**Gradle Tutorial Notes:**  
<https://spring.io/guides/gs/gradle/#scratch>  
  
**Project Structure:**  
PRJ\_ROOT/src/main/java  
Put all your java code into the java folder.  
Then on command line change directory to PRJ\_ROOT.  
Then:  
gradle build

**To run on command line:**cd C:\Users\jmadison\Desktop\LearnGradel\PrjRoot\build\classes\main  
java –cp . hello.HelloWorld  
  
**This will work, till you get dependencies, like the YODA time library.**  
You will need something similiar to maven’s mvn exec:java –Dexec.mainClass=”hello.HelloWorld”  
  
This post seems to have help on that:  
http://stackoverflow.com/questions/16350757/  
The easiest solution is to use the [Application Plugin](http://www.gradle.org/docs/current/userguide/application_plugin.html), which among other things provides a run task. To make the main class configurable from the command line, you'll have to set mainClassName to the value of some system (or project) property, and then pass that property from the command line:

**apply plugin: "application"**

**mainClassName = System.getProperty("mainClass")**  
*Now you can run the application with gradle run -DmainClass=thornydev.App.*

http://stackoverflow.com/questions/21358466/   
**CMD: gradle run -DmainClass=runclass.RunClass**

|  |  |
| --- | --- |
| Also From: http://stackoverflow.com/questions/21358466/ Use JavaExec. As an example put the following in build.gradle  task execute(type:JavaExec) {  main = mainClass  classpath = sourceSets.main.runtimeClasspath  }  *To run on CMD:* **gradle -PmainClass=Boo execute** *RESULT:*   |  | | --- | | $ gradle -PmainClass=Boo execute :compileJava :compileGroovy UP-TO-DATE :processResources UP-TO-DATE :classes :execute I am BOO! |   There is no mvn exec:java equivalent in gradle,  you need to either  1: **apply the application plugin**  or  2: **have a JavaExec task.**  **Comment to this solution:** -PmainClass=runclass.RunClass execute  My hacks: Try: *CMD:* **gradle -PmainClass=runclass.HelloWorld execute** |

**JMIM NOTES:  
Problem with PMainClass execute:  
I am not sure of the context of what people are writing.  
What is verbatam keywords, and what are project-specific paths.  
For the user’s example.**

|  |
| --- |
| //Inside build.gradle file:  //http://stackoverflow.com/questions/21358466  //will enable us to use: gradle -PmainClass=Boo execute  //on the command line.  // http://stackoverflow.com/questions/21358466/  task execute(type:JavaExec) {  //main = mainClass  //classpath = sourceSets.main.runtimeClasspath  main = HelloWorld  classpath = hello.HelloWorld  }  **//Case and point: is it main=mainClass, or main=helloWorld //If helloWorld, is it surrounded by quotes or not?** |

**JMIM NOTE:** Got it working using CMD: gradle run with this gradle file:  
Note that mainClassName is **FULLY QUALIFIED** class name. Starting from the  
package that HelloWorld belongs to.

|  |
| --- |
| **apply plugin: 'java'**  **apply plugin: "application"**  **//You just need to use the Gradle Application Plugin:**  **//http://stackoverflow.com/questions/21358466/**  **//And then simply: gradle run**  **mainClassName = "hello.HelloWorld"**  **repositories {**  **mavenCentral()**  **}**  **//Source compatibility needs to refer to the version**  **//Of the JDK you are using.**  **sourceCompatibility = 1.7**  **targetCompatibility = 1.7**  **dependencies {**  **compile "joda-time:joda-time:2.2"**  **}**  **jar {**  **baseName = 'gs-gradle'**  **version = '0.1.0'**  **}** |

JMIM NOTES:  
gradle build  
is failing on me.  
Something about changing source sets in build.gradle  
http://stackoverflow.com/questions/24924932/  
  
This answer is great, however it doesn’t address that  
gradle build  
on command line no longer works:

<http://solvedstack.com/questions/gradle-to-execute-java-class-without-modifying-build-gradle>