We recommend you use Linux or MacOS, rather than Windows, please use Hadoop 2.8.5 to match the Hadoop in docker.

Contents

1. Run the WordCount example (Windows 10+IDEA)

You can follow this guide to run the first MapReduce example: **WordCount**.

Step1:

1.1 Make sure you have installed JDK on Windows

1.2 Make sure the installation path do not have a space, e.g. **C:/Program Files/Java/jdk1.8.0_144** will cause some errors.

Step2:

2.1 Download the Hadoop and install it on windows. (here we use Hadoop 2.7.4 as the example, you can download 2.8.5 to match the Hadoop version in docker)

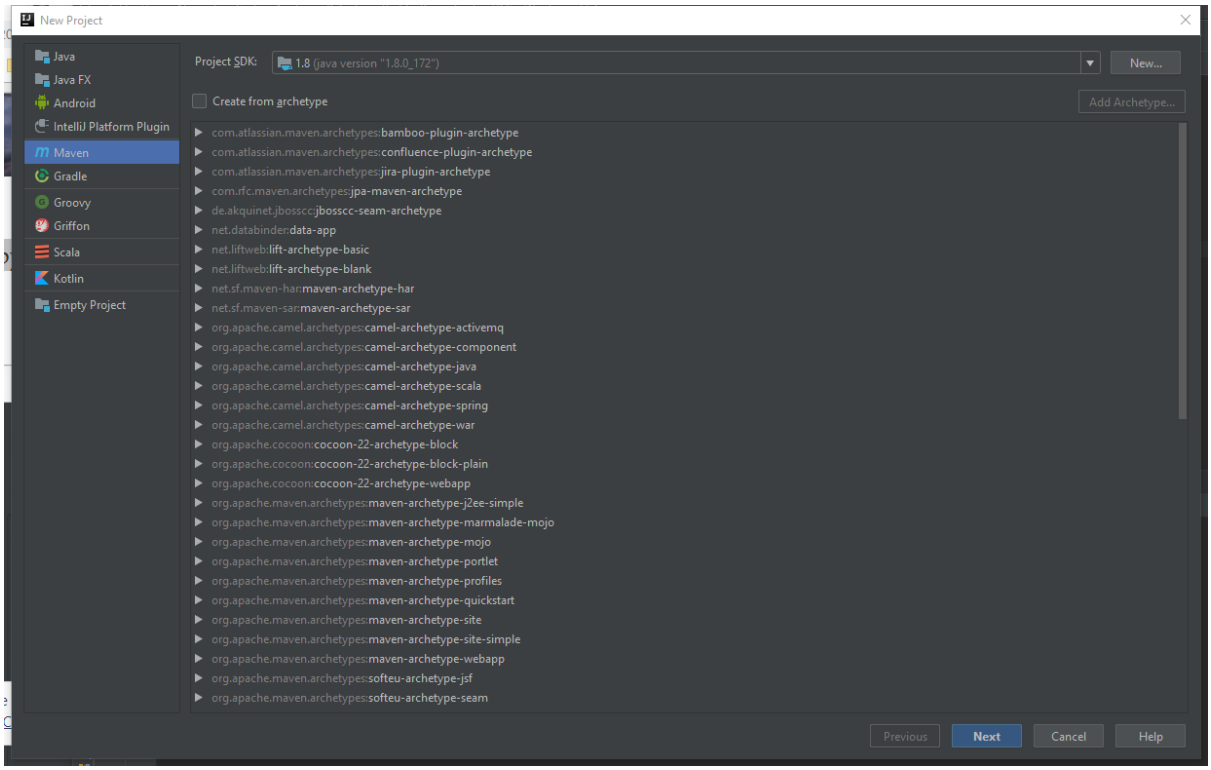
2.2 Set the Environment variables:
HADOOP_BIN_PATH : %HADOOP_HOME%\bin
HADOOP_HOME : E:\soft\hadoop-2.7.4 (your Hadoop's installation path)
Add **%HADOOP_HOME%\bin;%HADOOP_HOME%\sbin** to your system Path

2.3 The downloaded binary files from Apache web site do not contain some Windows native components(such as winutils.exe, hadoop.dll…). I will upload these components (tested for Hadoop 2.7.4 & 2.8.5) in IVLE, you can move them into your **hadoop/bin.**

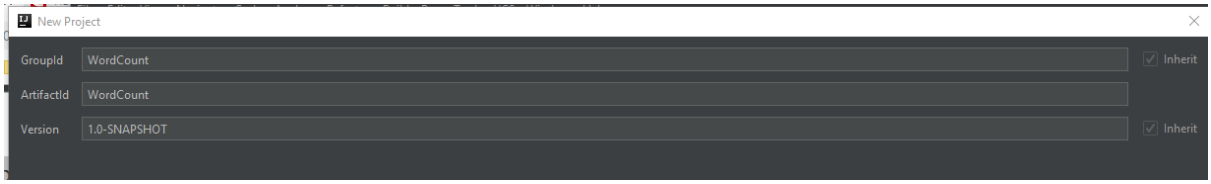
Step3: (this step is same if you use idea, not just for windows)

3.1 Installing Intellij IDEA as the IDE
3.2 Start Intellij IDEA as Administrator

3.3 Create Maven project by File -> New -> Project -> Maven, please see below picture:



3.3 Next, specify GroupId, ArtifactId as WordCount.

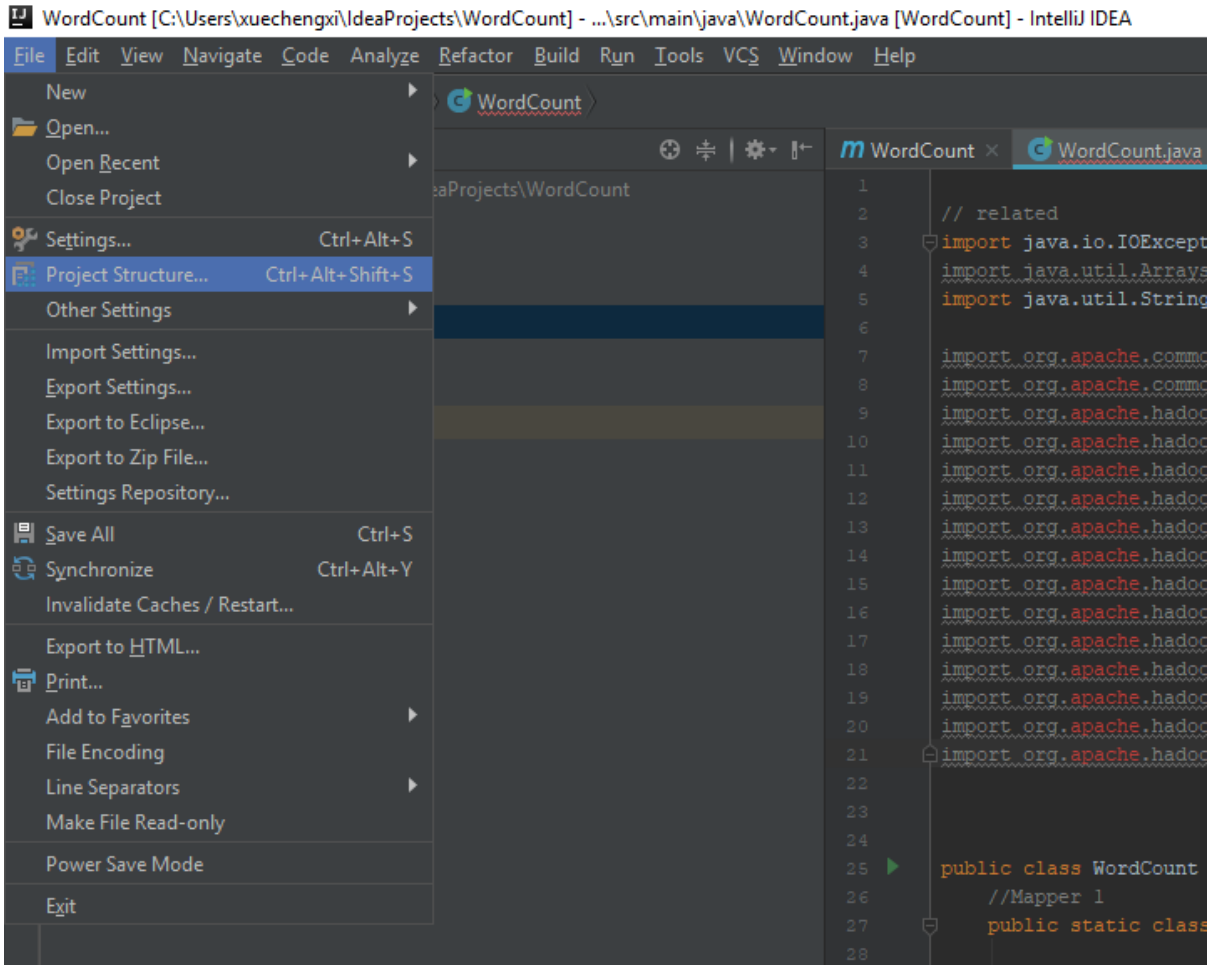


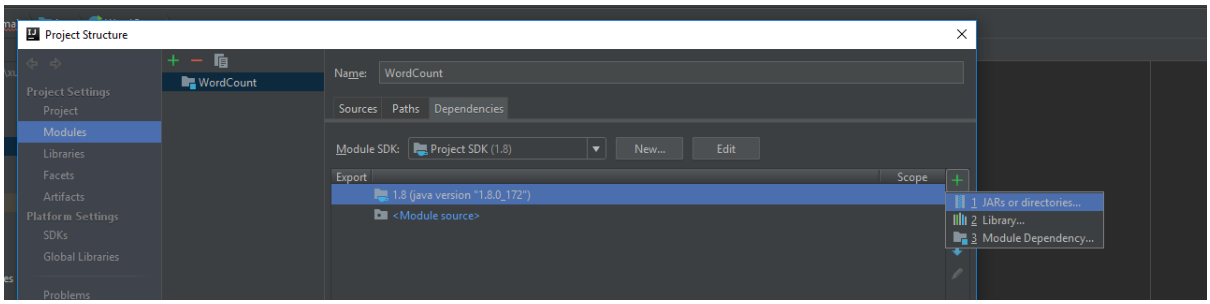
3.4 Next and click Finish.

3.5 Create a Java class on left panel of Intellij IDEA, click WordCount -> src -> main ->java, then right click to choose New ->Java Class, type WordCount as class name.

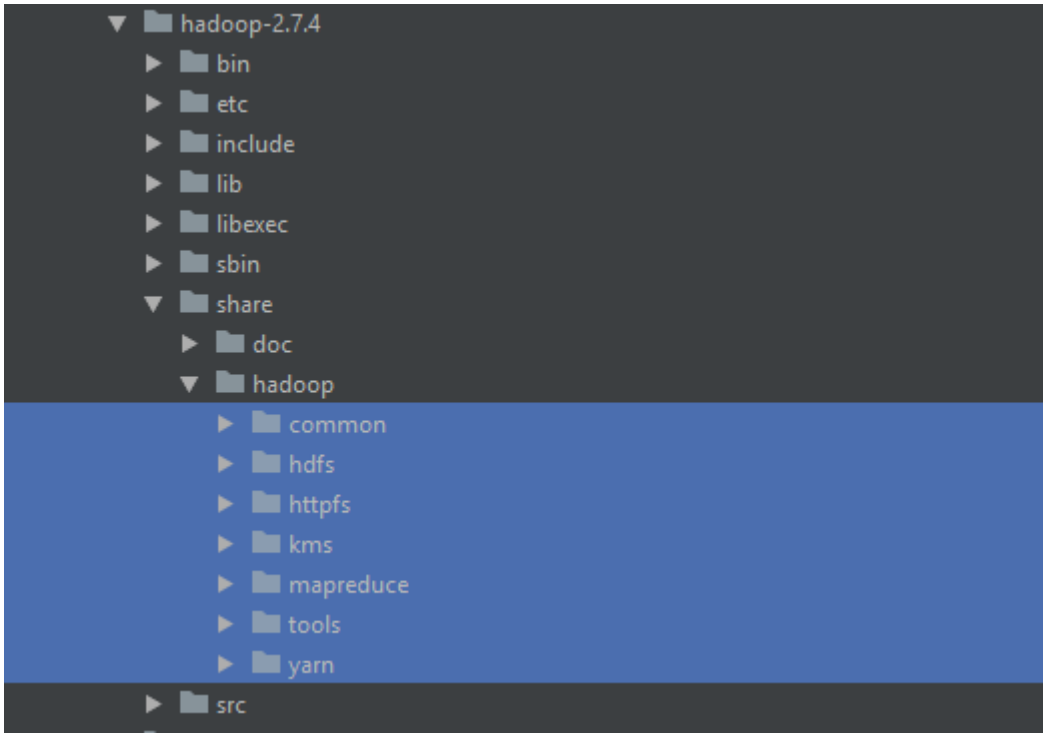
Add the provided code in **WordCount.java** (you can remove the first line: package wordcountpkg; if you do not create a package)

3.6 you will see some errors, don't worry. We need to add Hadoop Library. click File -> Project Settings -> Modules -> Dependencies -> +.

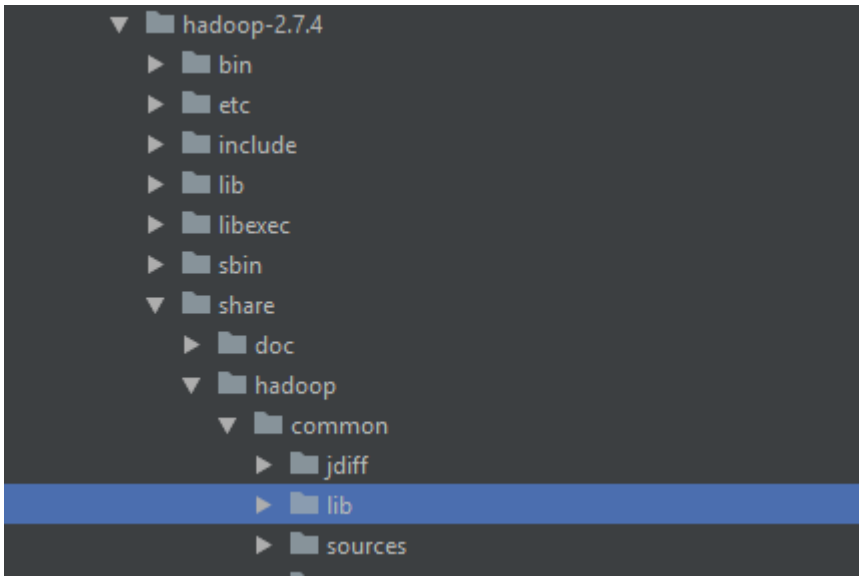




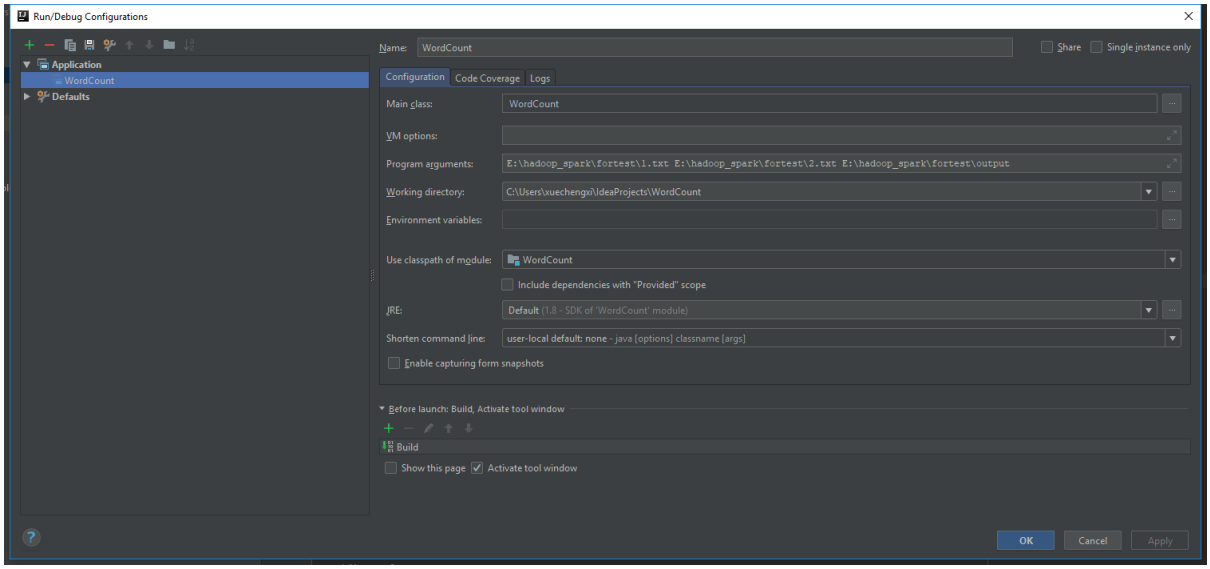
Find you Hadoop, add all these libraries.



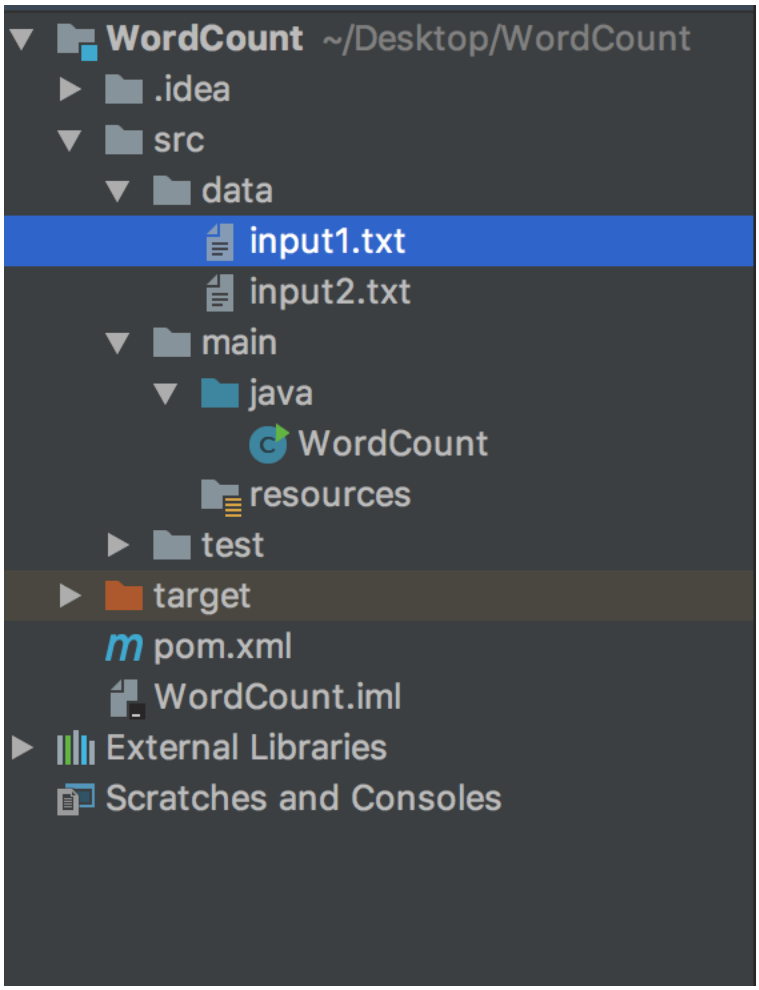
Also add this lib.

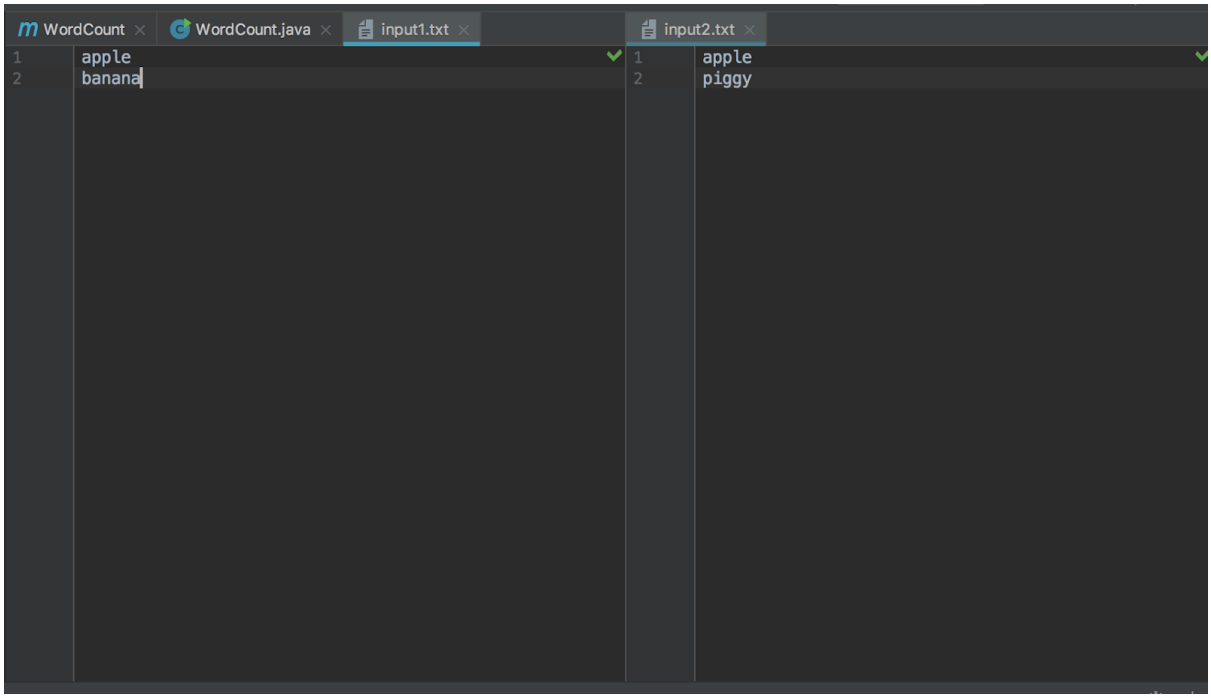


3.7 Edit Running Configuration. Click Run -> Edit Configuration.. -> + , see the below picture:

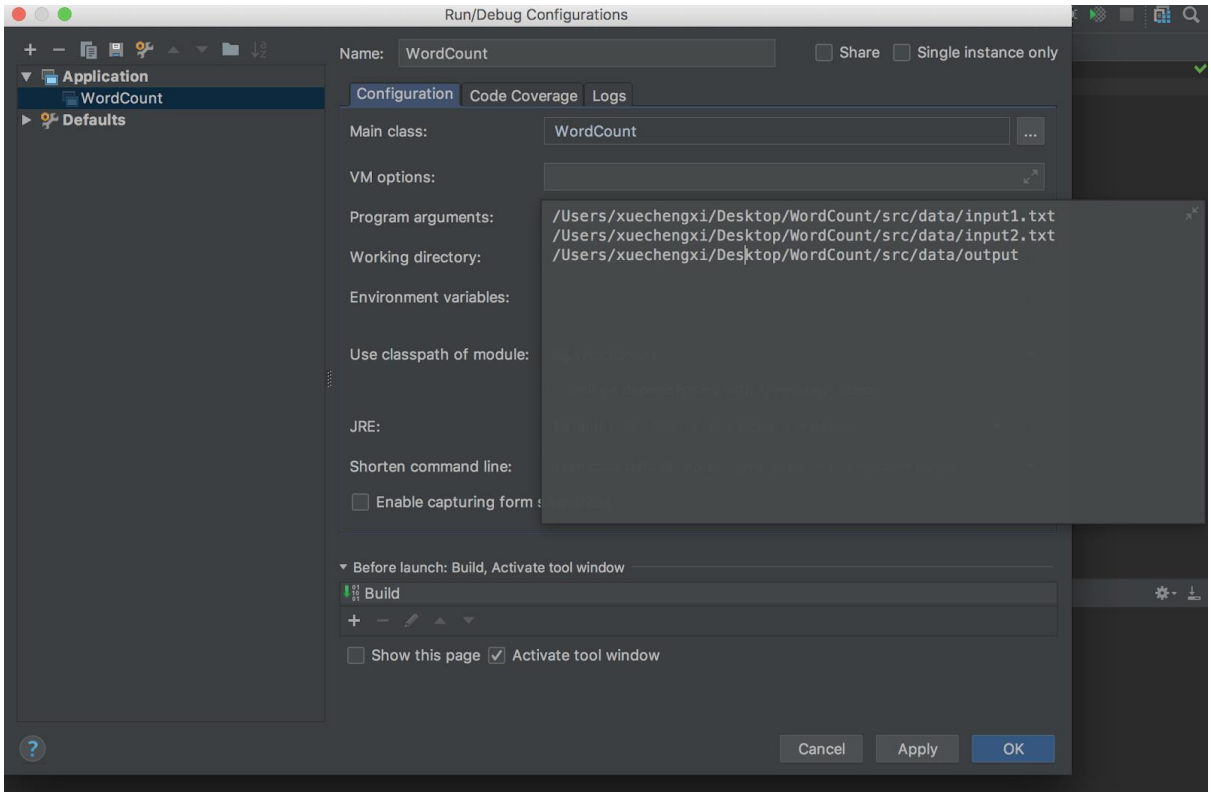


You need to use your own Program arguments. Two input files and the output path.





The arguments:



You do not need to create the output directory; Hadoop application will create the output directory for you automatically.

3.8 Run the application, if no error, you will see the result in your output file.

if you want to run the Hadoop and use HDFS on windows, here are some materials for you **(untested)**: 1.**https://wiki.apache.org/hadoop/Hadoop2OnWindows**
2.https://github.com/MuhammadBilalYar/Hadoop-On-Window/wiki/Step-by-step-Hadoop-2.8.0-installation-on-Window-10
3.http://hadooponwindows10.blogspot.com/2016/07/apache-hadoop-271-installation-on-win10.html

2. Run the WordCount example (Linux Ubuntu 14+ IDEA)

You can follow this guide to run the first MapReduce example: **WordCount**.

Step1:

1.1 Make sure you have installed JDK (jdk1.8 has been tested)

Step2:

2.1 Download the Hadoop 2.8.5 and install it.
you can read: **http://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html**

Step3:

Same as previous step3 on windows.

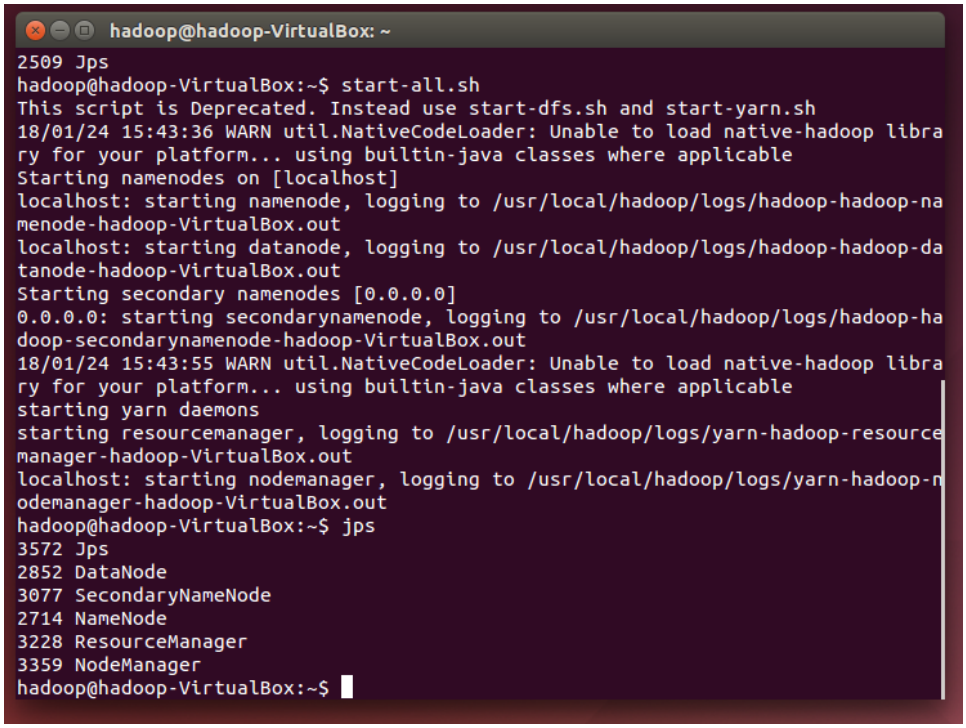
3. Run the WordCount example (Linux Ubuntu + Eclipse)

You can use eclipse with Hadoop plugin. In this example, you can access the hdfs directly.

Step1:

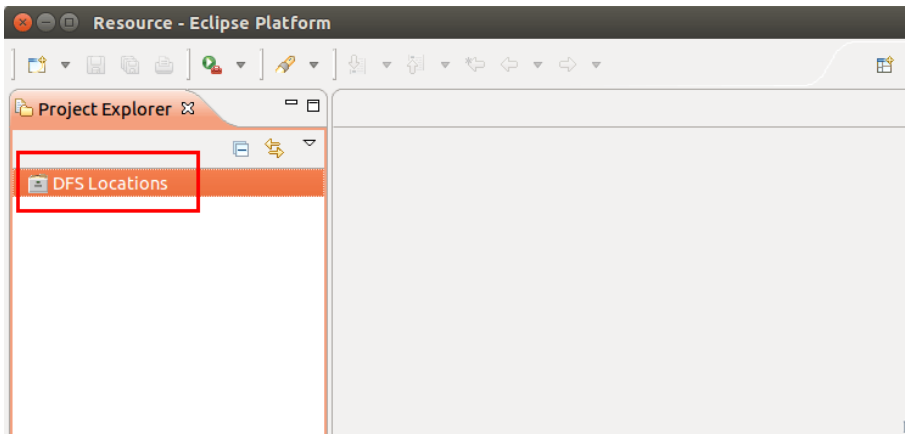
1.1 Download eclipse

1.2 Put “hadoop-eclipse-plugin-2.8.3.jar” (we assume Hadoop is 2.8.5 and this plugin can work, or



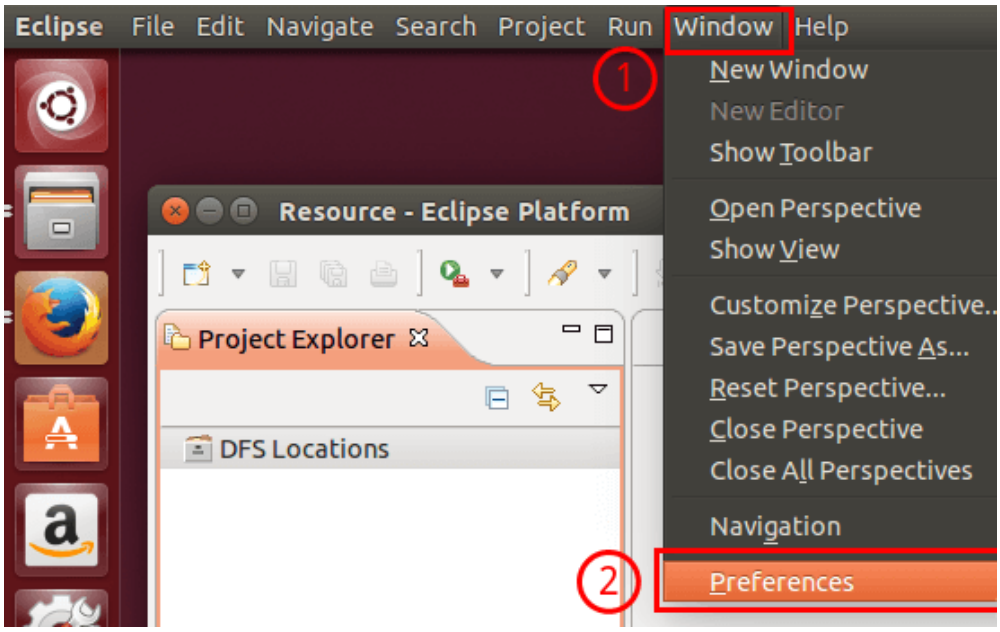
you need to find corresponding plugins) into eclipse/dropins.

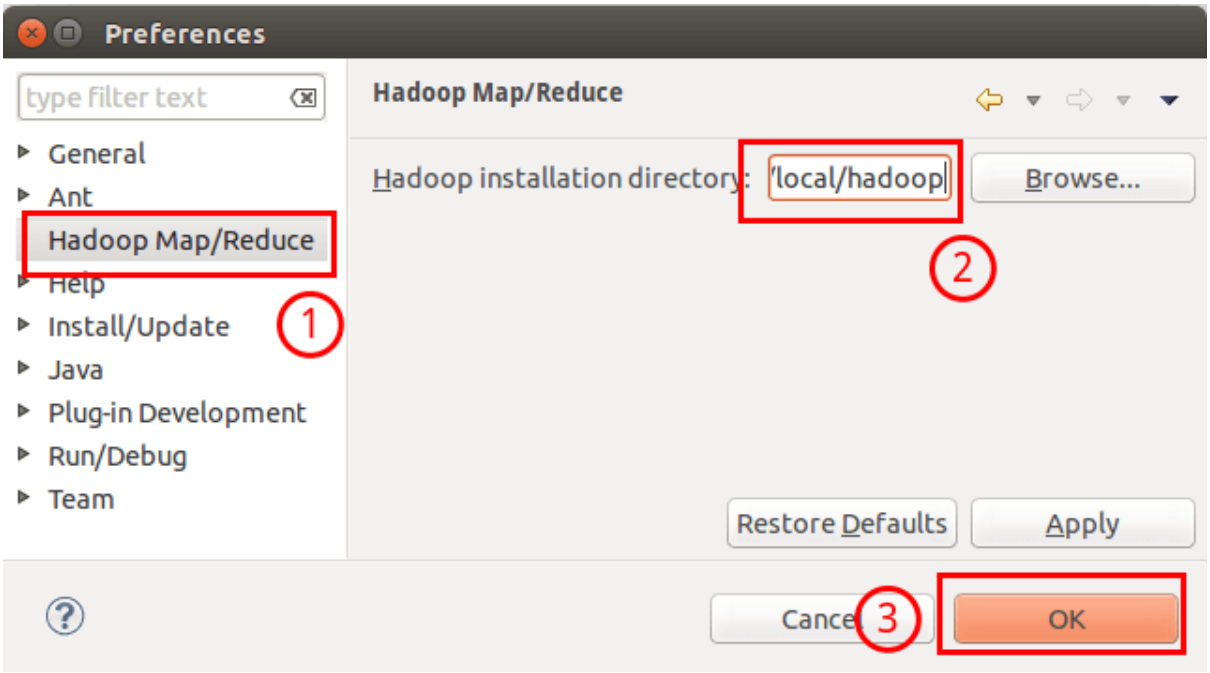
1.3 Restart eclipse (make sure you have started the Hadoop, /sbin/start-all.sh)



You will see the dfs locations

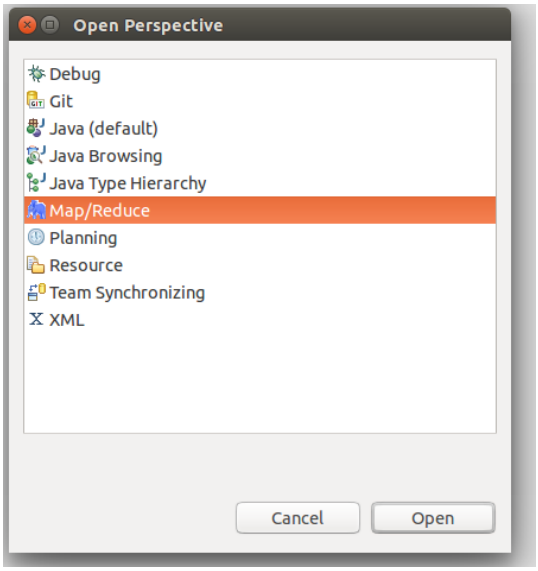
1.4 Click “Windows” then “preferences”



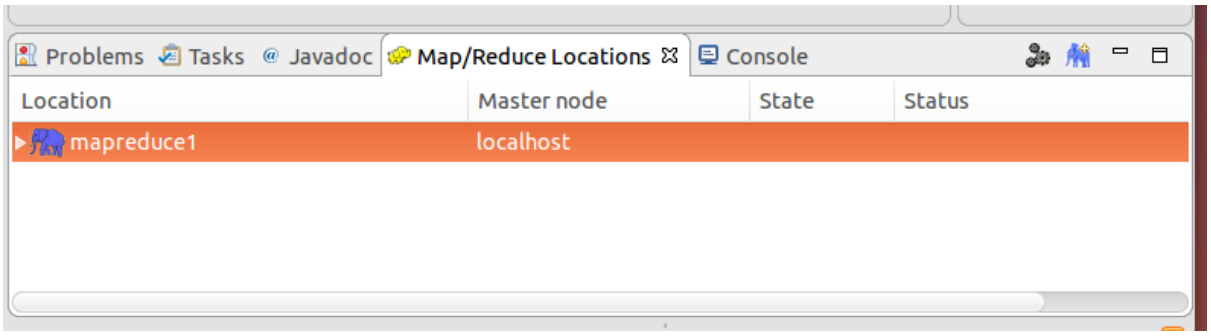


1.5 Choose your Hadoop path

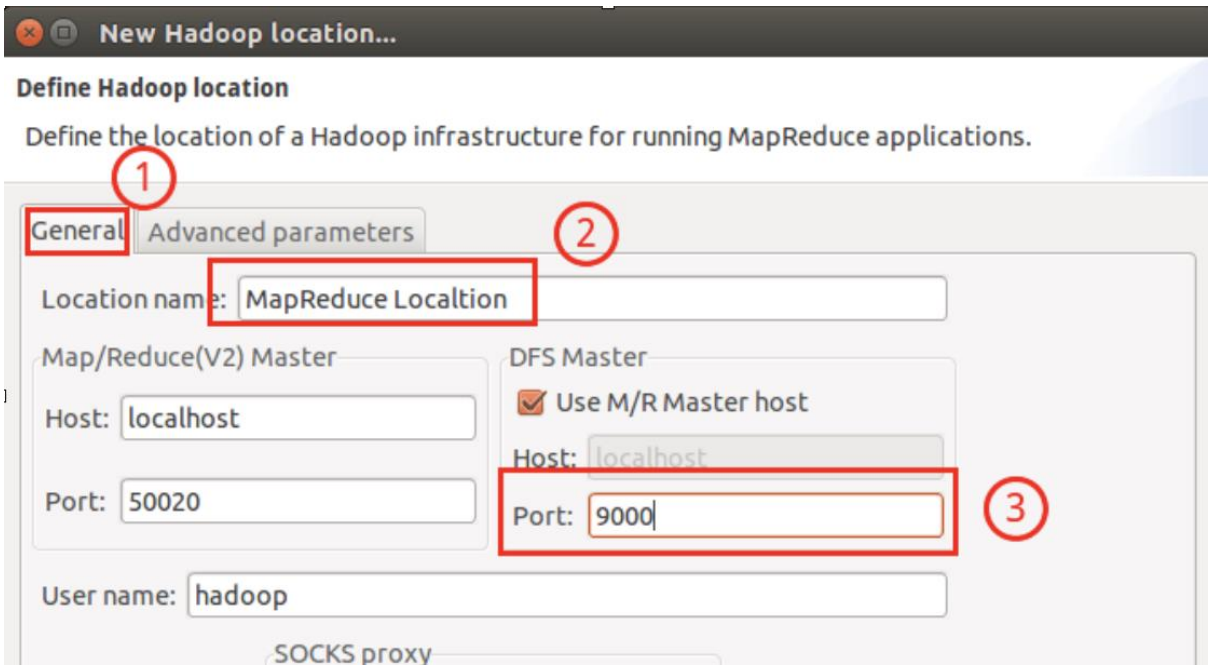
1.6 Click “windows” —> ”perspective” —> ”open perspective”—> ”other” —> ”Map/Reduce”



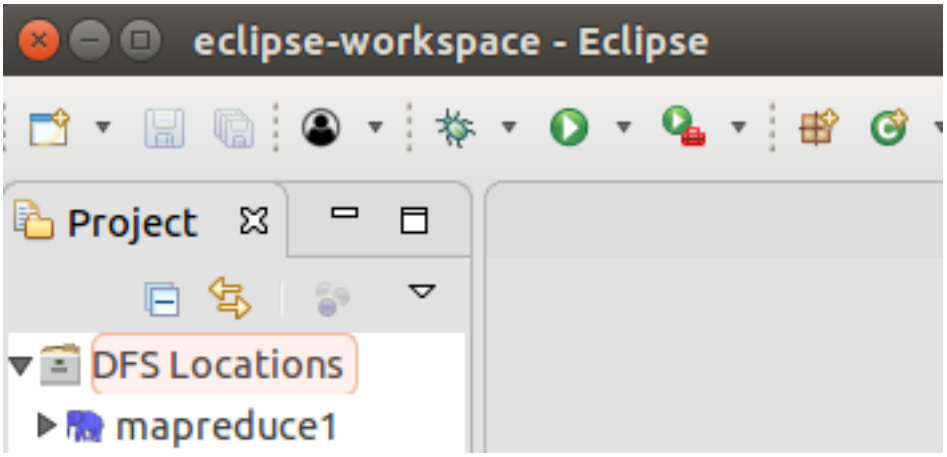
1.7 Connect to Hadoop, right click the window below the Map/Reduce Location, choose New Hadoop Location



1.8 Choose the location name, fill up the DFS Master port. (in core-site.xml)

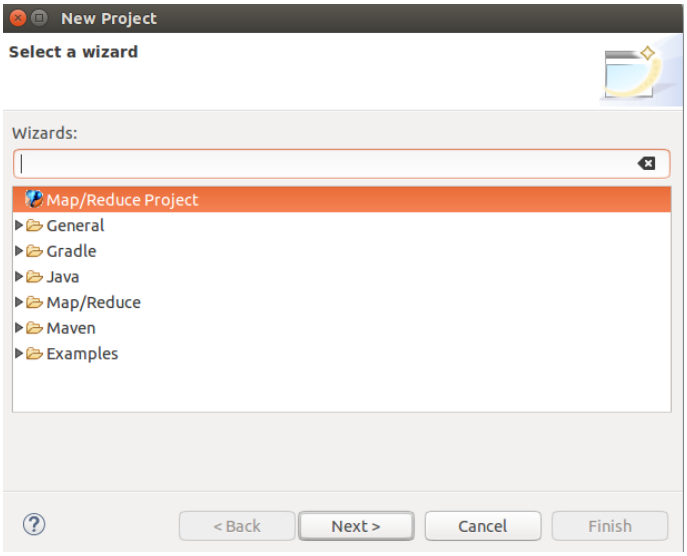


1.9 Then you can see a blue elephant!



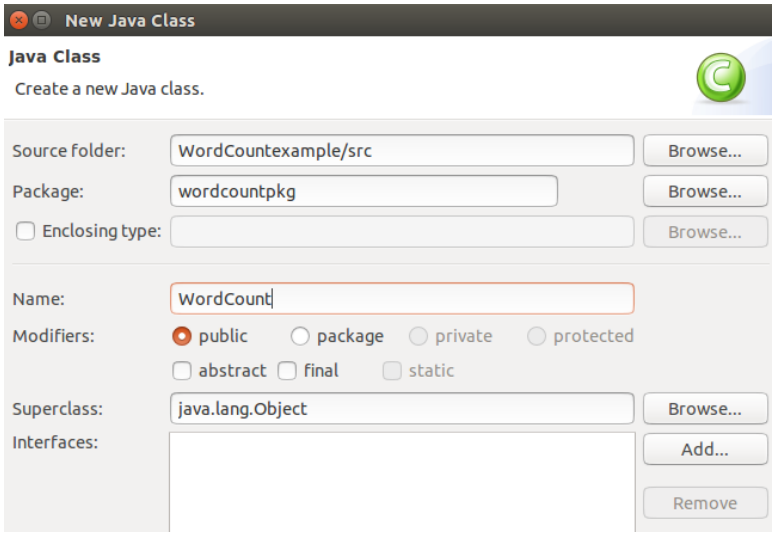
Step2

2.1 Now you can create a project click file—>new—>project—>Map/Reduce Project—>next—>

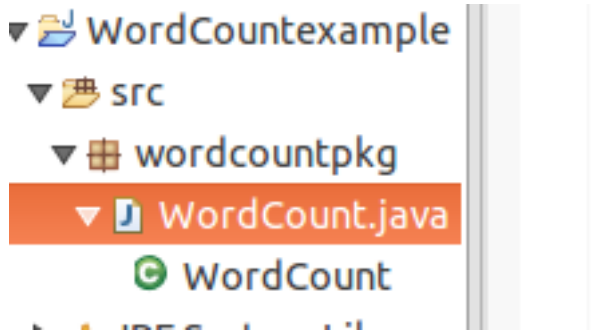


Project name—>finish. You can see your project now.

2.2 Right Click your project—> new —>class, you need to fill up the package and name

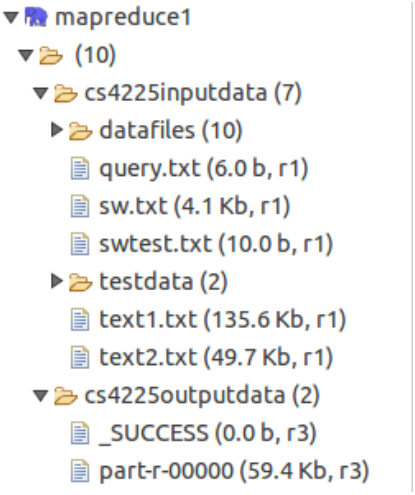


2.3 Copy WordCount.java into your java file.(you need to change some details if using different



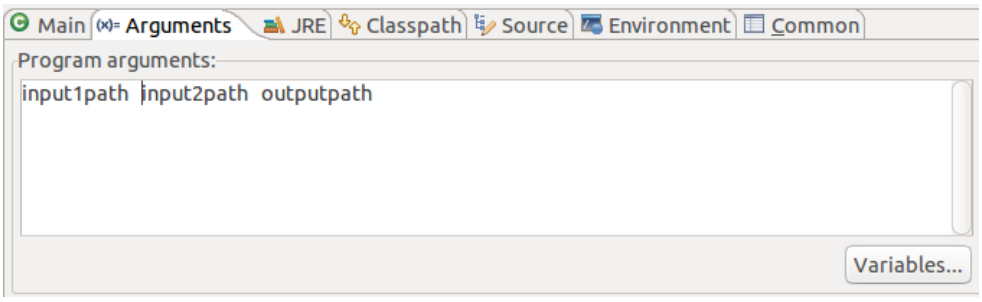
configuration like paths)

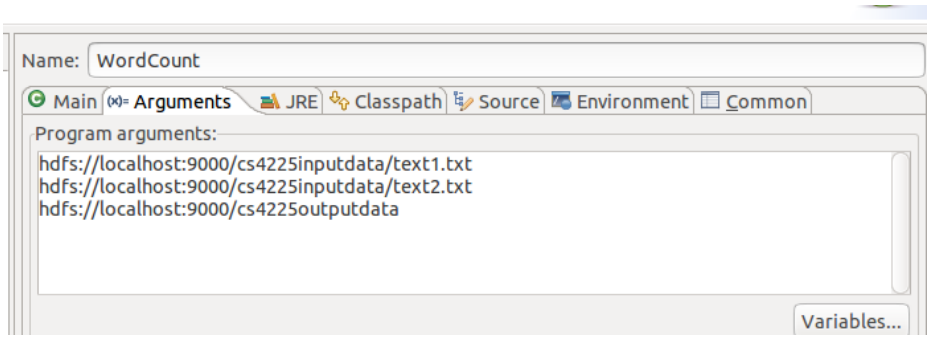
2.4 Using Hadoop commands (put) to put our data into hdfs (like: hadoop fs -put data1.txt /data/inout) you can find them here (need to refresh after every operation)



2.5 Right click your java file —>run as—>run configurations—>Arguments

Here you need to fill up your arguments, for example:





2.6 Run and you will see the result.

4. Run the WordCount example (MacOS + IDEA)

You can follow this guide to run the first MapReduce example: **WordCount**.

Step1:

1.1 Make sure you have installed JDK (jdk1.8 has been tested)

Step2:

2.1 Download the Hadoop and install it.

I recommend you install Hadoop 2.8.5 to match the Hadoop version in docker.

https://hadoop.apache.org/docs/r2.8.5/hadoop-project-dist/hadoop-common/SingleCluster.html

You can use Homebrew to make it very easy, but "brew install Hadoop" (it will install the latest version (3.1.1), so pay attention to the difference between Hadoop 3.* and Hadoop 2.*)

You can use this guide to build a single cluster Hadoop system (3.1.1) on mac to test:
https://hadoop.apache.org/docs/r3.1.1/hadoop-project-dist/hadoop-common/SingleCluster.html

Note: difference between Hadoop 3.* and Hadoop 2.*: 1. default ports are different,
2. start-yarn.sh, and a lot of other differences.
3. If you use brew install, the default installation path for Hadoop is /usr/local/Cellar/Hadoop/3.1.1/, and the /etc is in /libexec/

Step3:

Same as previous step3 on windows.