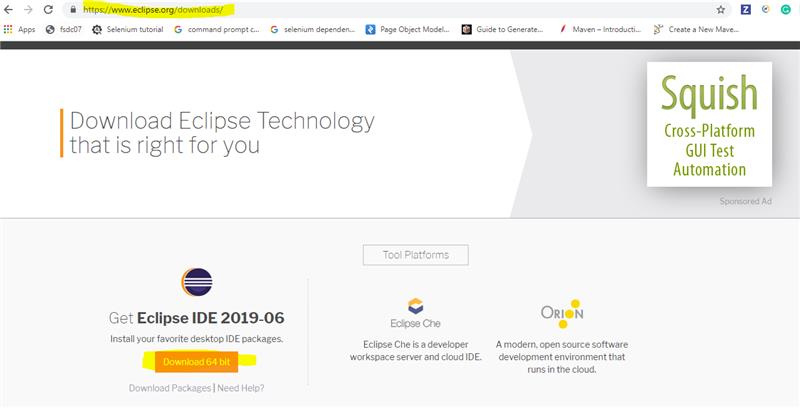
**Step 1:** Link to Download Java:

[Java SE - Downloads | Oracle Technology Network | Oracle](https://www.oracle.com/java/technologies/javase-downloads.html)

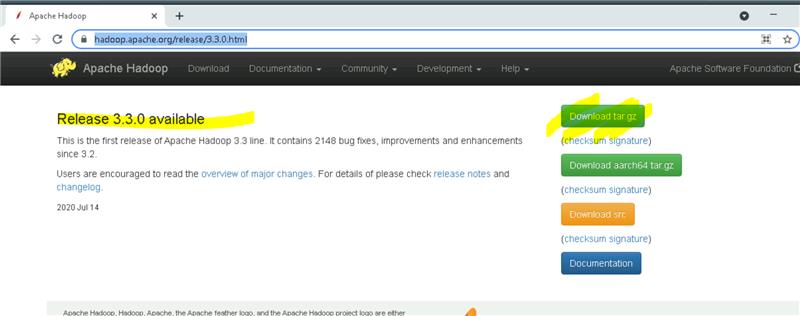


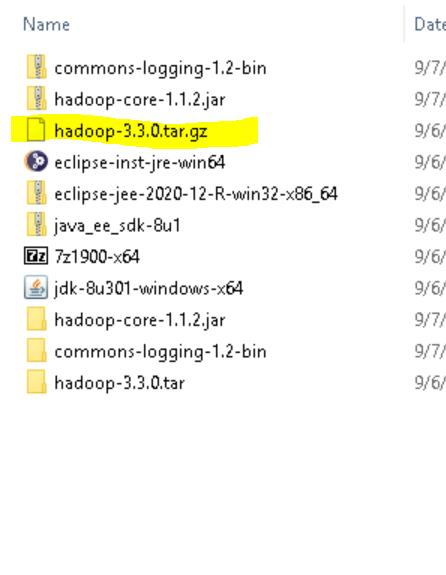
**Step2:** link to download Eclipse IDE

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



**Step 3:** Link to download Hadoop





Unzip with 7zip

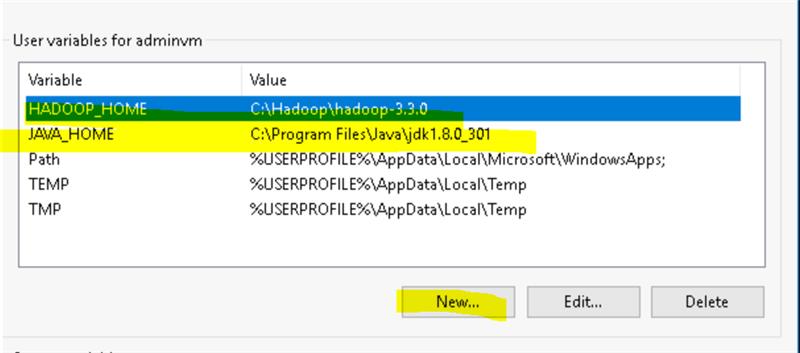
**Step4:**

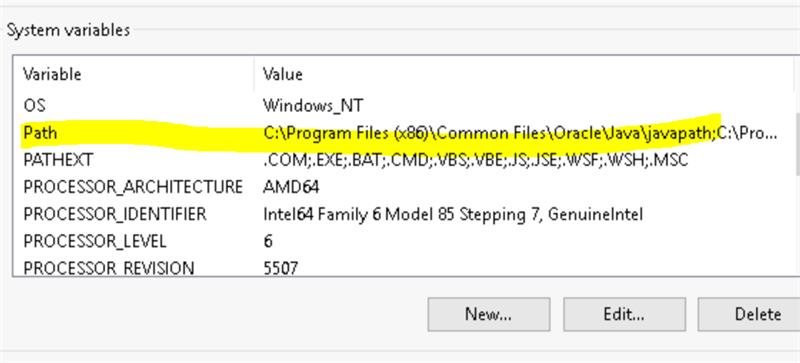
Download all files from winutils

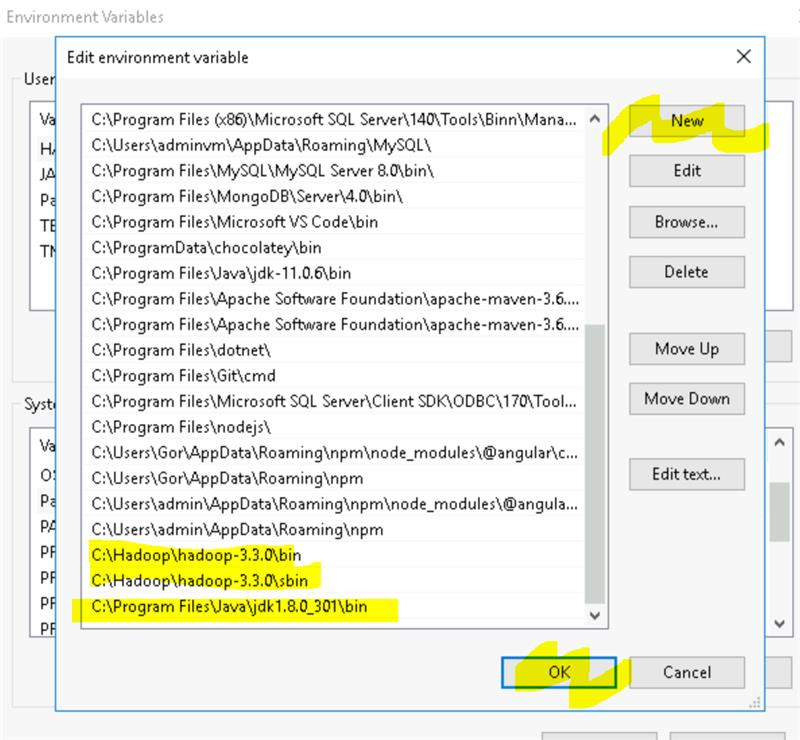
(<https://github.com/kontext-tech/winutils/tree/master/hadoop-3.3.0/bin>)

Put all these files in C:\Hadoop\hadoop-3.3.0\bin

**Step 5:**  Add environment variables (type Env in start menu and select setup system environment variables -> Environment Variables)







**Step 6:** Go to C:\Hadoop\hadoop-3.3.0

Create a new folder and name it as **data**. Inside data create 2 new folders. Rename them as **datanode** and **namenode**.

**Step 7**:Go to C:\Hadoop\hadoop-3.3.0\etc

Edit the following files and add the given text inside configuration tag of each file as follows:

**core-site.xml**

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost:9000</value>

</property>

**hdfs-site.xml**

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>PATH~1\namenode</value>

<final>true</final>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>PATH~2\datanode</value>

<final>true</final>

</property>

Replace PATH~1 and PATH~2 with C:\Hadoop\hadoop-3.3.0\data

(or appropriate path for namenode and datanode folders in data folder)

**mapred-site.xml**

<property>

<name>yarn.app.mapreduce.am.env</name>

<value>HADOOP\_MAPRED\_HOME=$HADOOP\_HOME</value>

</property>

<property>

<name>mapreduce.map.env</name>

<value>HADOOP\_MAPRED\_HOME=$HADOOP\_HOME</value>

</property>

<property>

<name>mapreduce.reduce.env</name>

<value>HADOOP\_MAPRED\_HOME=$HADOOP\_HOME</value>

</property>

**yarn-site.xml**

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

**hadoop-env.cmd**

Paste jdk folder path at JAVA\_HOME variable

JAVA\_HOME= "C:\Program~1\Java\jdk1.8.0\_301" (or appropriate path)

**Step 8:**

Go to C drive. Create a new folder **data**. Open the folder and create a text file **sample.txt**. Enter a few words in the sample.txt file and save it.

**Step 9:**

Open command prompt and change directory to C:\Hadoop\hadoop-3.3.0

Run the command hadoop namenode -format

Then run the command start-all.cmd

4 daemons will start running. Ensure that they are not throwing any errors.

**Step 10:**

Now change directory to C:\Hadoop\hadoop-3.3.0\sbin (cd sbin)

Type the command hdfs dfs -mkdir /input (This will create a new directory **input** in hdfs)

**Step 11:**

Now type the command hdfs dfs -copyFromLocal C:/data/sample.txt /input

(This will copy the sample.txt file from the c drive to hdfs file system)

**Step 12:**

Open a browser and type localhost:9870

Go to Utilities> Browse the file system. You should see the **input** directory there. Inside input directory **sample.txt** should be seen.

**Step 13:**

Now change directory to C:\Hadoop\hadoop-3.3.0 (cd ..)

Now type the command hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.0.jar wordcount /input /output

This will run the jar mapreduce program,which calculates wordcount for content in sample.txt file in inut foldet and stores it in output folder.

**Step 14:**

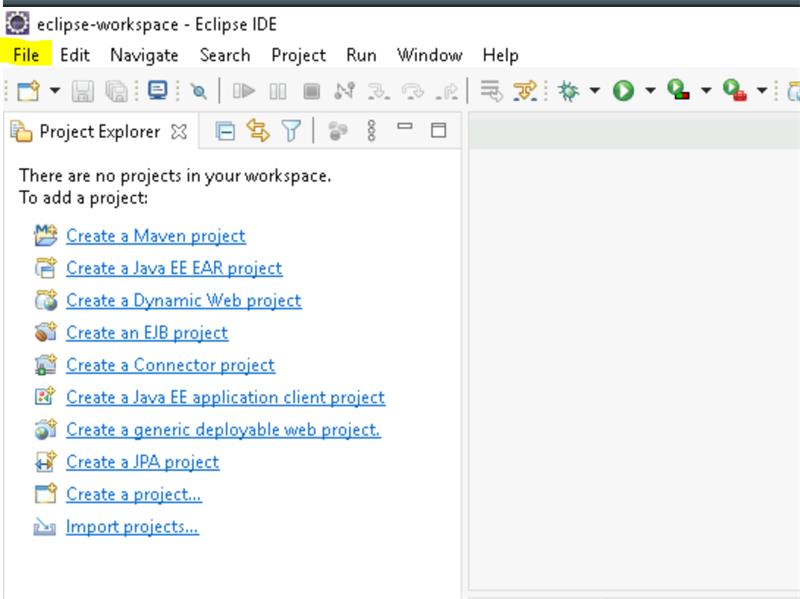
Open a browser and type localhost:9870

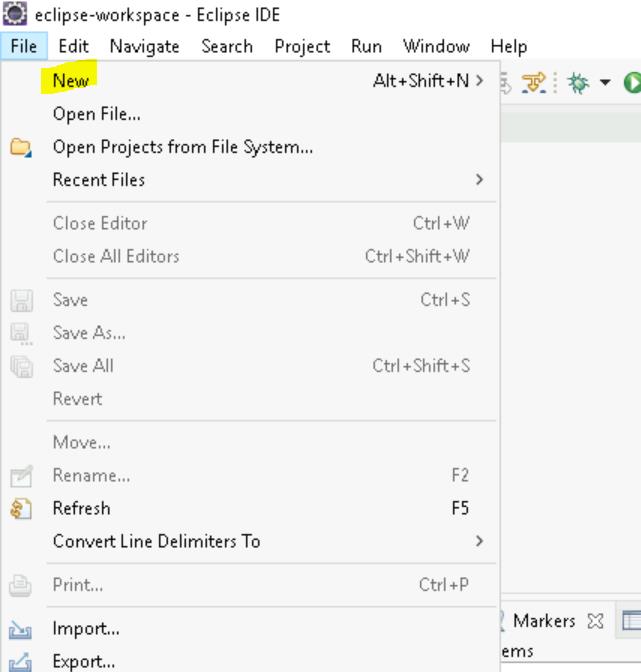
Go to Utlities> Browse the file system. You should see the **output** directory there. Inside input directory 2 files will be created. Where output file with wordcount will be seen.

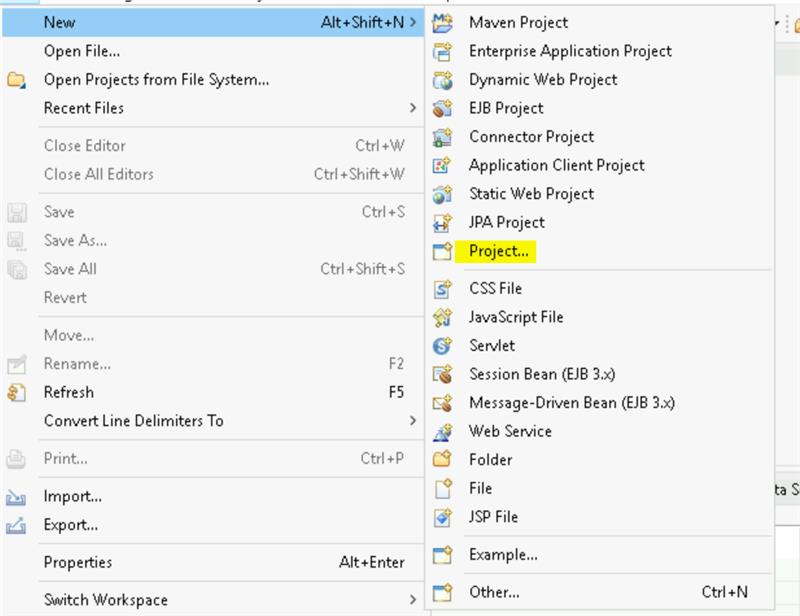
**Creating Map reduce program in Eclipse to calculate character count:**

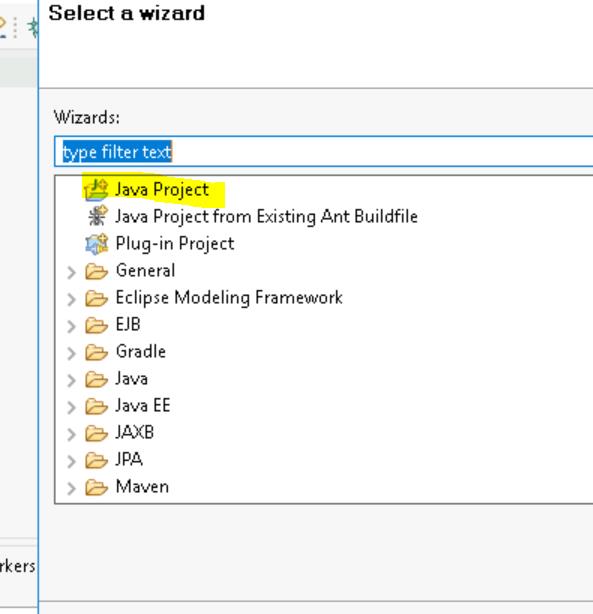
**Step 1:**

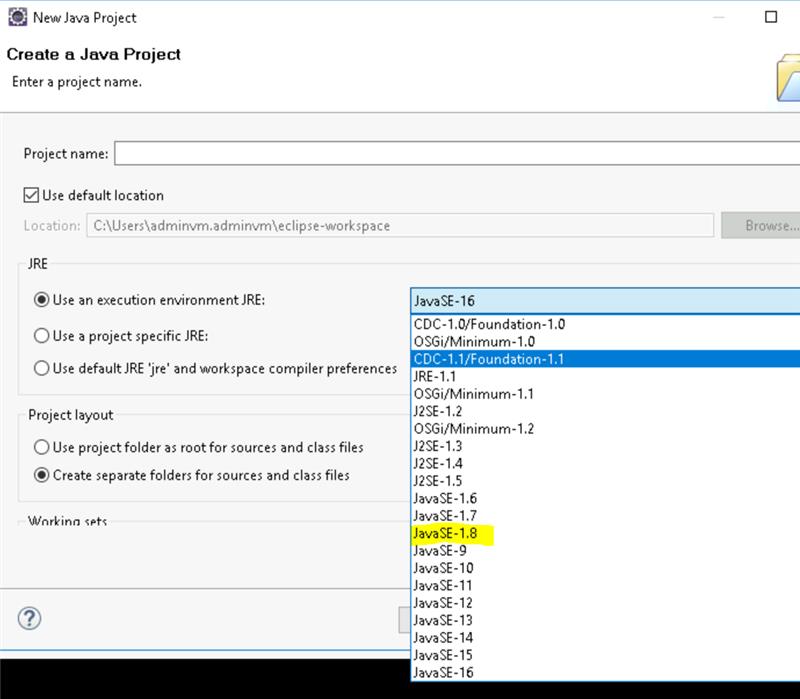
Open Eclipse IDE and create MapReduce project as below:

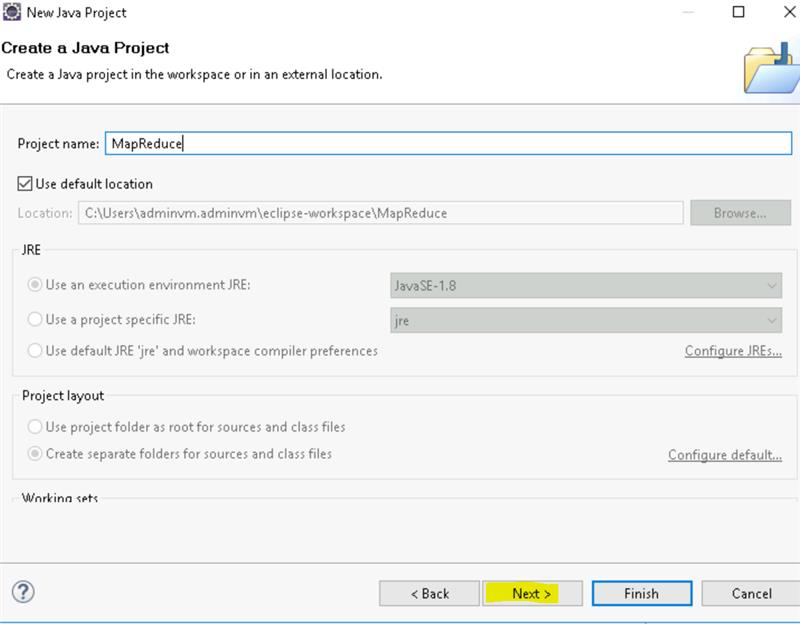


****

****

****





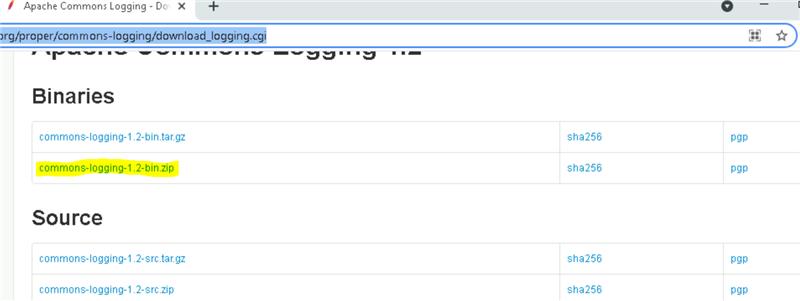
Download Java core 1.2.1 jar

<https://repo1.maven.org/maven2/org/apache/hadoop/hadoop-core/1.2.1/>

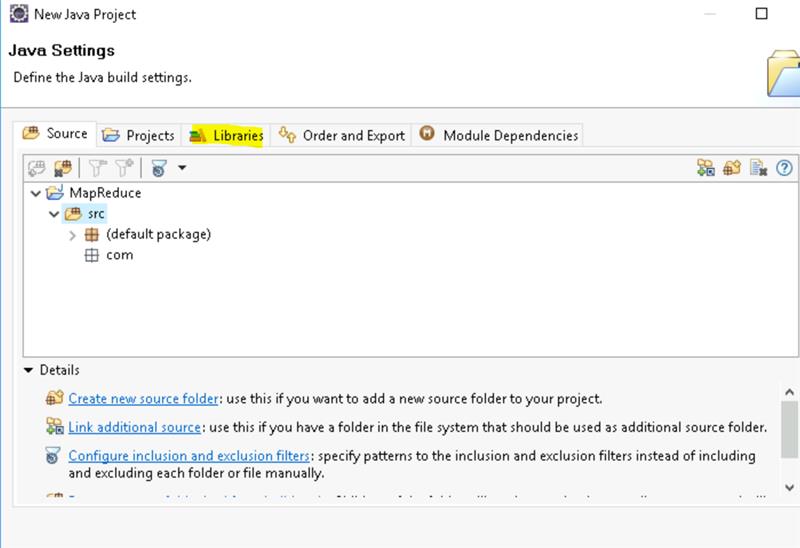


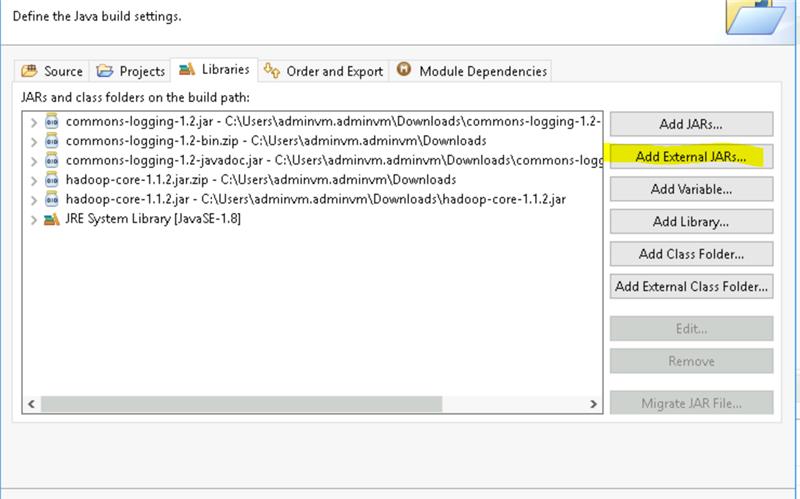
Download apache commons logging --> extract the files

<https://commons.apache.org/proper/commons-logging/download_logging.cgi>

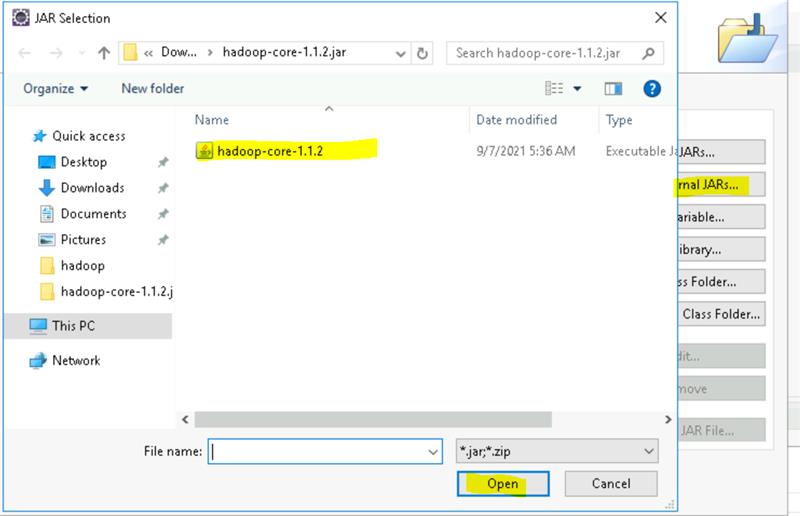


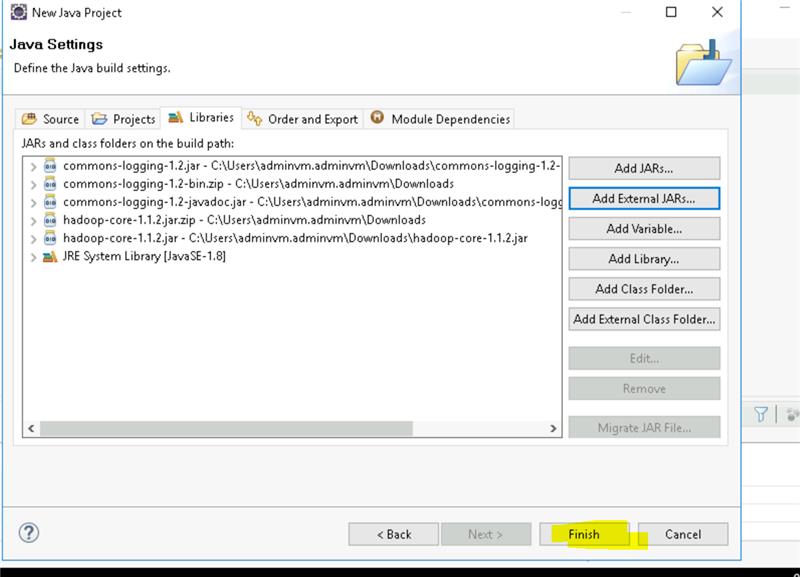
Unzip both files and include them in eclipse libraries as below:





browse for add external jars --> add downloaded lb and jars in commoms logging and hadoop core 1.2





Inside the project **MapReduce** create a package as described below:

Right click on src-> new-> package-> enter package name as **com.cgi** and click on finish.

Now create 3 new classes as described below:

Right click on com.cgi-> new-> class-> enter class name as **WC\_Mapper** and click on finish.

Similarly create classes **WC\_Reducer** and **WC\_Runner** also

Copy the below codes for each of the 3 classes.

**CLASS-1 WC\_Mapper**

​

package com.cgi;

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.MapReduceBase;

import org.apache.hadoop.mapred.Mapper;

import org.apache.hadoop.mapred.OutputCollector;

import org.apache.hadoop.mapred.Reporter;

public class WC\_Mapper extends MapReduceBase implements Mapper<LongWritable,Text,Text,IntWritable>{

public void map(LongWritable key, Text value,OutputCollector<Text,IntWritable> output,

Reporter reporter) throws IOException{

String line = value.toString();

String tokenizer[] = line.split("");

for(String SingleChar : tokenizer)

{

Text charKey = new Text(SingleChar);

IntWritable One = new IntWritable(1);

output.collect(charKey, One);

}

}

}

​

​

​

​

**CLASS-2 WC\_Reducer**

**​**

​

package com.cgi;

import java.io.IOException;

import java.util.Iterator;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.MapReduceBase;

import org.apache.hadoop.mapred.OutputCollector;

import org.apache.hadoop.mapred.Reducer;

import org.apache.hadoop.mapred.Reporter;

public class WC\_Reducer extends MapReduceBase implements Reducer<Text,IntWritable,Text,IntWritable> {

public void reduce(Text key, Iterator<IntWritable> values,OutputCollector<Text,IntWritable> output,

Reporter reporter) throws IOException {

int sum=0;

while (values.hasNext()) {

sum+=values.next().get();

}

output.collect(key,new IntWritable(sum));

}

}

​

​

​

**CLASS-3 WC\_Runner**

​

​

package com.cgi;

import java.io.IOException;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.FileInputFormat;

import org.apache.hadoop.mapred.FileOutputFormat;

import org.apache.hadoop.mapred.JobClient;

import org.apache.hadoop.mapred.JobConf;

import org.apache.hadoop.mapred.TextInputFormat;

import org.apache.hadoop.mapred.TextOutputFormat;

public class WC\_Runner {

public static void main(String[] args) throws IOException{

JobConf conf = new JobConf(WC\_Runner.class);

conf.setJobName("CharCount");

conf.setOutputKeyClass(Text.class);

conf.setOutputValueClass(IntWritable.class);

conf.setMapperClass(WC\_Mapper.class);

conf.setCombinerClass(WC\_Reducer.class);

conf.setReducerClass(WC\_Reducer.class);

conf.setInputFormat(TextInputFormat.class);

conf.setOutputFormat(TextOutputFormat.class);

FileInputFormat.setInputPaths(conf,new Path(args[0]));

FileOutputFormat.setOutputPath(conf,new Path(args[1]));

JobClient.runJob(conf);

}

}

​

Once the three classes are made click on

File-> export -> Java-> select **JAR file** and click on next

Select the export destination by clicking on browse and name the file as **CharacterCount**

Then click on next-> next->finish.

Now you can see the **CharacterCount.jar** file at the given destination.

Go to C drive. Open the folder **data** create a text file **character.txt**. Enter a few words in the character.txt file and save it.

Open a command prompt and change directory to C:\Hadoop\hadoop-3.3.0\sbin

Type the command hdfs dfs -copyFromLocal C:/data/character.txt /input

(This will copy the character.txt file from the c drive to hdfs file system)

Change directory to C:\Hadoop\hadoop-3.3.0 (cd..)

Type the command hadoop jar file\_path/CharacterCount.jar WC\_Runner /input/character.txt /output2

In the above command replace file\_path with the actual location of the **CharacterCount.jar** file

After running this command you can see output2 folder in localhost:9870 in the browser. (By navigating through Utilities> Browse the file system)