Jamie Sinn

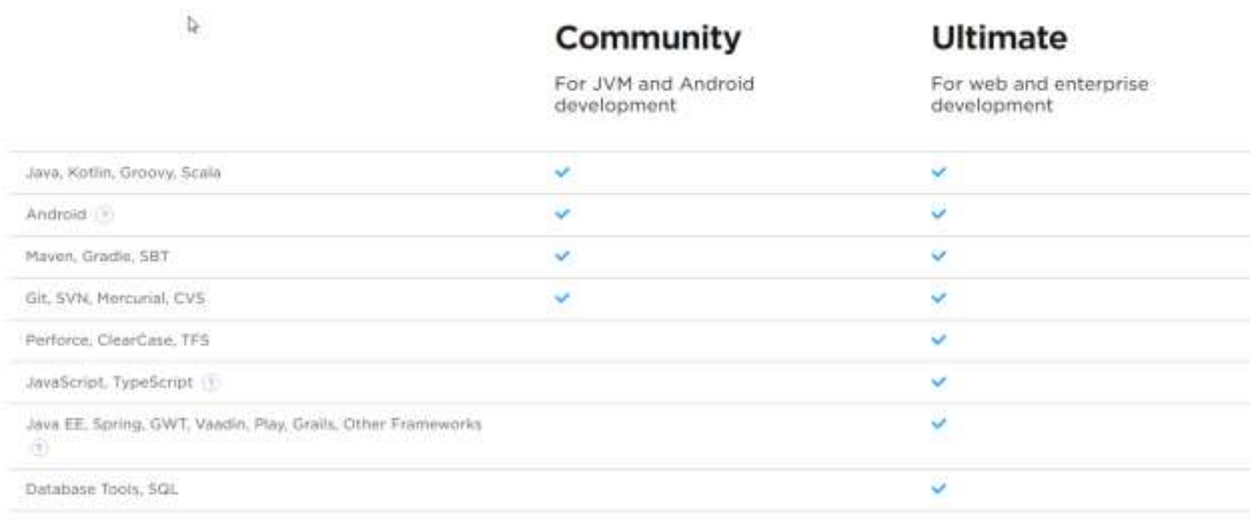
IntelliJ IDEA 2016.1 Getting Started Guide for FIRST Robotics Competition

1 Pre-requisites

- GitHub account.
- Knowledge of your computer and how to use it.
- Administrator Account on your computer.
- Access to the Internet

2 Installing IntelliJ IDEA

IntelliJ IDEA by JetBrains is the leading Java IDE for professional use. It is the most user friendly and ergonomic IDE that new and veteran developers can use alike. It has built in tools for Git, any JVM language, as well as enterprise frameworks and web development. For FIRST Robotics use, you can choose to register a student account with JetBrains and get a free license for the Ultimate Version. For FRC purposes, the Community Edition is just fine.



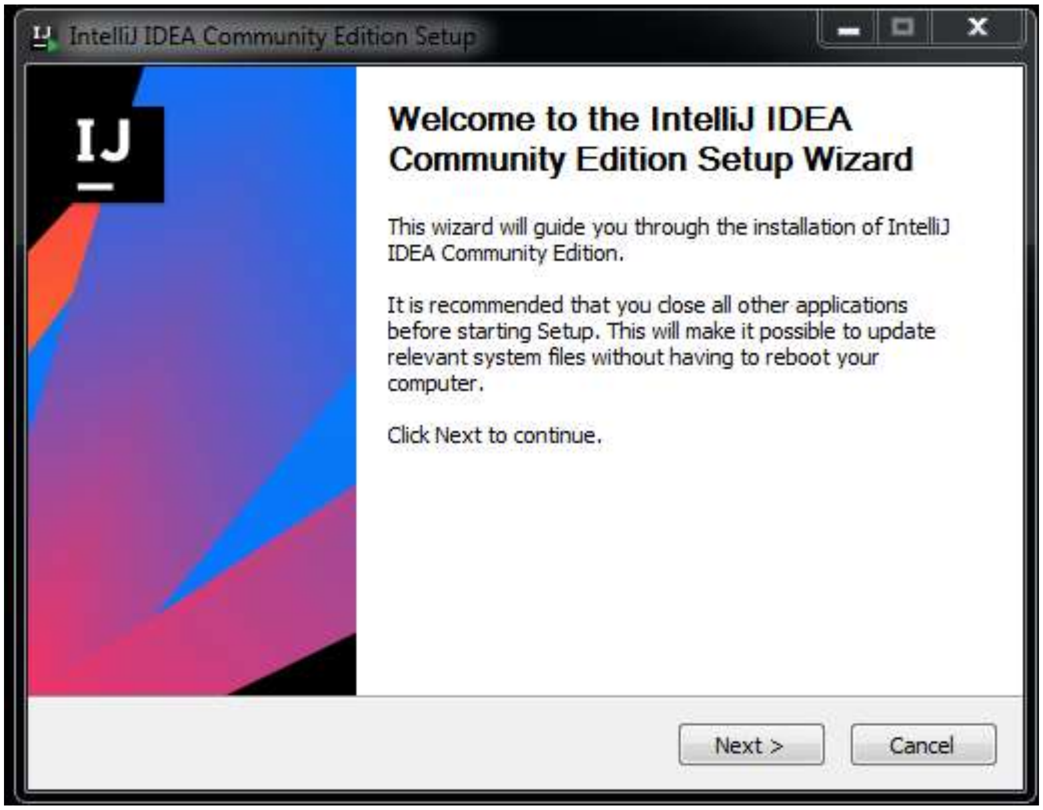
	Community For JVM and Android development	Ultimate For web and enterprise development
Java, Kotlin, Groovy, Scala	✓	✓
Android	✓	✓
Maven, Gradle, SBT	✓	✓
Git, SVN, Mercurial, CVS	✓	✓
Perforce, ClearCase, TFS		✓
JavaScript, TypeScript		✓
Java EE, Spring, GWT, Vaadin, Play, Grails, Other Frameworks		✓
Database Tools, SQL		✓

Figure 1 Overview of version comparisons

To download the Community edition, visit this link and download the appropriate version for your operating system. For the sake of simplicity, I'll assume you're using Windows.

https://www.jetbrains.com/idea/download/ - Download Link. Prone to changes.

Once you have accepted the UAC prompt (if UAC is enabled.), you will be prompted to this window:



IntelliJ IDEA Community Edition Setup

Welcome to the IntelliJ IDEA Community Edition Setup Wizard

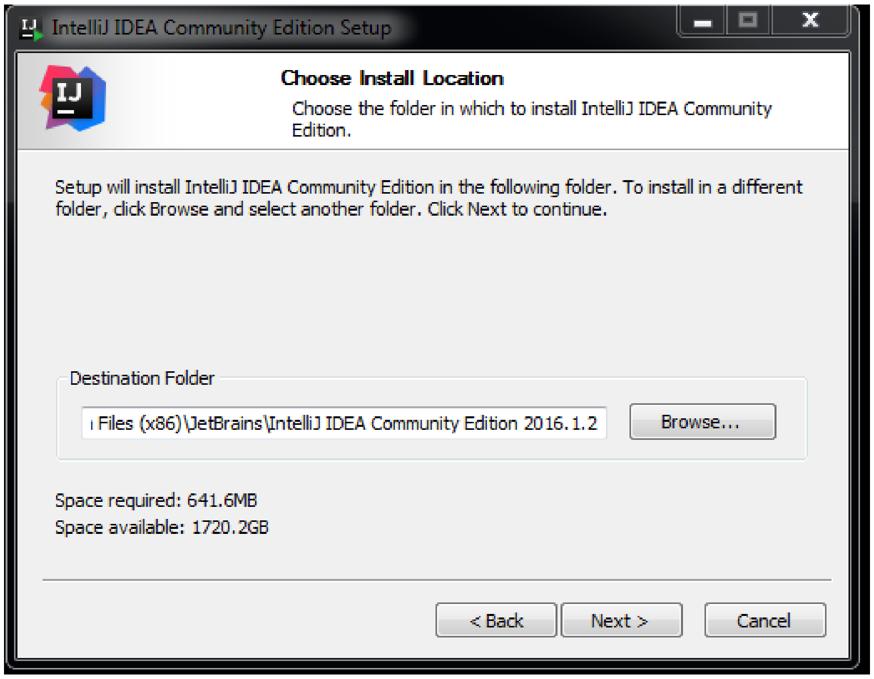
This wizard will guide you through the installation of IntelliJ IDEA Community Edition.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Figure 2 Beginning of install

Once you have clicked next, you will be prompted for the install location. This is perfectly fine to be left as default.



IntelliJ IDEA Community Edition Setup

Choose Install Location

Choose the folder in which to install IntelliJ IDEA Community Edition.

Setup will install IntelliJ IDEA Community Edition in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

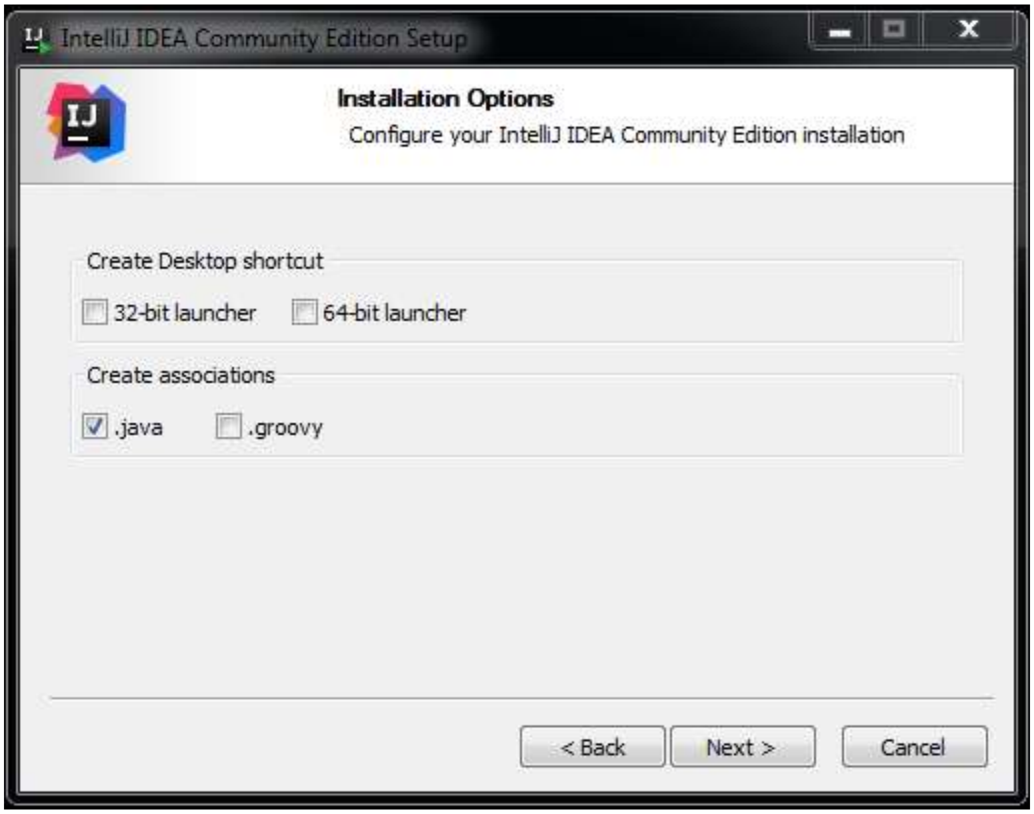
Destination Folder

ı Files (x86)\JetBrains\IntelliJ IDEA Community Edition 2016.1.2

Space required: 641.6MB

Space available: 1720.2GB

On the next screen, you will be prompted to create associations to file extensions and create a desktop shortcut. Select .java and whatever desktop link you wish


IntelliJ IDEA Community Edition Setup

Installation Options

Configure your IntelliJ IDEA Community Edition installation

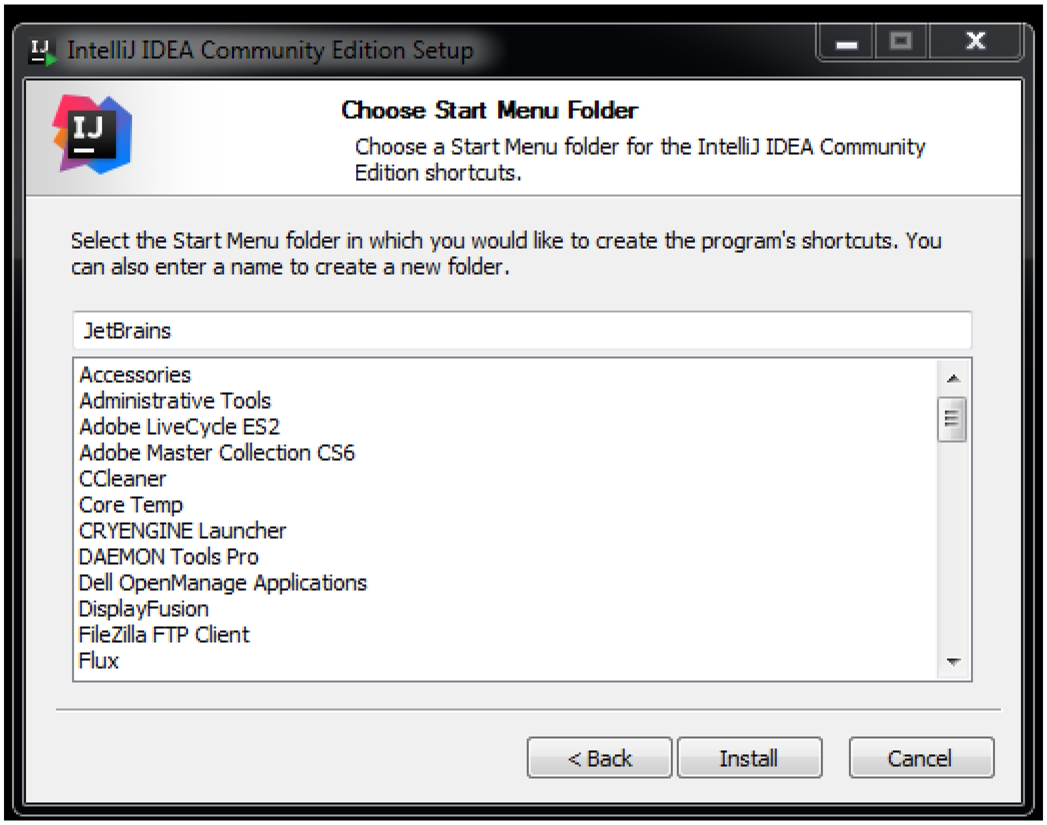
Create Desktop shortcut

- [ ] 32-bit launcher
- [ ] 64-bit launcher

Create associations

- [x] .java
- [ ] .groovy

Once you click next, you will be prompted to click the final Install button. Click this, and you are done the install! Once it is installed, you'll be prompted for settings and personalization, this is not covered in this guide as it is very personalized.



Choose Start Menu Folder

Choose a Start Menu folder for the IntelliJ IDEA Community Edition shortcuts.

Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

JetBrains

Accessories
Administrative Tools
Adobe LiveCycle ES2
Adobe Master Collection CS6
CCleaner
Core Temp
CRYENGINE Launcher
DAEMON Tools Pro
Dell OpenManage Applications
DisplayFusion
FileZilla FTP Client
Flux

3 GitHub Client

IntelliJ comes with a basic set of Git tools, but it does not come with enough to operate perfectly with GitHub. Now, this guide does assume that you will be using Windows again, due to simplicity. OSX is a very similar install process. And if you're using Linux, I assume you know how to use your package manager to install Git. You'll be doing everything command line.

3.1 GitHub client

https://desktop.github.com/

GitHub's desktop client is very good, and is very nice to work with for new users. Click the download button, and follow the install guide for your given OS. The best guide is available on GitHub itself.

https://help.github.com/desktop/guides/getting-started/installing-github-desktop/

4 Java Development Kit

IntelliJ again comes pre-packaged with the JDK, but a very basic version of it, and it's in reality mostly the user end of it, the JRE.

Download and install the JDK for your operating system here:
http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
Remember to use the latest release for your operating system.

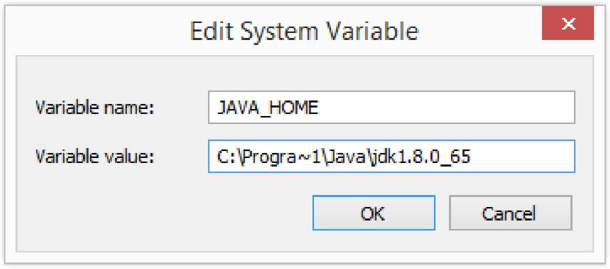
Once you have installed it, you will need to set an Environment Variable called JAVA_HOME. This tells all the Java applications where to find the JDK install directory. For the sake of brevity, please follow the guide here. (Thanks to Atlassian for this Guide: https://confluence.atlassian.com/doc/setting-the-java_home-variable-in-windows-8895.html)

4.1 Set the JAVA_HOME Variable

To set the JAVA_HOME variable:

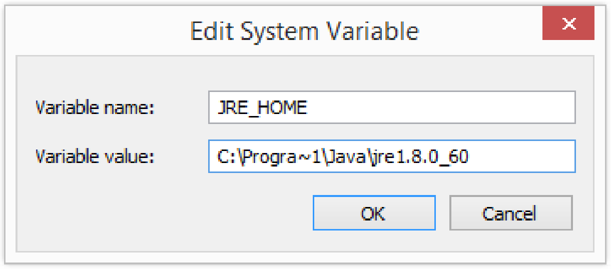
Find out where Java is installed. If you didn't change the path during installation, it will be something like this:
`C:\Program Files\Java\jdk1.8.0_65`

1. In Windows 7 right click **My Computer** and select **Properties** > **Advanced**.
   In Windows 8 go to **Control Panel** > **System** > **Advanced System Settings**.
2. Click the **Environment Variables** button.
3. Under **System Variables**, click **New.**
4. In the **Variable Name** field, enter:
5. `JAVA_HOME` if you installed the JDK (Java Development Kit)
   or
6. `JRE_HOME` if you installed the JRE (Java Runtime Environment)
7. In the **Variable Value** field, enter your JDK or JRE installation path.
8. If the path contains spaces, use the shortened path name, for example `C:\Progra~1\Java\jdk1.8.0_65`)


Edit System Variable

Variable name: JAVA_HOME

Variable value: C:\Progra~1\Java\jdk1.8.0_65


Edit System Variable

Variable name: JRE_HOME

Variable value: C:\Progra~1\Java\jre1.8.0_60

Note for Windows users on 64-bit systems

Progra~1 = 'Program Files'
Progra~2 = 'Program Files(x86)'

Click **OK** and **Apply Changes** as prompted.

You'll need to close any re-open any command windows that were open before you made these changes as there's no way to reload environment variables from an active command prompt. If the changes don't take effect even after reopening the command window, restart Windows.

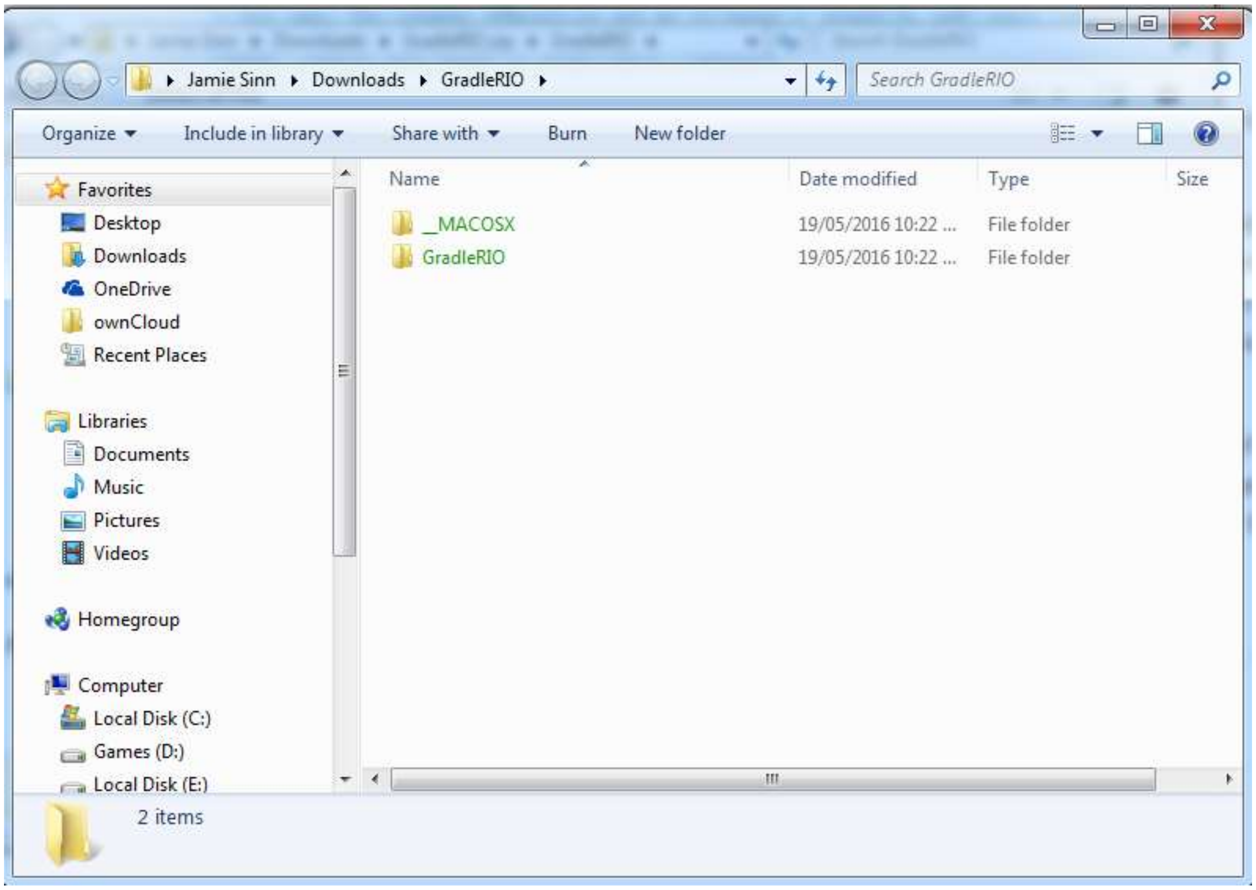
5 GRADLERIO

All the steps prior to this is just to set up Java and IntelliJ for development work. Now, we have to do the hard part and install the build manager/middleware that handles deploying code to the robot, as well as building.

https://github.com/Open-RIO/GradleRIO/releases

GradleRIO is the project, run by team 5333's programmer. It runs off of Gradle, a dependency/build manager for many languages.

Once you have downloaded the latest version of GradleRIO, you must unzip the folder. When done, the folder will look somewhat like this.



Name	Date modified	Type	Size
__MACOSX	19/05/2016 10:22 ...	File folder	
GradleRIO	19/05/2016 10:22 ...	File folder	

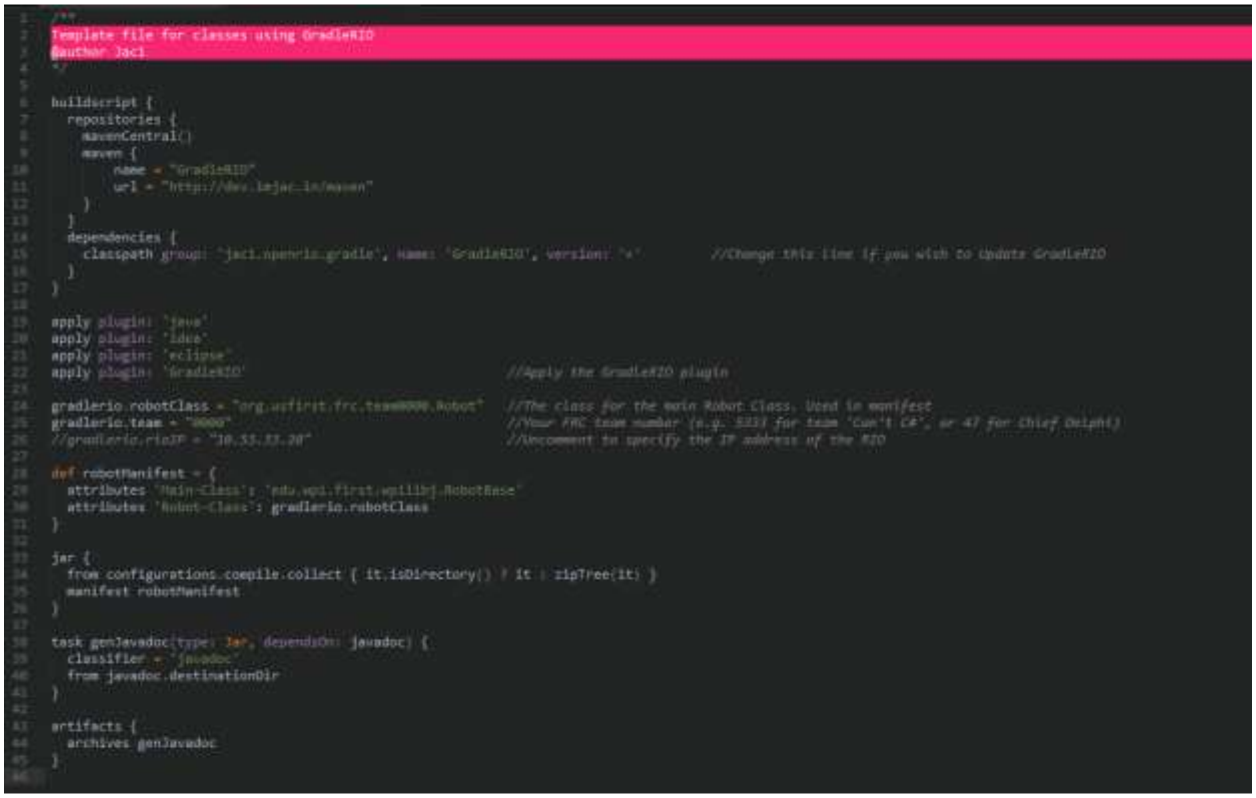
Open a command prompt window, and cd to the GradleRIO directory inside of the root folder. Inside of it, there will be several files. The ones we are looking at and need to edit are:

- build.gradle
- gradlew.bat

Now would be a good time to read of the GradleRIO4Dummies.txt file, which will explain the use and basics of GradleRIO.

Open build.gradle with your favourite text editor (Preferably Notepad++ or Sublime Text).

Inside, you will see the following:

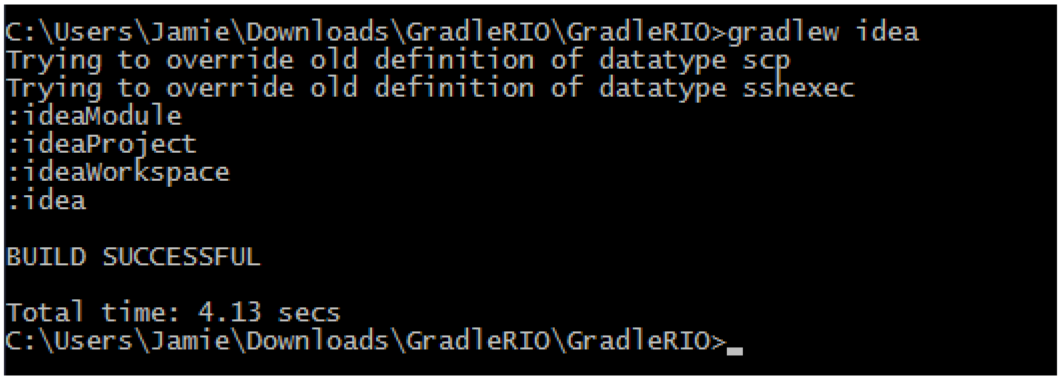


```
/**
Template file for classes using GradleRIO
@author Jaci
*/

buildscript {
  repositories {
    mavenCentral()
    maven {
        name = "GradleRIO"
        url = "http://dev.imjac.in/maven"
    }
  }
  dependencies {
    classpath group: 'jaci.openrio.gradle', name: 'GradleRIO', version: '+'          //Change this line if you wish to Update GradleRIO
  }
}

apply plugin: 'java'
apply plugin: 'idea'
apply plugin: 'eclipse'
apply plugin: 'GradleRIO'                                 //Apply the GradleRIO plugin

gradlerio.robotClass = "org.usfirst.frc.team0000.Robot"   //The class for the main Robot Class. Used in manifest
gradlerio.team = "0000"                                   //Your FRC team number (e.g. 5333 for team 'Can't C#', or 47 for Chief Delphi)
//gradlerio.rioIP = "10.53.33.20"                         //Uncomment to specify the IP address of the RIO

def robotManifest = {
  attributes 'Main-Class': 'edu.wpi.first.wpilibj.RobotBase'
  attributes 'Robot-Class': gradlerio.robotClass
}

jar {
  from configurations.compile.collect { it.isDirectory() ? it : zipTree(it) }
  manifest robotManifest
}

task genJavadoc(type: Jar, dependsOn: javadoc) {
  classifier = 'javadoc'
  from javadoc.destinationDir
}

artifacts {
  archives genJavadoc
}
```

This is the build script that Gradle uses. The key that you need to edit is **gradlerio.team** and **gradlerio.robotClass** – In both of these, you will find the numbers **0000**, replace this with your team number, zero padding if it is under 1000.

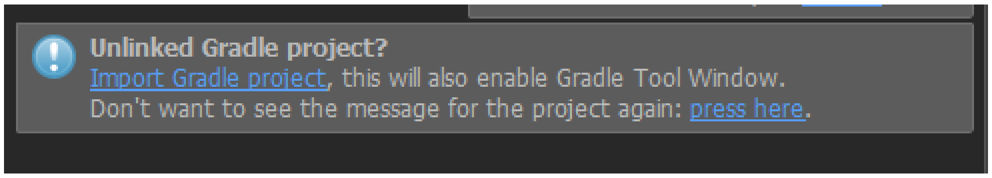
5.1 INTELLIJ SETUP

Once this is complete, return to the command prompt window that you have open, and type **gradlew idea**. This will set up the IntelliJ IDEA. You will see a similar output to below.



```
C:\Users\Jamie\Downloads\GradleRIO\GradleRIO>gradlew idea
Trying to override old definition of datatype scp
Trying to override old definition of datatype sshexec
:ideaModule
:ideaProject
:ideaWorkspace
:idea

BUILD SUCCESSFUL

Total time: 4.13 secs
C:\Users\Jamie\Downloads\GradleRIO\GradleRIO>
```

Once created, open the project in IntelliJ. Once opened, you will see the popup balloon of the following.



Unlinked Gradle project?
Import Gradle project, this will also enable Gradle Tool Window.
Don't want to see the message for the project again: press here.

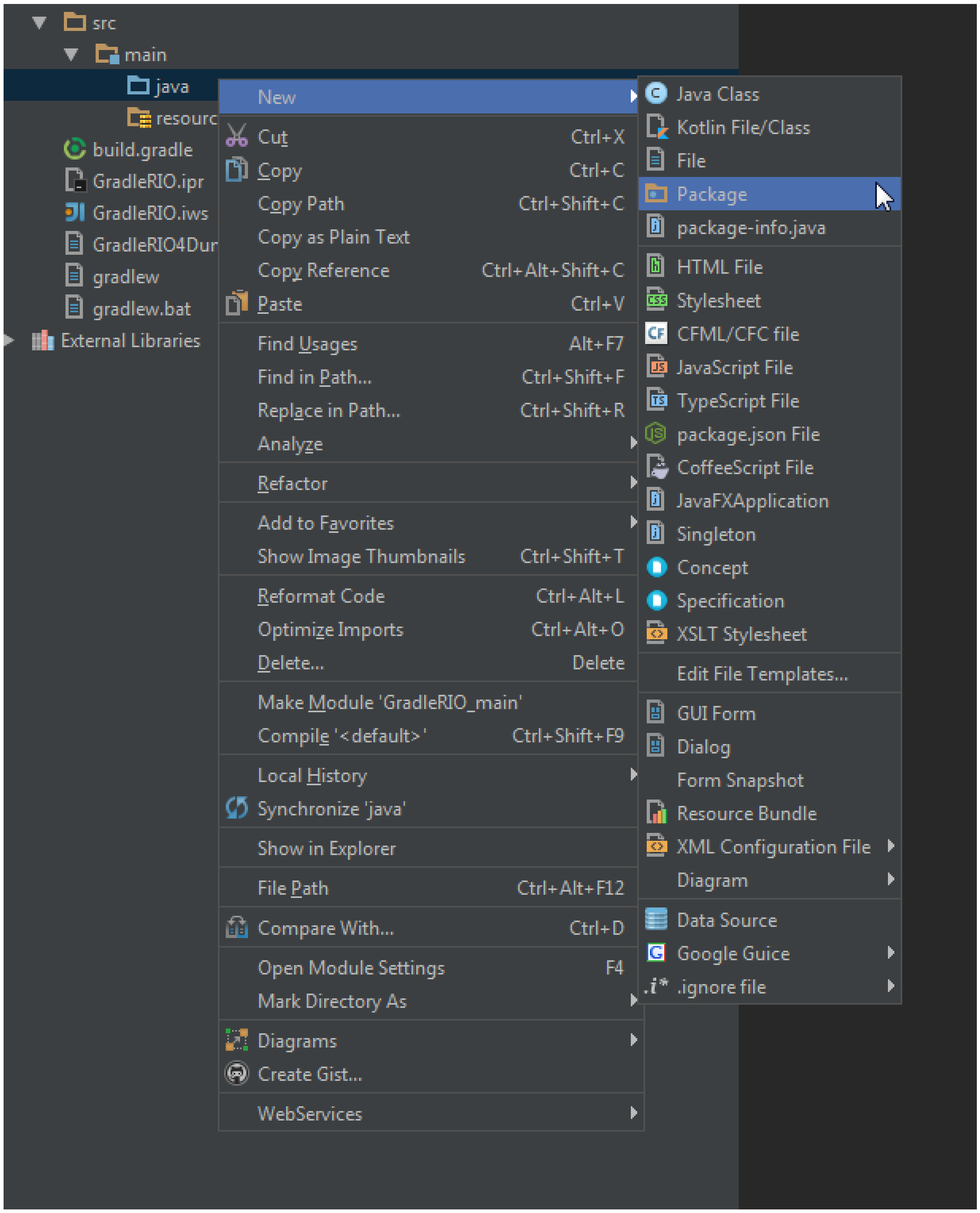
Click Import Gradle Project.

This will prompt you to set the settings for the project. You can simply click OK. It will collect the information it needs and build the project according to IntelliJ's liking.

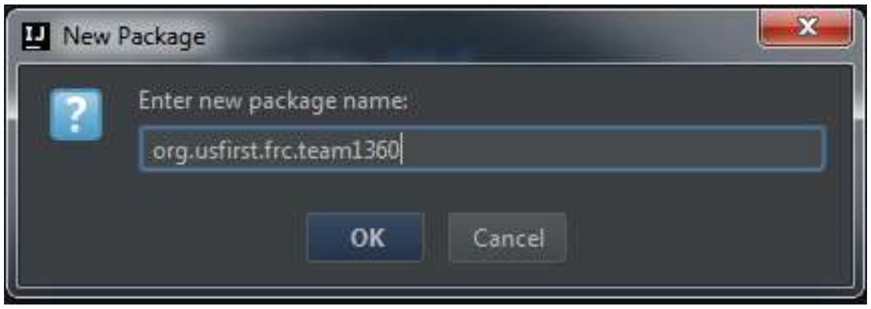
5.2 Project Setup

Once you have opened the project, you will notice that is completely blank! Time to add some classes and packages.

Expand the src folder tree, and right click on the java folder. Follow the below tree to create a new package.



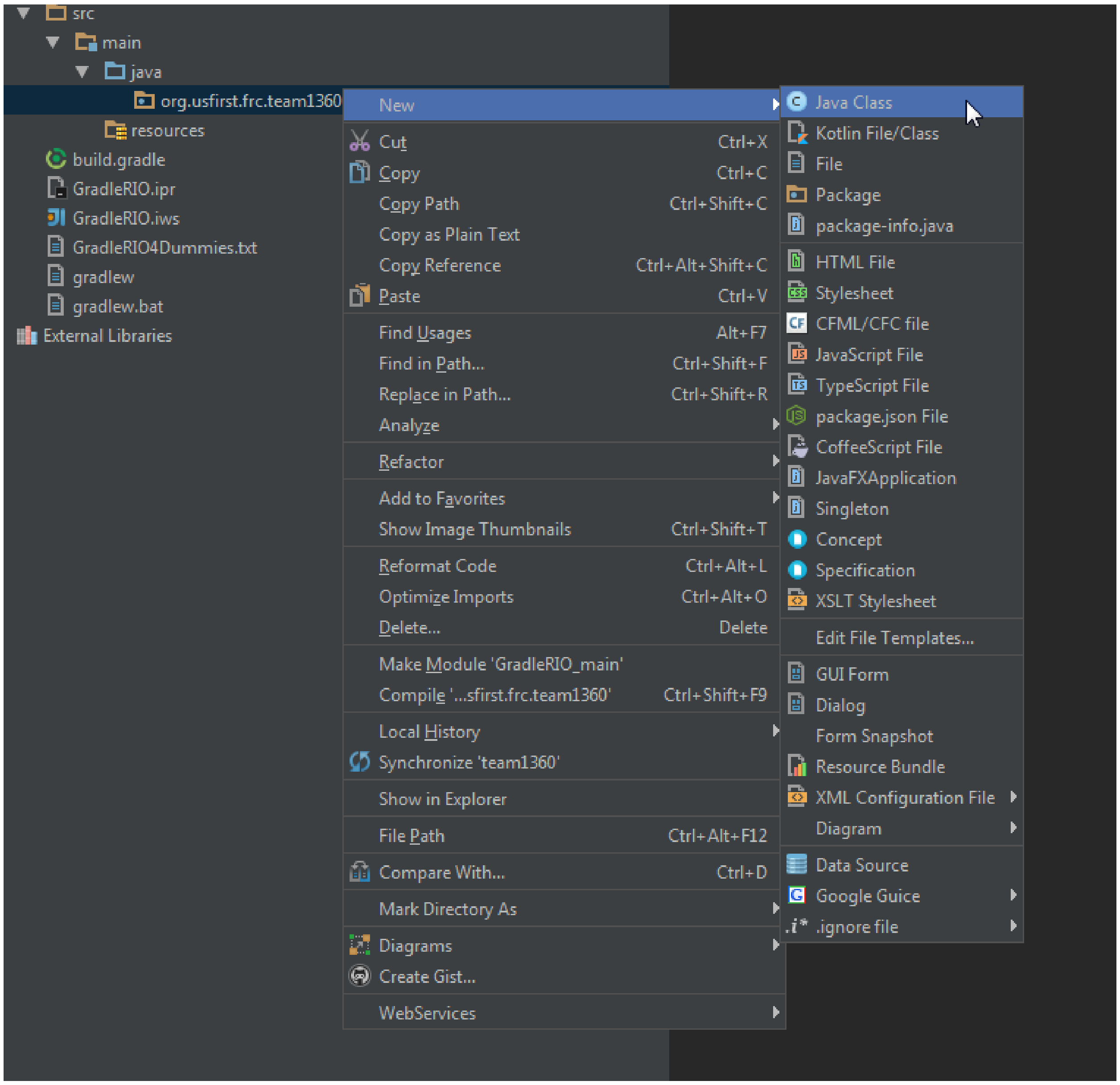
The package name is very important, it must correspond to the one that you have set in the build.gradle file. **org.usfirst.frc.team####** is what was set, use this for the value, replacing #### with your team number.



org.usfirst.frc.team1360

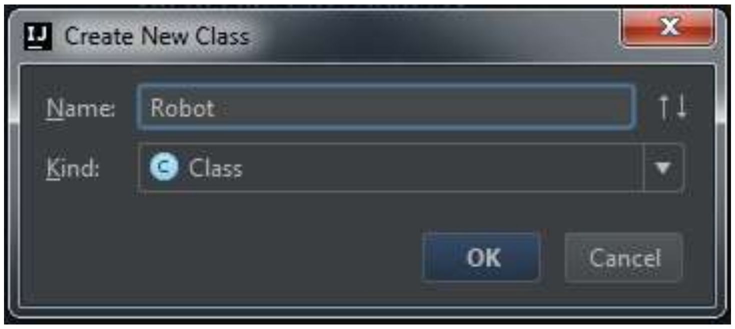
Once you have this set up, you are free to create whatever remaining packages you wish. But first, we must create the main Robot class!

5.3 Main Robot Class



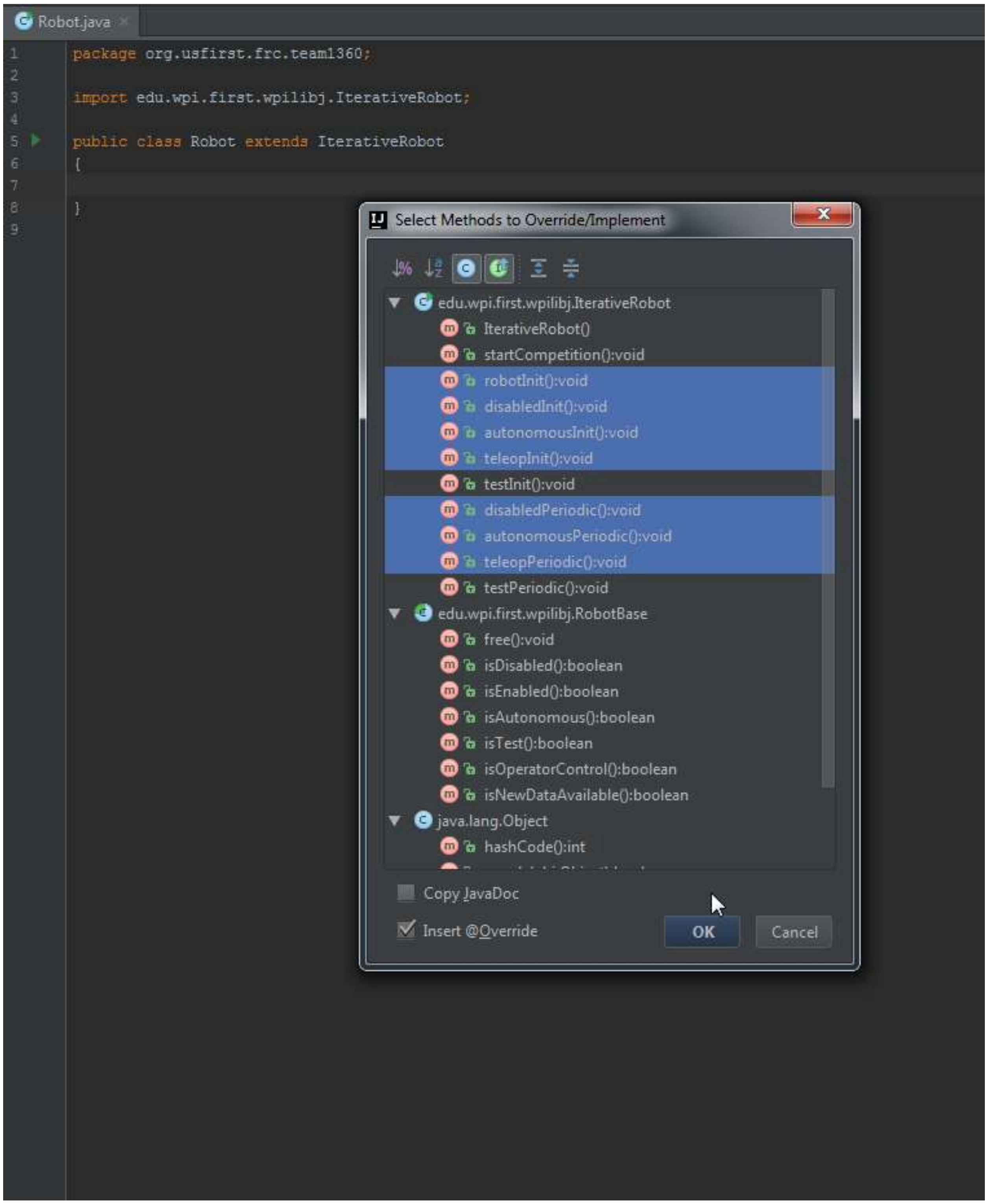
Jamie Sinn

Click Java Class, and name it Robot



Now, we have a bare class. Great. Now what? We have to edit it to become an actual robot class! Make the class extend the **IterativeRobot** class. IntelliJ will automatically recognize this, and add the relevant import. Once you have reached this stage, nothing happens. You still have a blank class. This is where the fun comes in. Press **Ctrl + O**

This brings up the Override/Implement dialog. Select any methods that you need. Generally, robotInit, disabledInit, autonomousInit, teleopInit, and the relevant periodic classes for the competition times are all that’s needed.



```java
package org.usfirst.frc.team1360;

import edu.wpi.first.wpilibj.IterativeRobot;

public class Robot extends IterativeRobot
{

}
```

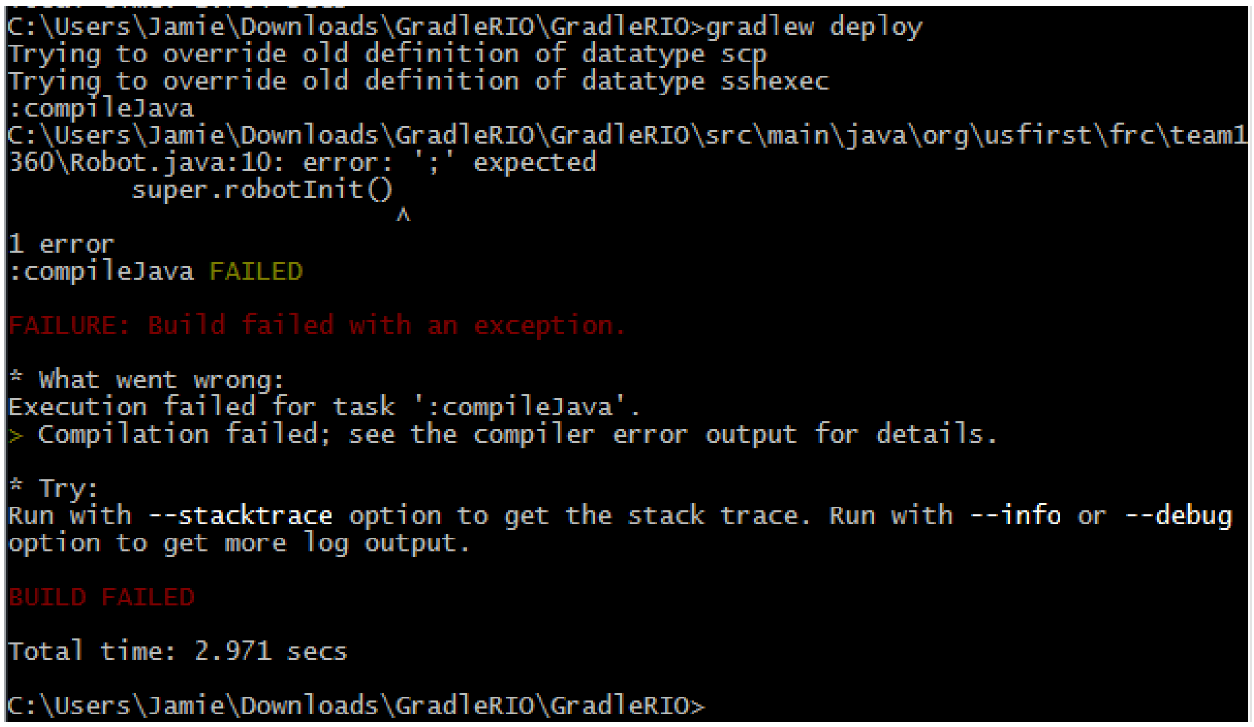
From here, it's all you! You create new packages via the same method as before. Commands simply extend the Command class, Subsystems extend the Subsytem class, and you're off to the competition!

6 Robot Control and Downloading Code

(Note. There **WILL** be errors in the screenshots, I do not have a roboRIO to experiment with and give examples.)

Now, back to that command prompt window. Remember the first argument we gave it was “idea”? Now, we want to give it the argument of “deploy”. This will run a simple build of the project, nothing huge. This will test if your code compiles properly, but IntelliJ will also throw errors at you once you do something you’re not supposed to.

The output will look something like this when you have errors.



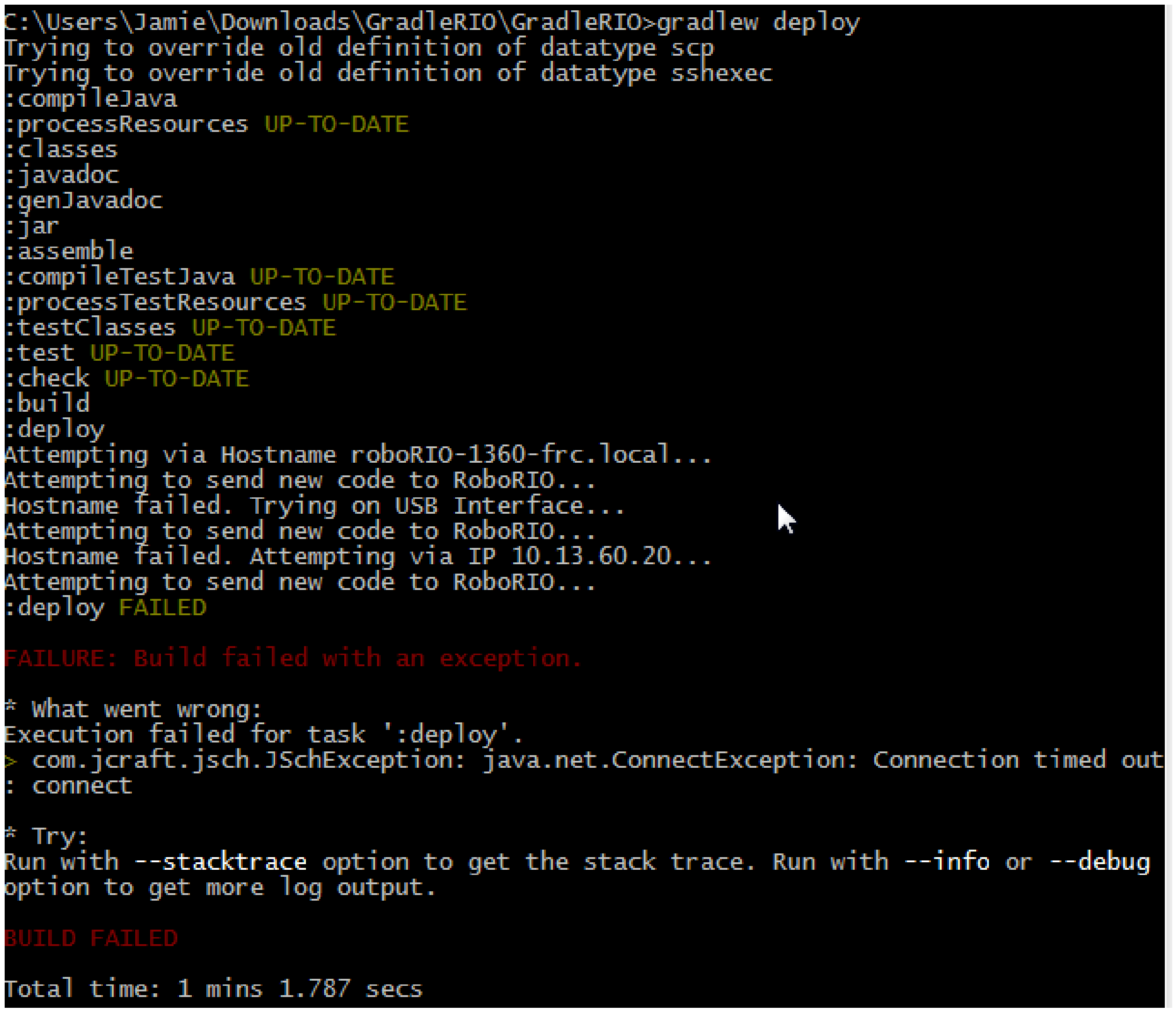
```
C:\Users\Jamie\Downloads\GradleRIO\GradleRIO>gradlew deploy
Trying to override old definition of datatype scp
Trying to override old definition of datatype sshexec
:compileJava
C:\Users\Jamie\Downloads\GradleRIO\GradleRIO\src\main\java\org\usfirst\frc\team1
360\Robot.java:10: error: ';' expected
        super.robotInit()
                         ^
1 error
:compileJava FAILED

FAILURE: Build failed with an exception.

* What went wrong:
Execution failed for task ':compileJava'.
> Compilation failed; see the compiler error output for details.

* Try:
Run with --stacktrace option to get the stack trace. Run with --info or --debug
option to get more log output.

BUILD FAILED

Total time: 2.971 secs

C:\Users\Jamie\Downloads\GradleRIO\GradleRIO>
```

Neat! It even gives descriptions and Java syntax validation!

Now, when you have no errors, it goes ahead and attempts to connect to the roboRIO via mDNS, then static IP as configured in the build.gradle file, and lastly falls back to the direct USB connection.

An example of such is below. Please ignore the errors about not finding a roboRIO, I don’t have one personally, and it’s quite expensive to buy.

Jamie Sinn



```
C:\Users\Jamie\Downloads\GradleRIO\GradleRIO>gradlew deploy
Trying to override old definition of datatype scp
Trying to override old definition of datatype sshexec
:compileJava
:processResources UP-TO-DATE
:classes
:javadoc
:genJavadoc
:jar
:assemble
:compileTestJava UP-TO-DATE
:processTestResources UP-TO-DATE
:testClasses UP-TO-DATE
:test UP-TO-DATE
:check UP-TO-DATE
:build
:deploy
Attempting via Hostname roboRIO-1360-frc.local...
Attempting to send new code to RoboRIO...
Hostname failed. Trying on USB Interface...
Attempting to send new code to RoboRIO...
Hostname failed. Attempting via IP 10.13.60.20...
Attempting to send new code to RoboRIO...
:deploy FAILED

FAILURE: Build failed with an exception.

* What went wrong:
Execution failed for task ':deploy'.
> com.jcraft.jsch.JSchException: java.net.ConnectException: Connection timed out
: connect

* Try:
Run with --stacktrace option to get the stack trace. Run with --info or --debug
option to get more log output.

BUILD FAILED

Total time: 1 mins 1.787 secs
```

That's it! You now know how to install IntelliJ, install the JDK and set the JAVA_HOME variable, as well as install the GitHub client (The person presenting this will give a tutorial on how to use it, or if you're simply reading this yourself, you can read the GitHub documentation.)

If you have any questions, please feel free to email me at james.sinn@sinndevelopment.com

Jamie Sinn