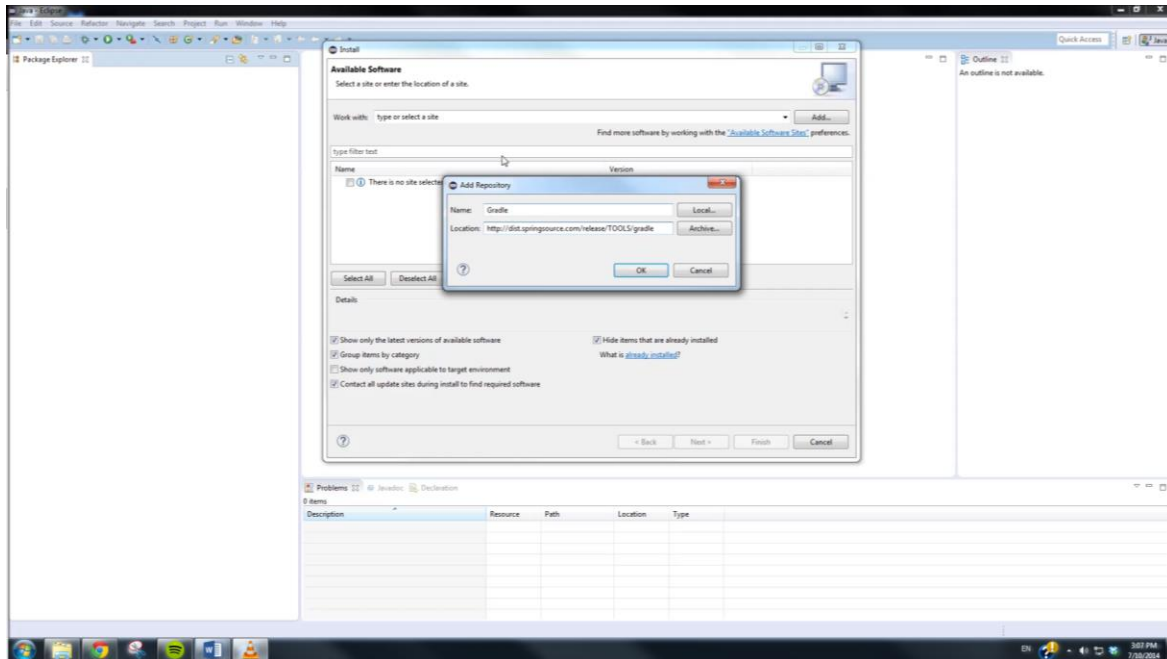


2. In the “Install New Software” dialog, click the “Add” button to add a new software site.

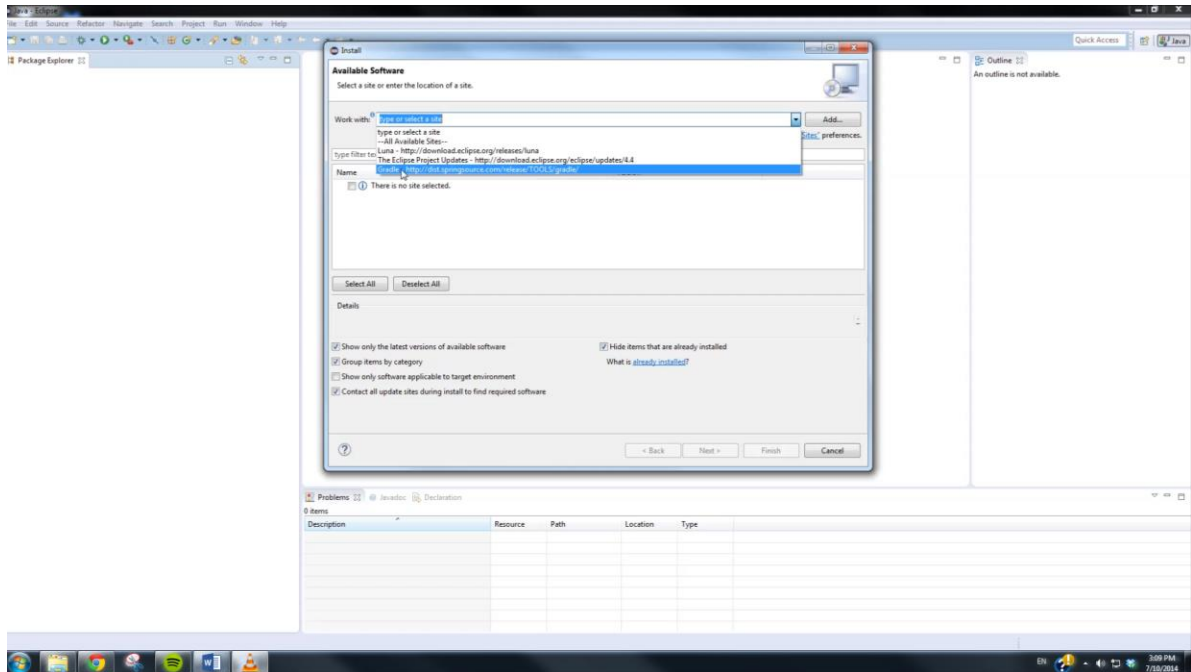


3. Enter a name for the repository. It can be something like “Gradle Repository,” but it doesn’t really matter.
4. Enter the location of the Gradle repository. At the time of this document’s creating, the Gradle plug-in is located at:

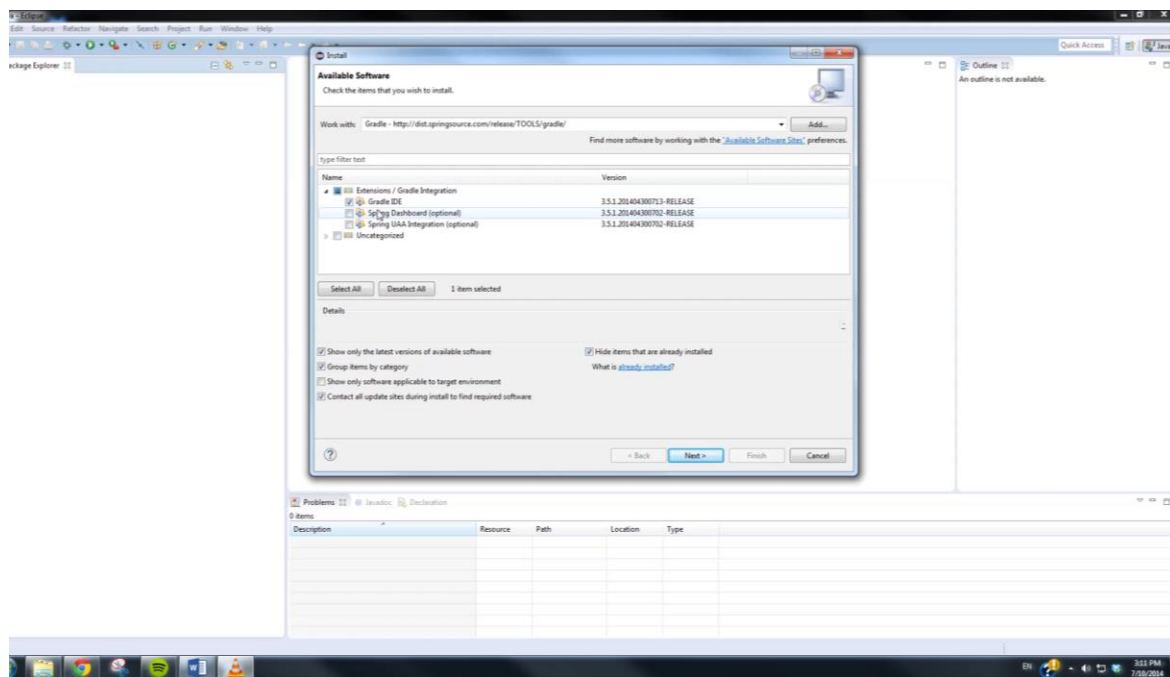
<http://dist.springsource.com/release/TOOLS/gradle>



5. Select the Gradle repository from the repository list.



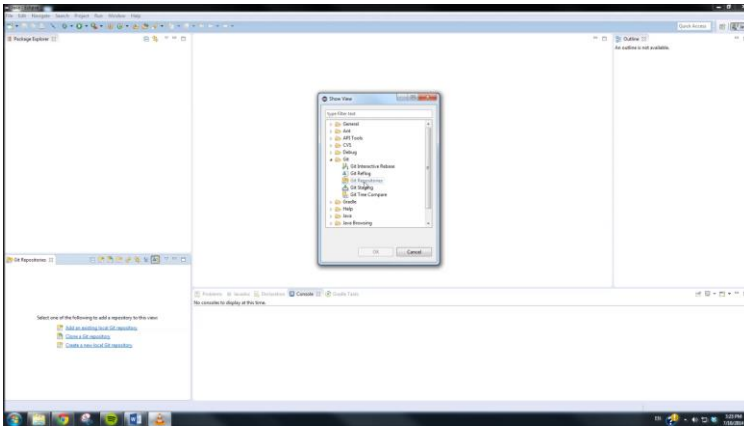
6. Check the box next to “Extensions / Gradle Integration” -> “Gradle IDE”



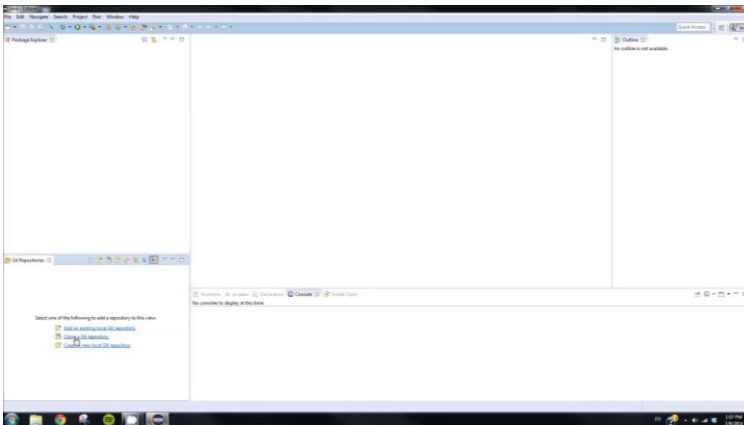
7. Hit “Next” twice, then accept the license agreement and click “Finish”. Gradle should then download and install.



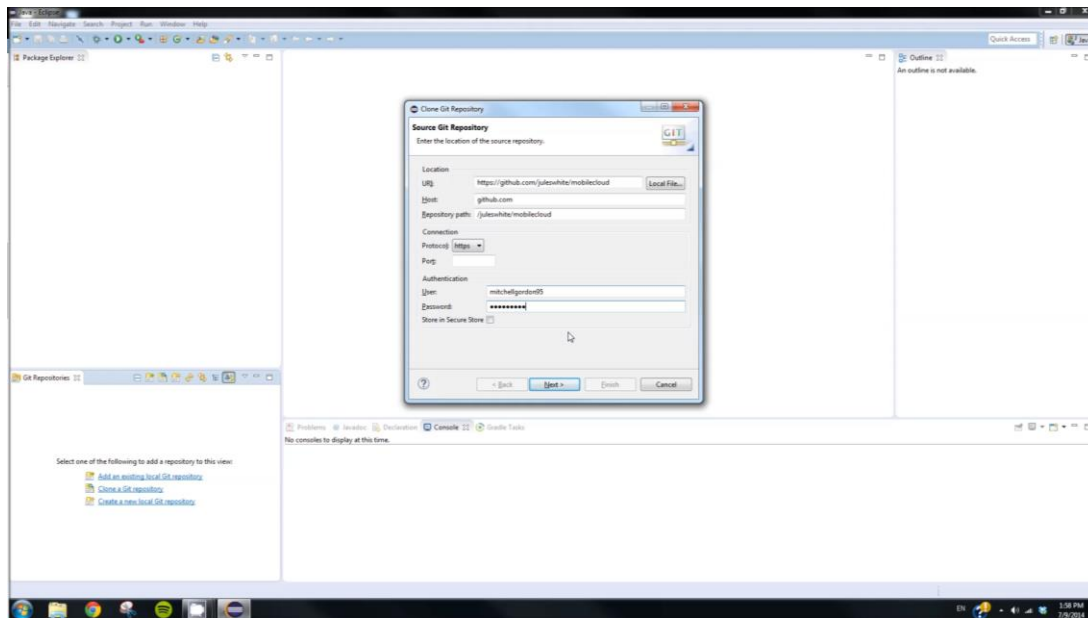




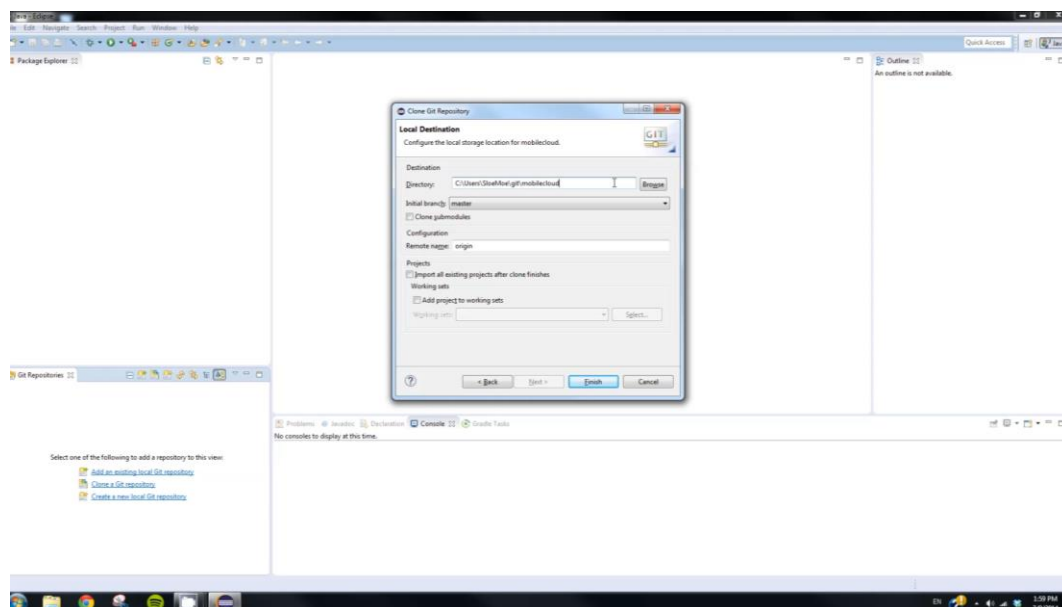
2. In the “Git Repositories” view, click “Clone a Git Repository”. You can also click on the button that has a yellow cylinder with a blue arrow on it to accomplish the same task.



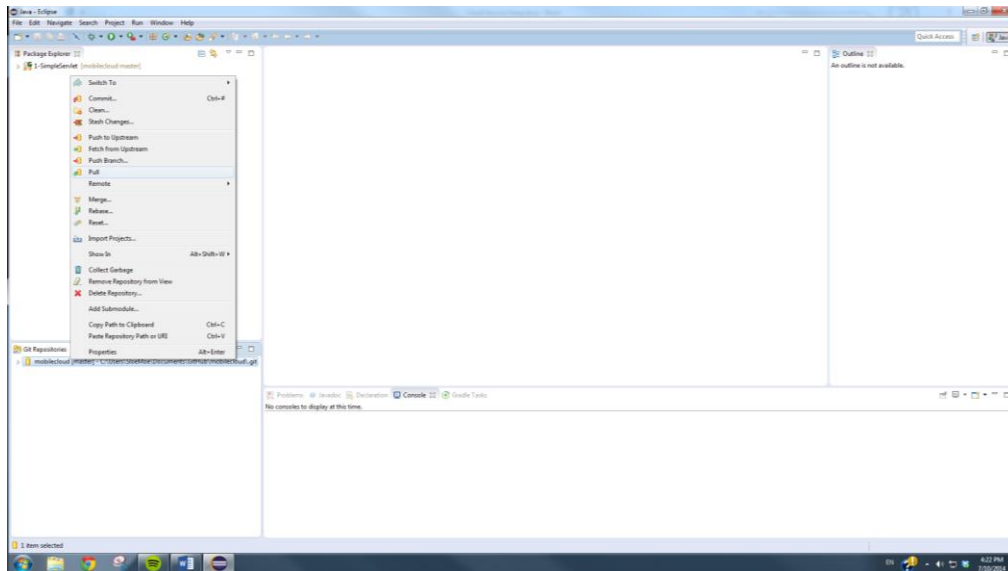
3. Enter the location of the Github repository. At the time this document was created, that repository is located at:  
<https://github.com/juleswhite/mobilecloud-14>
4. Enter your Github username and password. If you do not have a Github account, you may need to make one at <https://github.com>



5. Select a branch to clone.
6. Select a directory to clone the repository to. Normally the default location is fine. \*You will need this path in order to import the projects later\*



7. Click "Finish". This will download the source code into that directory on your hard drive.
8. In the future, we may make changes to the source code or assignments. To have these changes reflected in your local repository, you should regularly do a "git pull". To do this in Eclipse, simply right click on the repository in the "Git Repositories View", and select "Pull".

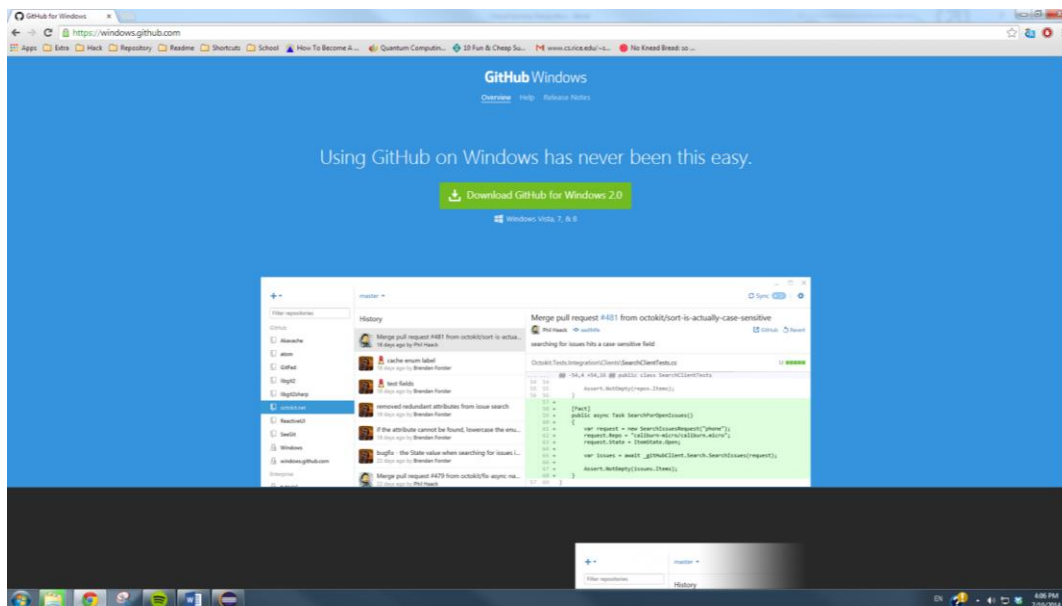


(Using the Github tool)

1. If you do not have git already installed, you will have to install it first. We recommend using the Github version, but there are others on the web. Open a web browser and navigate to:

<http://windows.github.com> or <http://mac.github.com>

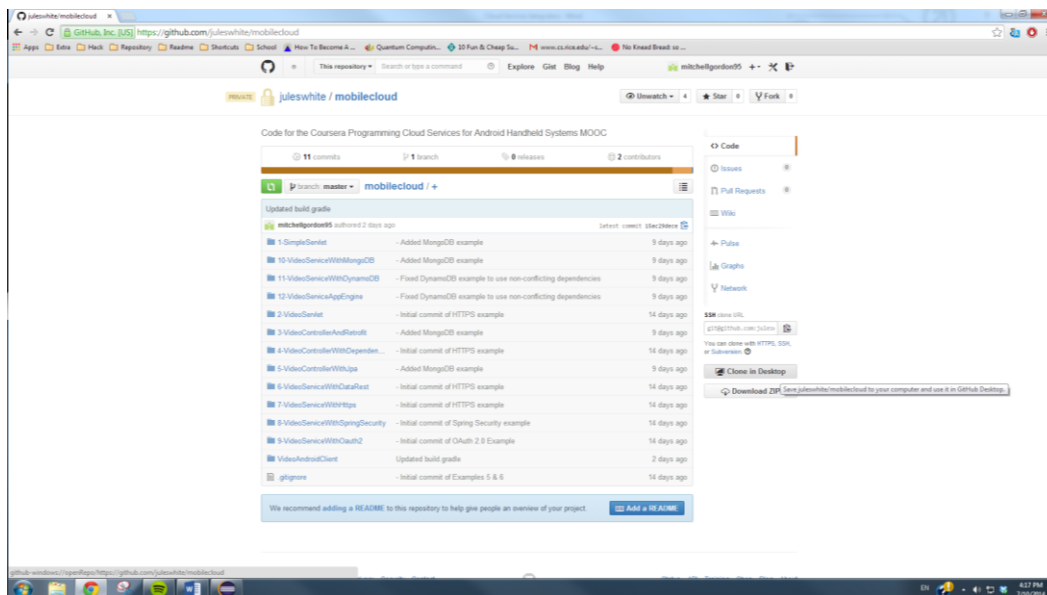
To download the Github tool, simply click “Download GitHub”.



2. Install the application by opening the downloaded file. For Windows, this should be an executable file, which will self-install. For Mac, this should be a zip file, which you will be able to extract and drag to your Applications folder.
3. Now you can clone the repository, either using the GUI or the command line. To use the GUI, open a web browser and navigate to the repository location on Github, which is currently:

<https://github.com/juleswhite/mobilecloud-14>

In the right hand column, there should be a button that says “Clone in Desktop”. That will automatically clone the repository into your local Github folder.



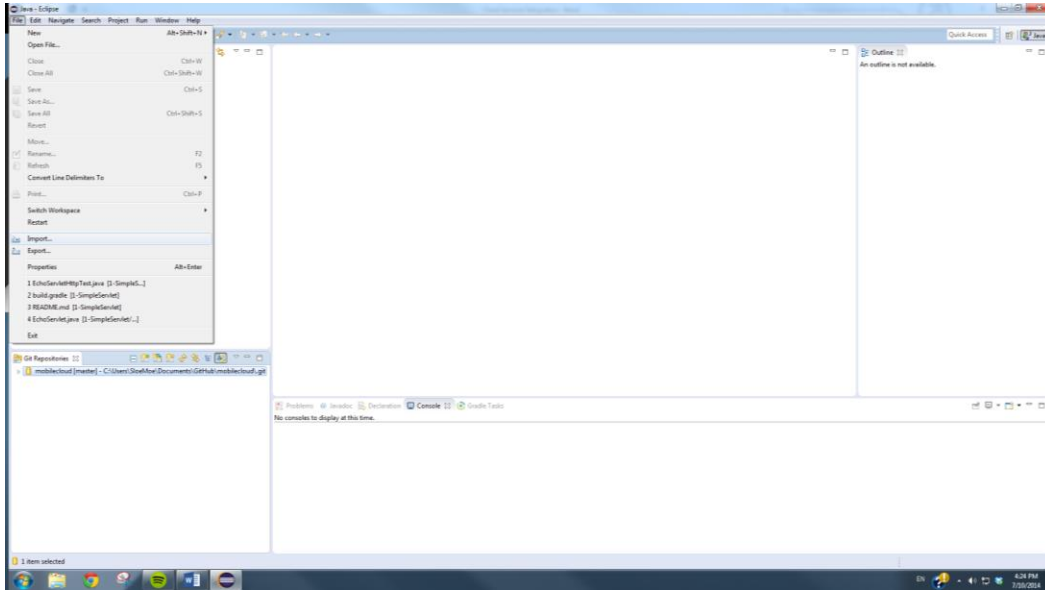
To use the command line, simply open a command line tool (cmd.exe for Windows, Terminal for Mac), and navigate to the folder where you want the repository to live. Then, type:

```
git clone https://github.com/juleswhite/mobilecloud
```

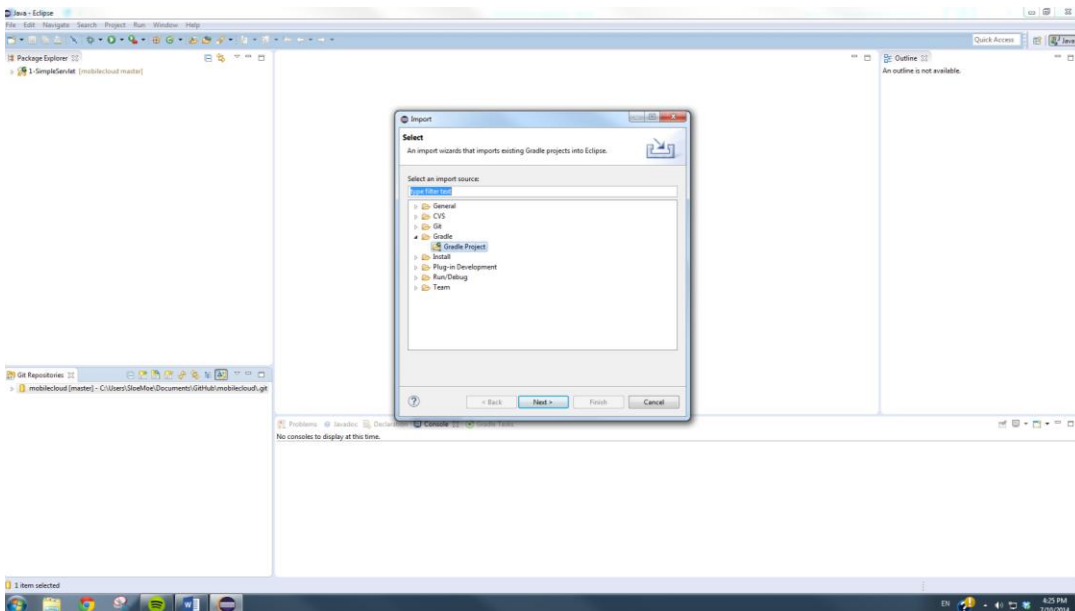
And hit enter. This will create a folder called “mobilecloud” and copy the repository into that folder.

## Importing the Gradle Projects from Eclipse

1. Open Eclipse. Select “File”->”Import”.

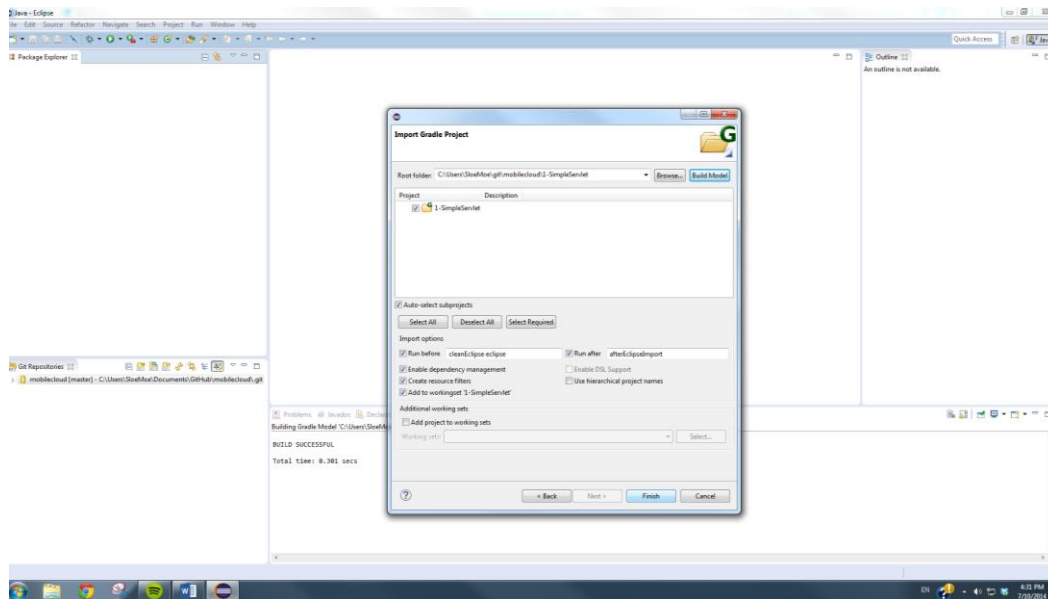


2. Select “Gradle Project”, under “Gradle”.



3. Locate the project. This is one of the folders inside the cloned Github repository that you downloaded in the above steps. The project folder should contain a file called “build.gradle”. However this file may or may not show up in Eclipse’s filesystem browser.

- Click “Build Model”, then check the box next to the project that should appear. Click “Finish” and you should be done.

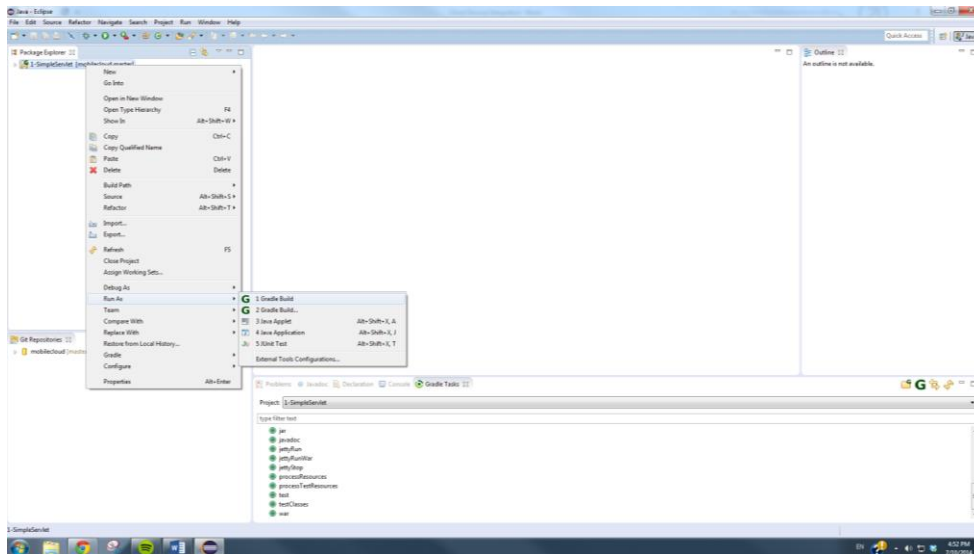


## Running Gradle Projects from Eclipse

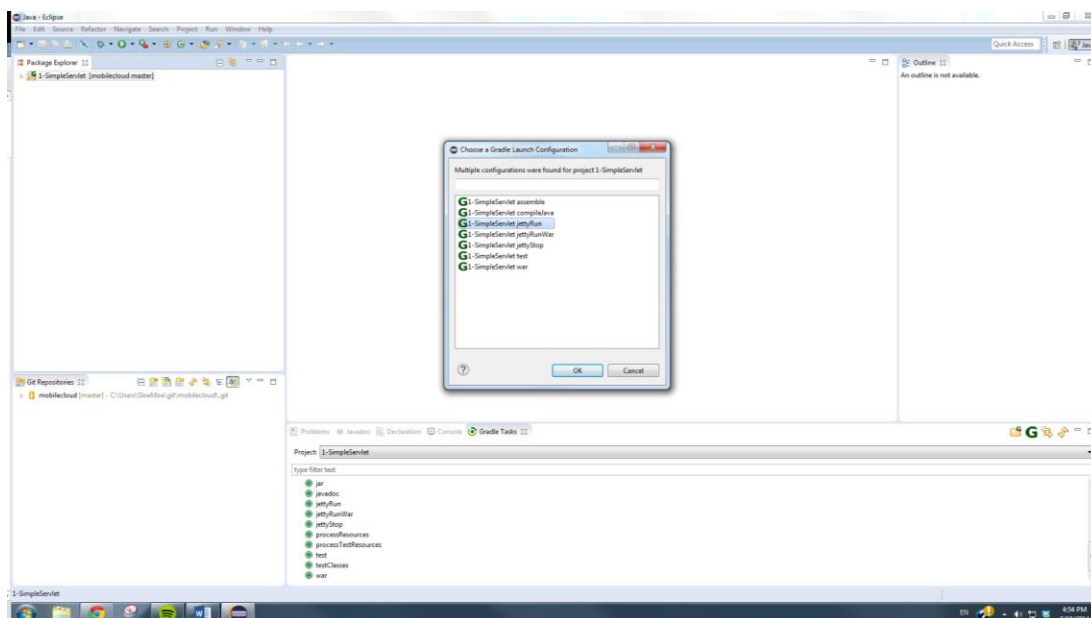
There are three methods of doing this. All methods involve executing Gradle tasks, which can build and run the project. The first two methods can be done from Eclipse, and are generally more convenient than the third method, which is using the Gradle command line tool.

(Method 1: Eclipse Run as...)

1. From Eclipse, right click on the project in the Package Manager and do “Run as...” -> “Gradle Build...”

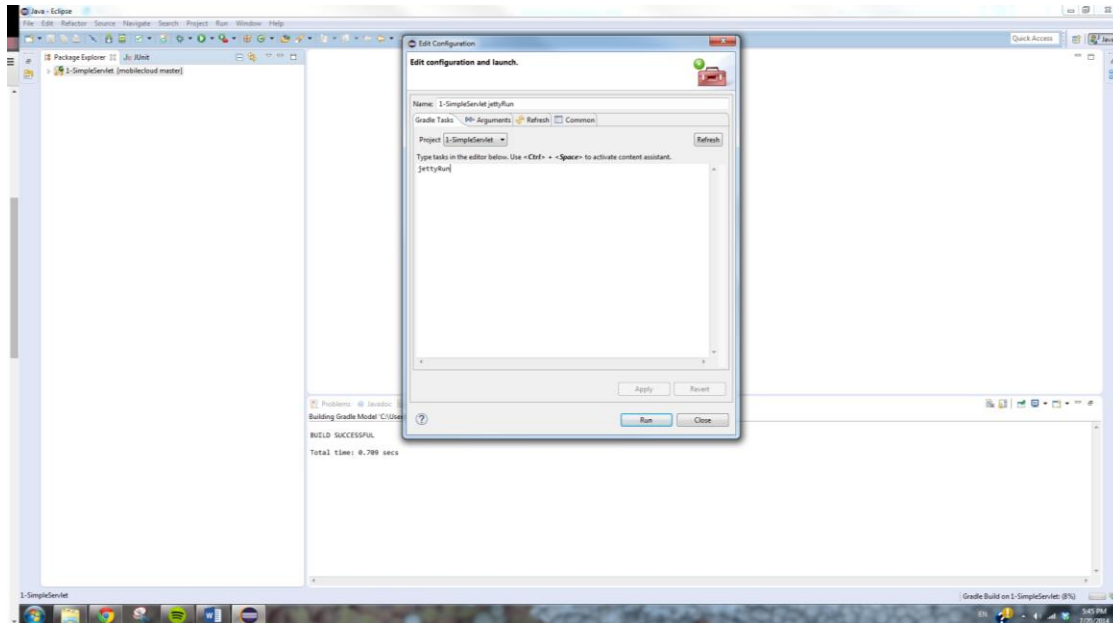


2. Select the “jettyRun” option. This will compile the project and run it as a servlet within the Eclipse environment.



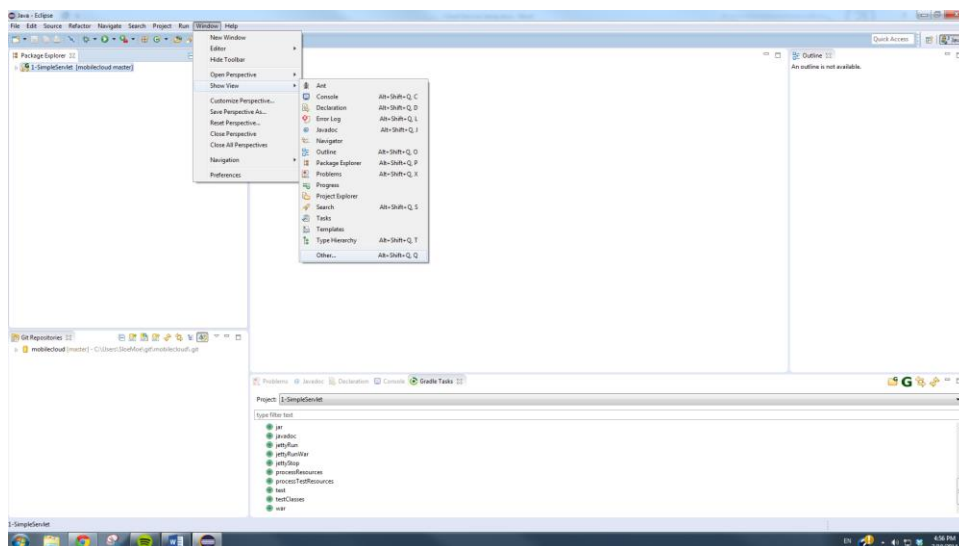


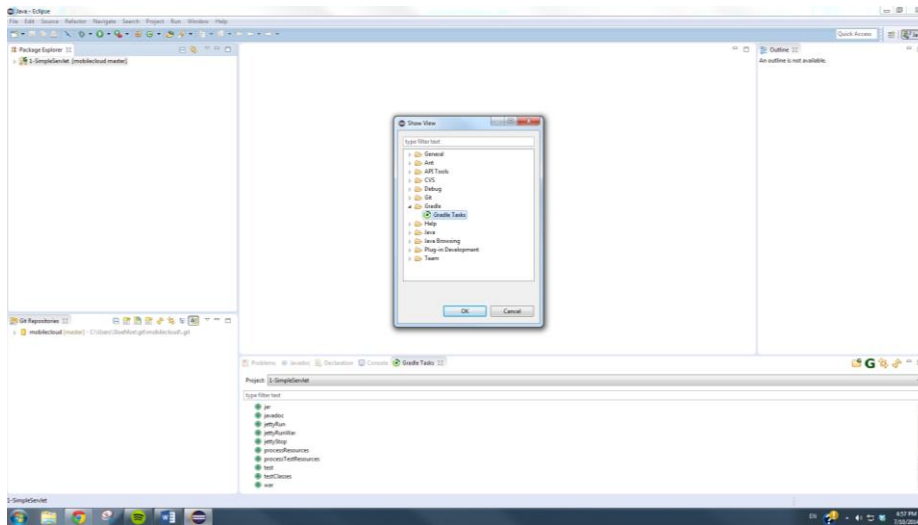
Update: The newest version of the Gradle plugin (as of 7/19/14) will show a more customizable dialogue. This box will ask you to “Type tasks in the editor below. Use <Ctrl> Space to activate content assistant.” In that box, simply type “jettyRun” to run the jettyRun task. If you want to view the other tasks available, use the key combination for the content assistant (<Ctrl> Space)



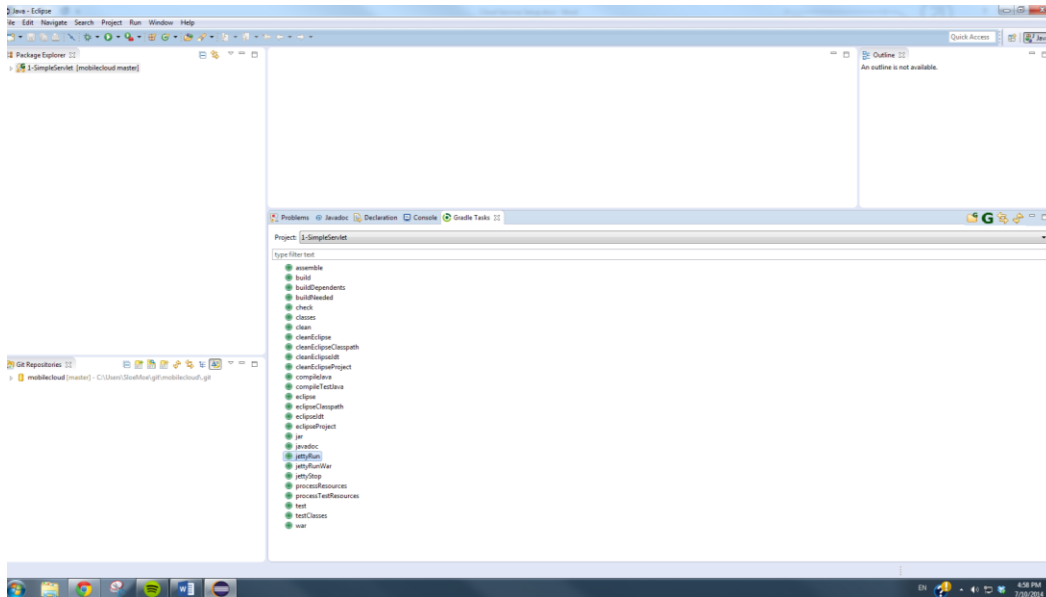
(Method 2: Eclipse Gradle Tasks view)

1. Open the “Gradle Tasks” view. To do this, do Window -> Show View -> Other. The “Gradle Tasks” view should be under the Gradle folder.





2. In the “Gradle Tasks” view, select the “jettyRun” option by double-clicking on it.



This should build and run your servlet.

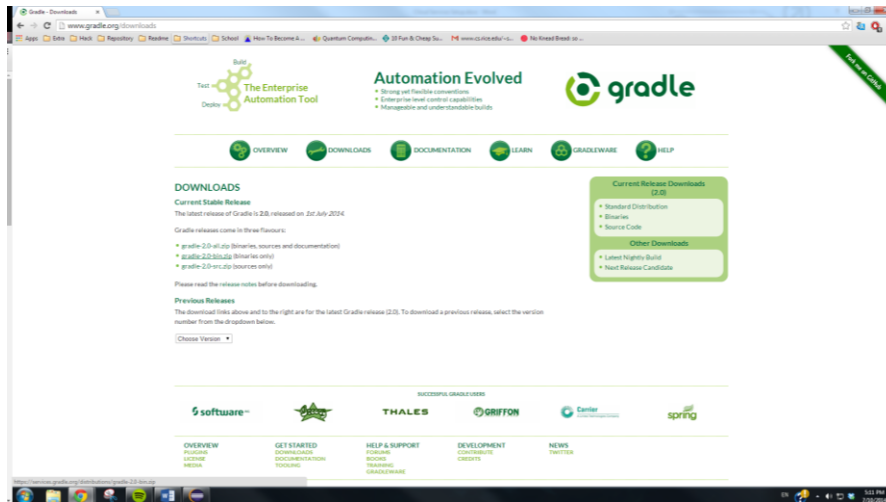
Update: The newest version of the Gradle plug-in (as of 7/19/14) requires you to click the “Refresh Tasks” button, which is right near the big green “G”. This should build the Gradle model and allow you to select a task to run.

(Method 3: Gradle Command Line Tool)

1. Download the Gradle tool from:

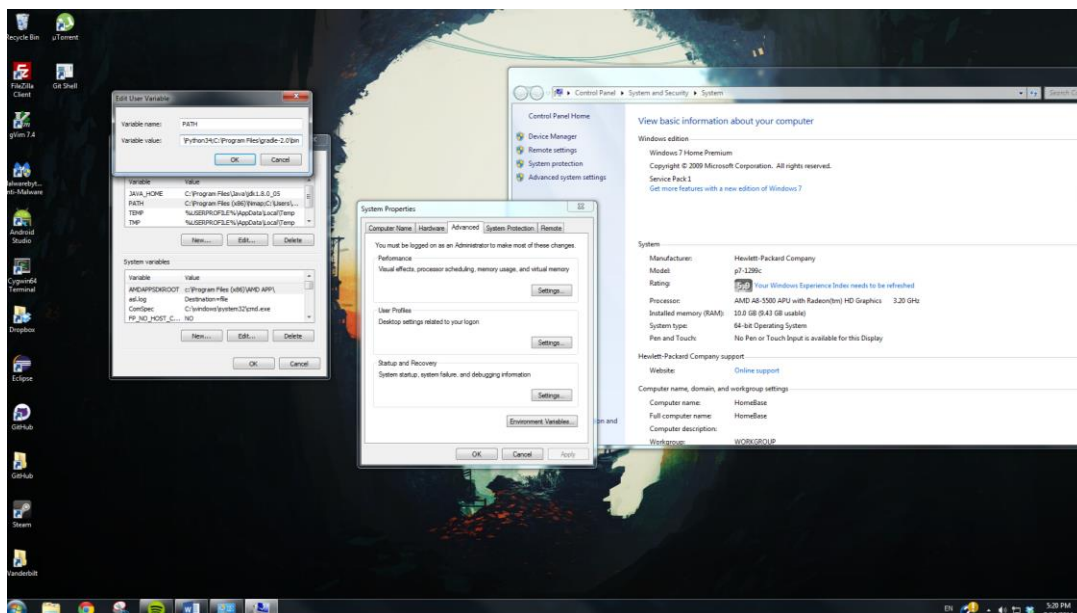
<http://www.gradle.org/downloads>

You want to download the binaries only, so select that link.



2. Unzip the .zip file to somewhere on your hard drive, like “C:\Program Files\gradle”, or “/Users/xxx/Installations/gradle”.
3. Add the “bin” folder from your gradle installation to your system PATH variable. This will tell your OS where to find the gradle executable when you type “gradle” into the command line.

(Windows) Start Menu-> Computer -> Right Click -> Properties. In the left column, click “Advanced System Settings”. Find the “Environment Variables” button and click it. Select the “PATH” variable and click “Edit”. At the end of the value, append a semi-colon, and then the location of your gradle binary. If gradle is in “C:\Program Files\gradle”, then you would append “;C:\Program Files\gradle\bin”. Click “Ok” to exit out of all the dialogs.



(Mac) Open your .profile file. You should be able to do this by opening a terminal and typing:

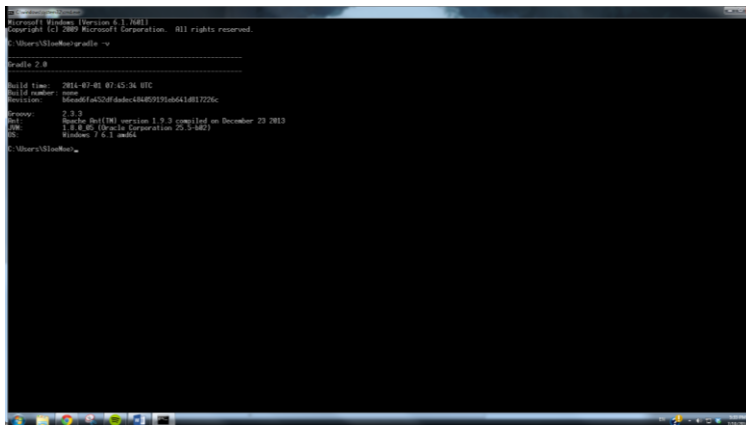
```
open -a TextEdit ~/.profile
```

Copy and paste the following into the file. Replace “/Users/xxx/Installations/gradle” with whatever folder you extracted gradle into:

```
GRADLE_HOME=/Users/xxx/Installations/gradle;  
export GRADLE_HOME  
export PATH=$PATH:$GRADLE_HOME/bin
```

Close the Terminal and reopen it.

4. Test that Gradle is working. Open a command line or terminal and type: “gradle -v” and hit enter. That should display the version of Gradle that we’re working with.



5. Navigate to the folder where your project is located. This folder should contain a “build.gradle” file, which contains all the build information.
6. Type “gradle jettyRun” and hit enter. This tells gradle that there is a build.gradle file in this directory, and that you want to execute the jettyRun task that is defined in that file, along with all the configuration options there. This should build and run your servlet. Note that if you close your terminal, that should stop Jetty.

```
C:\Windows\system32\cmd.exe - gradle jettyRun
07/09/2014 01:59 PM <DIR> 9-VideoService\Hith0auth2
07/09/2014 01:59 PM <DIR> Video\AndroidClient
1 File(s) 91 bytes
16 Dir(s) 257,002,643,456 bytes free

C:\Users\SloeMoe\git\mobilecloud>cd 1-
The system cannot find the path specified.

C:\Users\SloeMoe\git\mobilecloud>cd 1-SimpleServlet

C:\Users\SloeMoe\git\mobilecloud\1-SimpleServlet>dir
Volume in drive C is OS
Volume Serial Number is 56AC-A08F

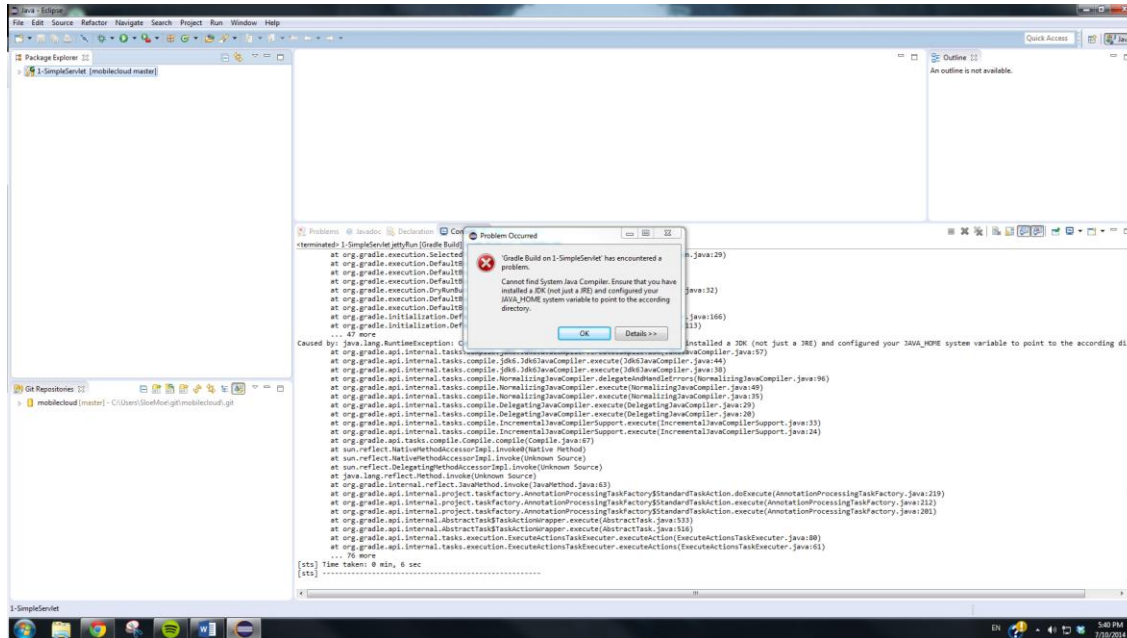
Directory of C:\Users\SloeMoe\git\mobilecloud\1-SimpleServlet

07/10/2014 04:31 PM <DIR> .
07/10/2014 04:31 PM <DIR> .\
07/10/2014 04:31 PM <DIR> 419 .\classpath
07/09/2014 02:01 PM <DIR> .gradle
07/10/2014 04:31 PM <DIR> 459 .project
07/10/2014 04:31 PM <DIR> .settings
07/10/2014 04:31 PM <DIR> bin
07/09/2014 02:01 PM <DIR> build
07/09/2014 01:59 PM <DIR> 321 build.gradle
07/09/2014 01:59 PM <DIR> 688 README.md
07/09/2014 01:59 PM <DIR> src
4 File(s) 1,887 bytes
7 Dir(s) 257,002,643,456 bytes free

C:\Users\SloeMoe\git\mobilecloud\1-SimpleServlet>gradle jettyRun
:compileJava
:processResources UP-TO-DATE
:classes
:jettyRun
```

## (Optional) Installing JDK to fix Gradle

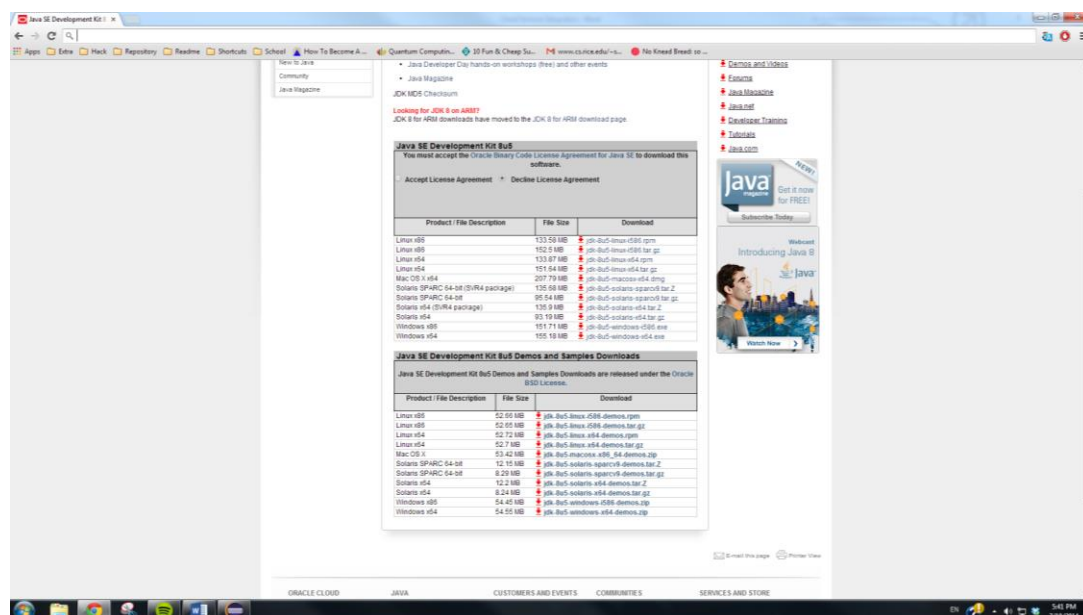
Sometimes Gradle may complain about not being able to find the JDK, like this:



To fix this, we have to install the JDK and then point the Gradle Plug-in to it.

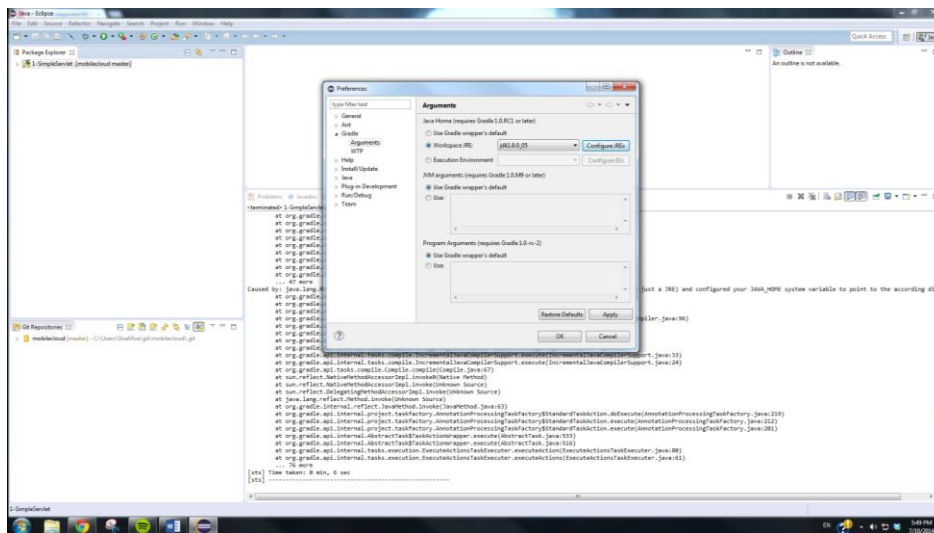
1. Open a web browser and download the JDK from:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

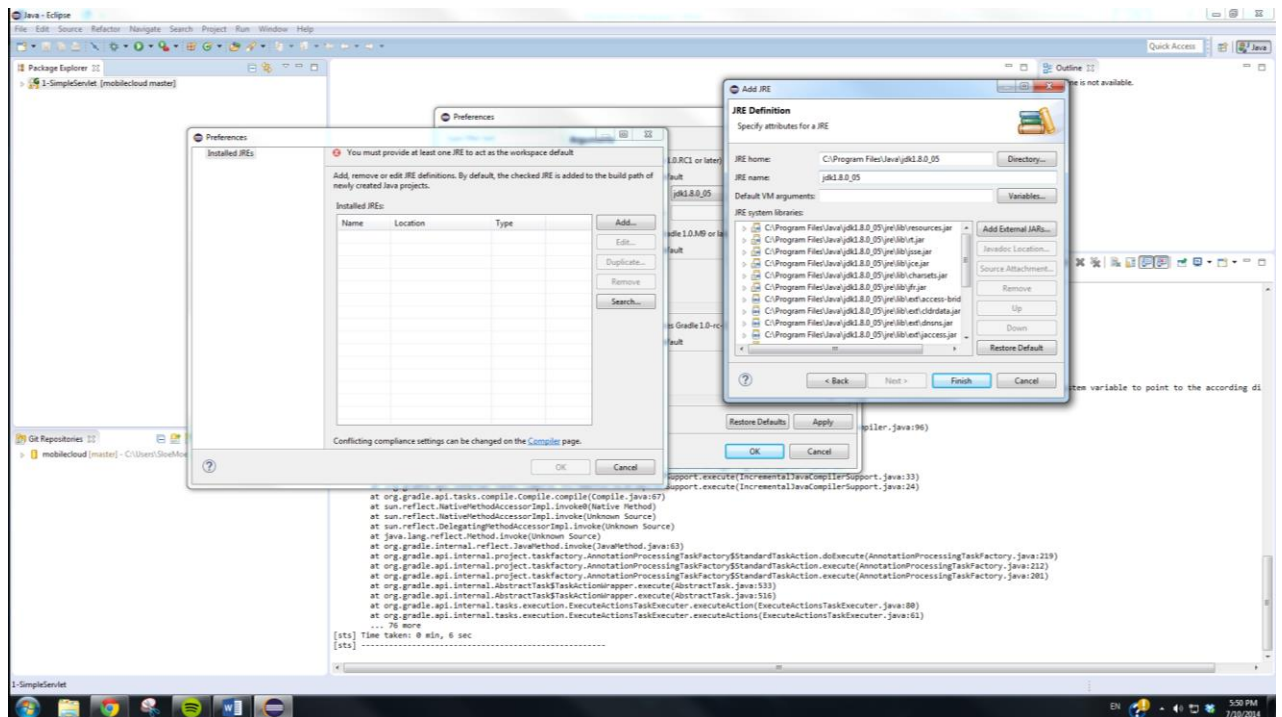


Select the appropriate file for your system.

2. Once you have downloaded the file, open it. It should be an executable that installs itself. Simply follow the prompts from the installation wizard.
3. In Eclipse, go to Window->Preferences->Gradle->Arguments.
4. At the top, select the radio button next to “Workspace JRE”. From the drop-down menu, select the JDK you just installed.



5. If the JDK does not show up, you must add the JDK to your workspace. Click the “Configure JRE’s” button, and then the “Add” button. Select “Standard VM”, and hit “Next”. You will be able to navigate to the JDK you just downloaded and installed. Then hit “Finish” to add the JDK to your workspace. Now go back to step 4.



This should fix any JDK related problems with Gradle.