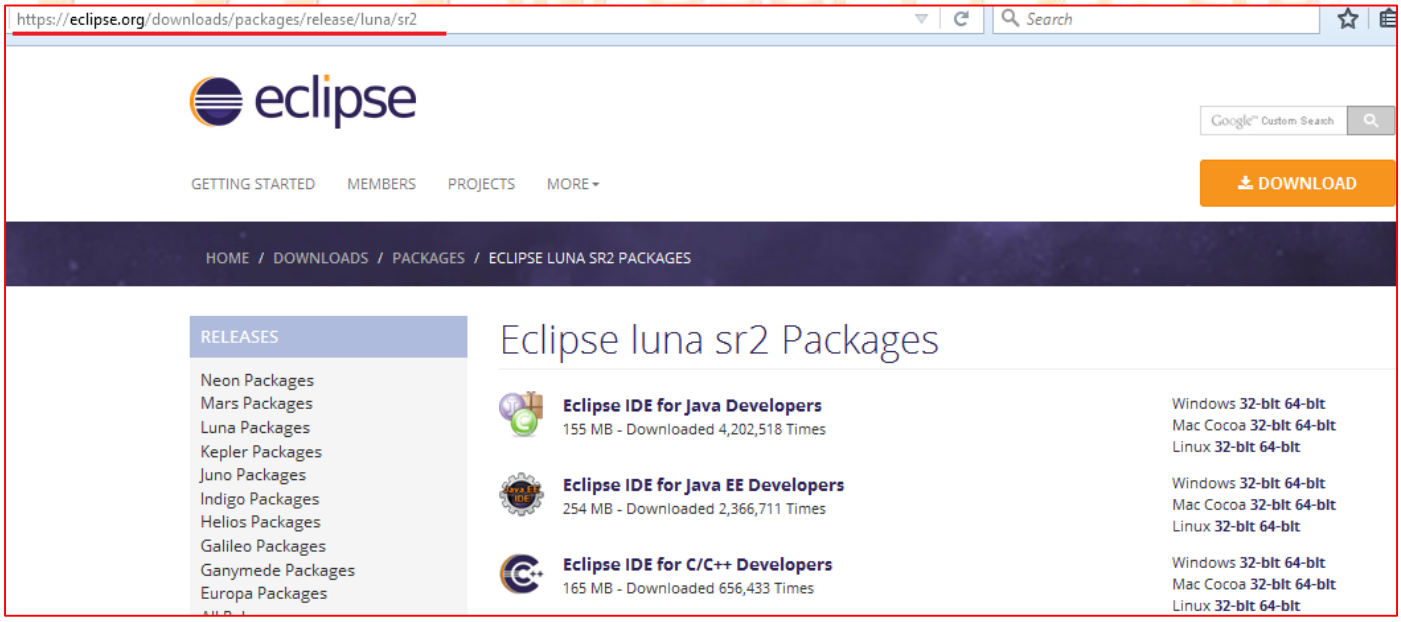
Configuration WordCount Program with Eclipse IDE and Run Program in Hadoop2.x

Steps to configure and Run WordCount Program

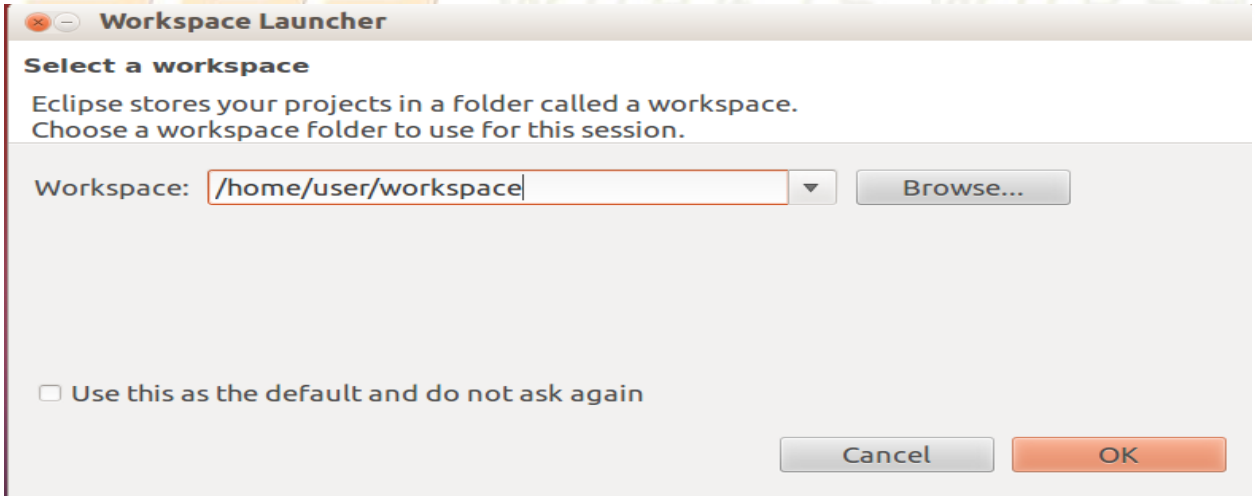
Step: Download Eclipse according to 32 bit or 64 bit.

https://eclipse.org/downloads/packages/release/luna/sr2

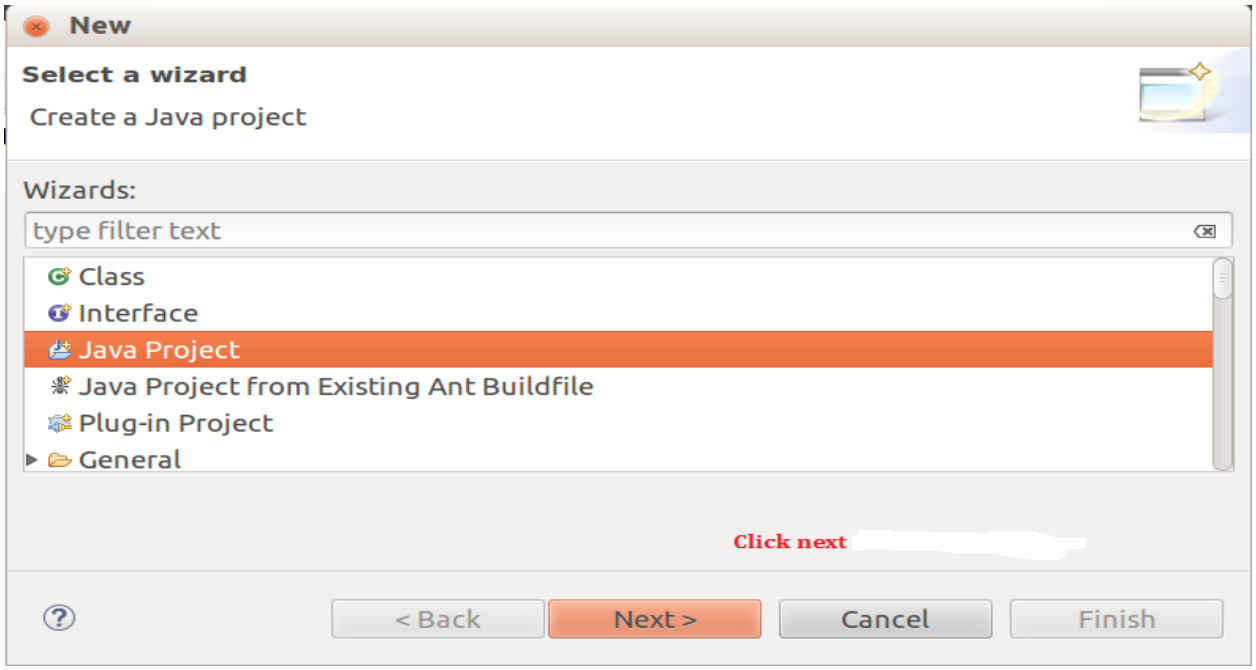


Step 1: Extract Eclipse and Click on eclipse icon

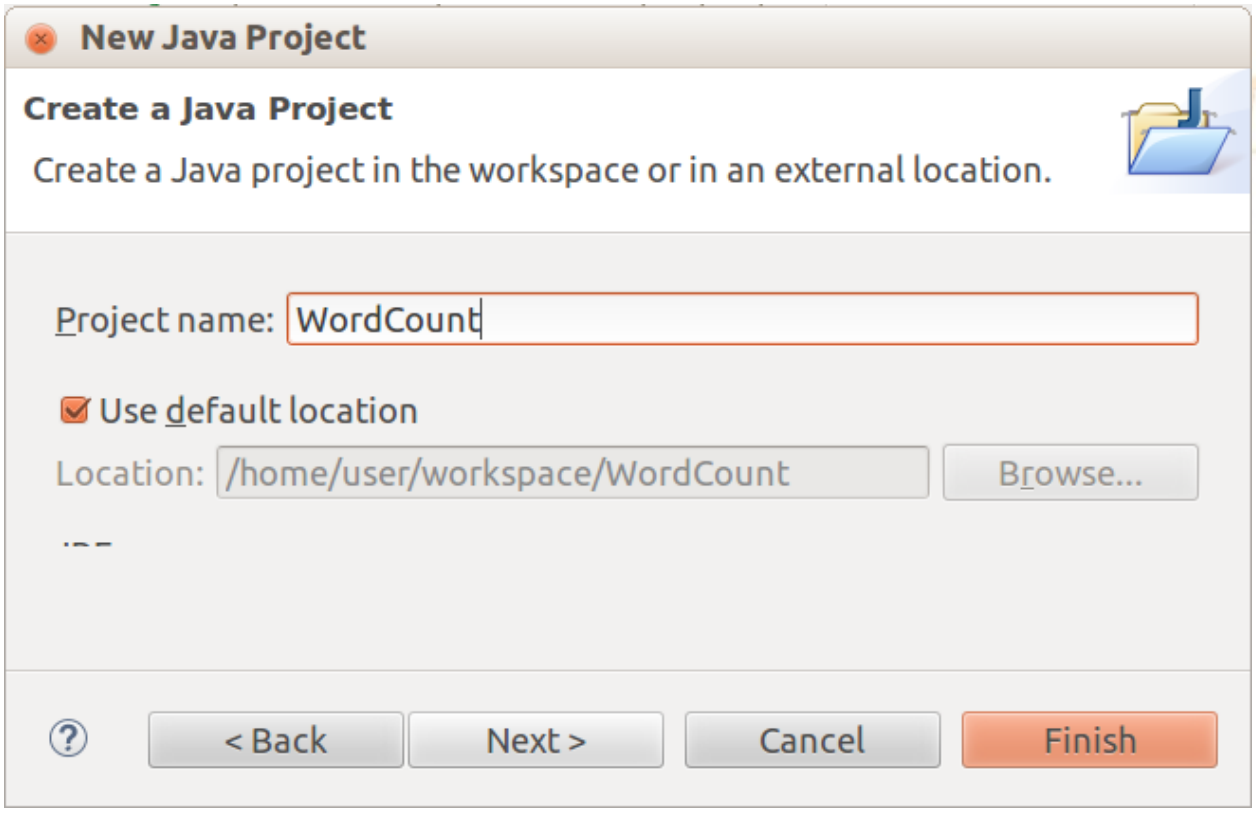
Step 2: Create workspace in /home/use/workspace if want to change then give location or browse.



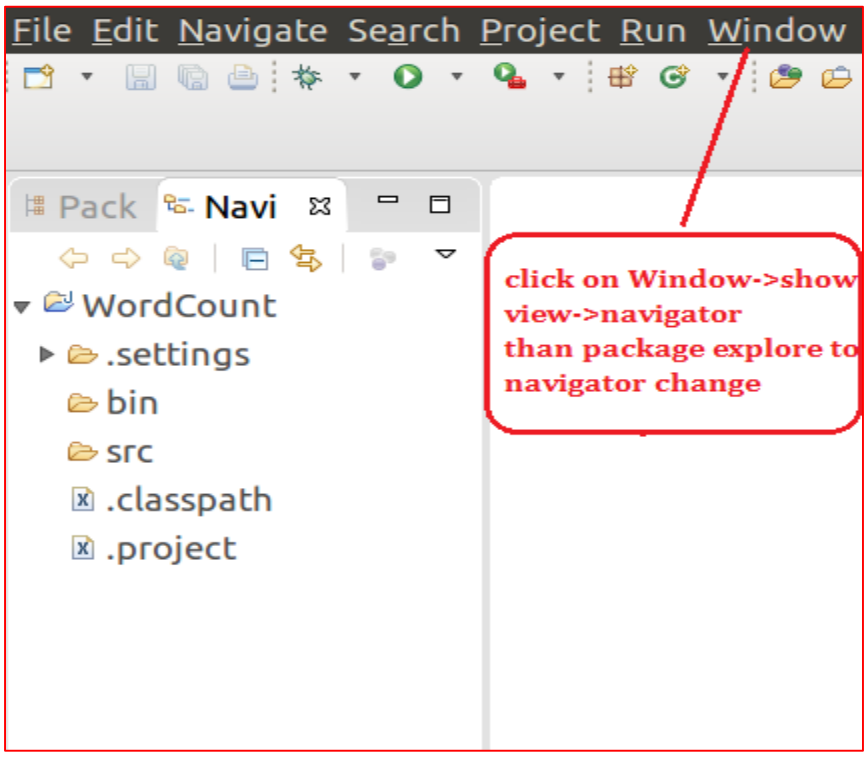
Step 3: Create project file->new->other->java->javaproject



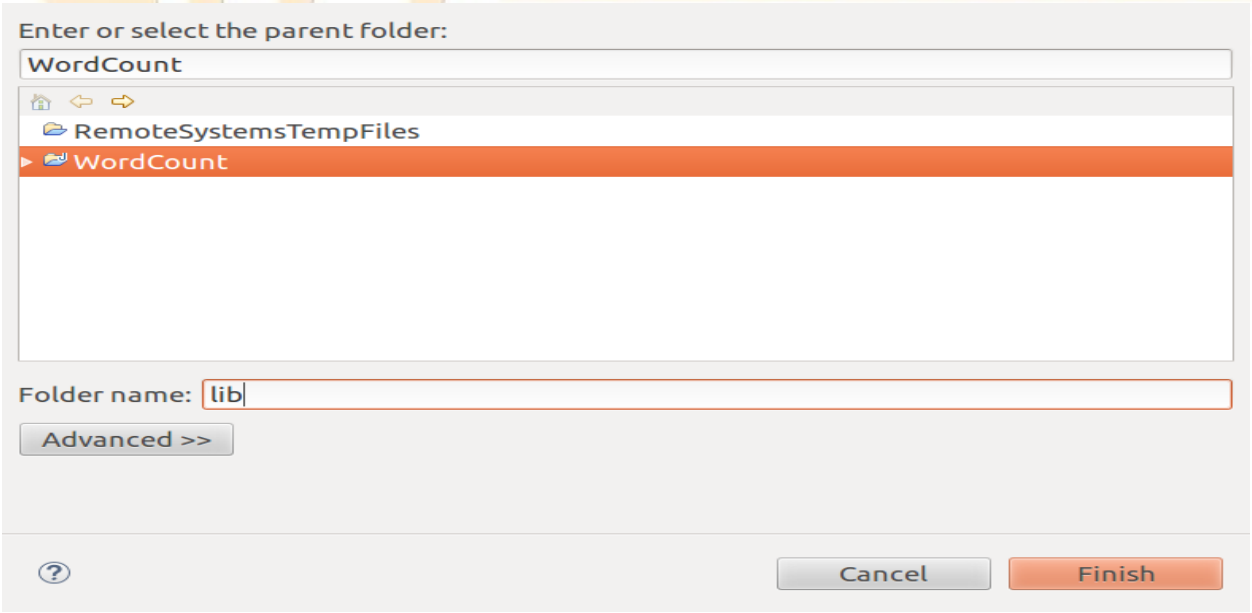
Step 4: Give Project name WordCount



Step 5: change the default view (Project Explore to Navigator) Window->show view->navigator)



Step 6: create lib folder inside WordCount project. Right click on WordCount->New->Folder



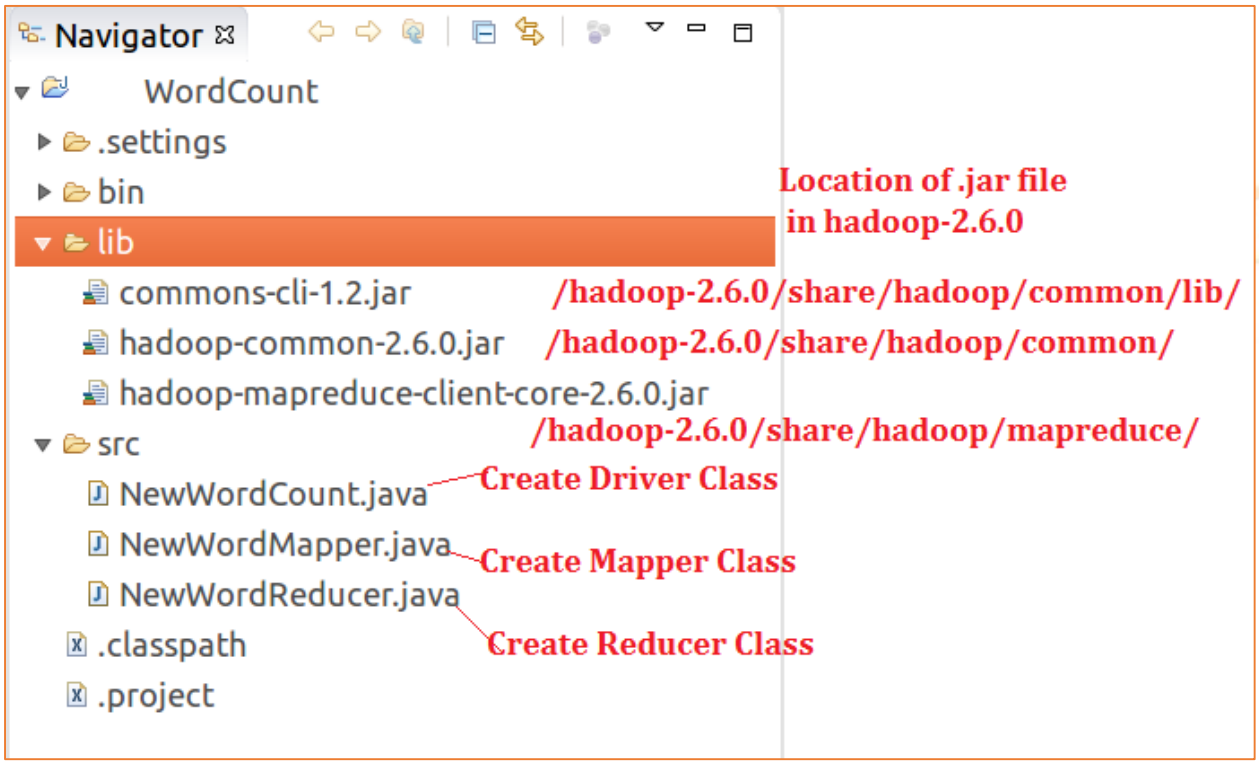
Step 7 : Copy the Three jar file in lib folder and Create Three java class.

Jar file name and location.

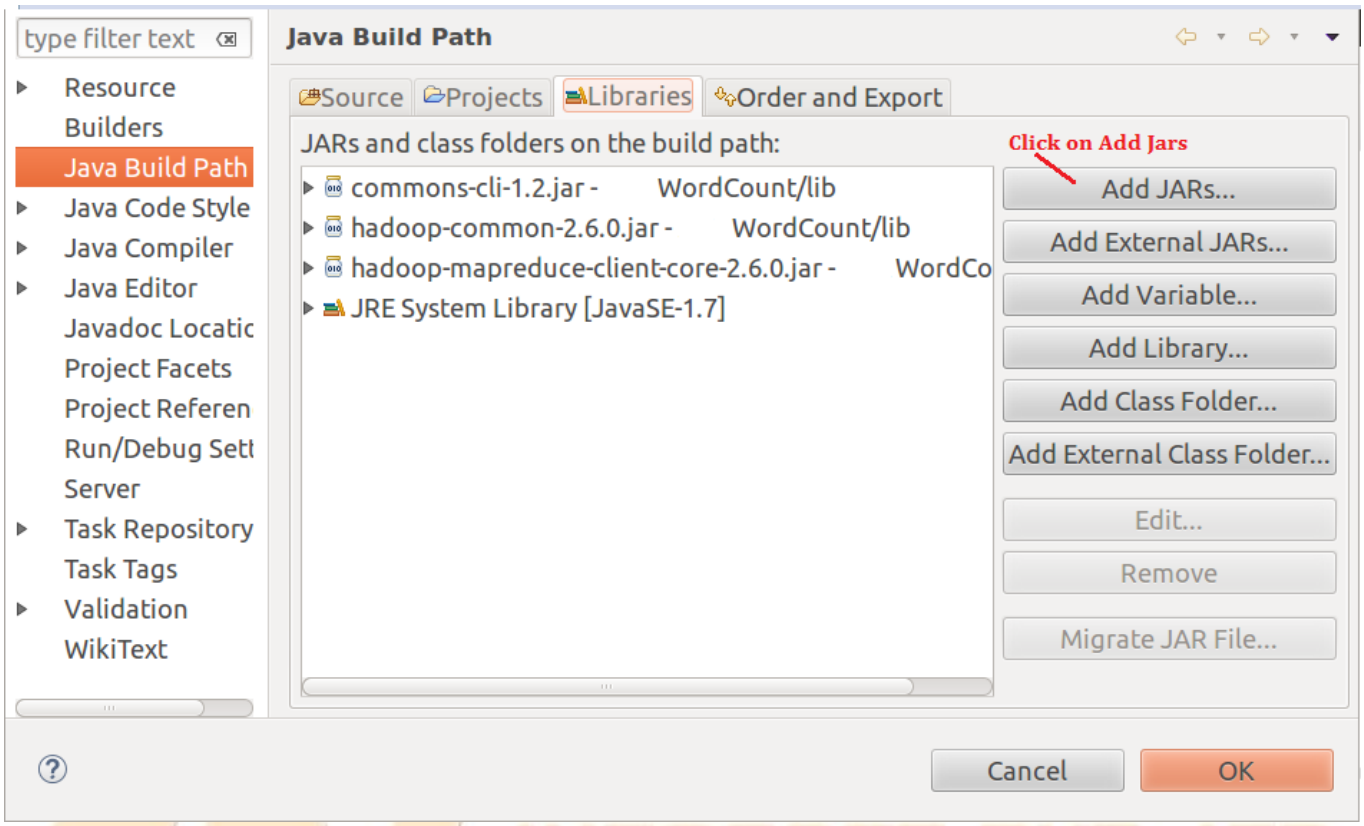
a) /hadoop-2.6.0/share/hadoop/common/lib : commons-cli-1.2.jar
b) /hadoop-2.6.0/share/hadoop/common : hadoop-common-2.6.0.jar
c) /hadoop-2.6.0/share/hadoop/mapreduce/ : hadoop-mapredure-client-core-2.6.0.jar

Three java file for Drivercode,Mapper code,Reducer code.

NewWordCount.java : main class
NewWordMapper.java
NewWordReducer.java



Step 8: set java build path (class path) by Right Click on ->WordCountProject-> Properties->JavaBuildPath->Libaries->Click on Add jar and find three jar file in lib folder of WordCount project.



NewWordCount.java

```
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class NewWordCount
{
        /* ~J$ hadoop jar <jar file name> <Driver Code> <input file name> <output file name>
         * ~J$ hadoop jar WordCount.jar NewWordCount i/p o/p

         */

        public static void main(String[] args) throws Exception
```

```
	{
		//Creating an object of Configuration class, which loads the configuration parameters
		Configuration conf = new Configuration();
		//Creating the object of Job class and passing the conf object  and Job name as arguments. The Job class allows the user  to configure the job, submit it and control its execution.
		Job job = new Job(conf, "wordcount");
		//Setting the jar by finding where a given class came from
		job.setJarByClass(NewWordCount.class);
		//Setting the key class for job output data
		job.setOutputKeyClass(Text.class);
		//Setting the value class for job output data
		job.setOutputValueClass(IntWritable.class);
		//Setting the mapper for the job
		job.setMapperClass(NewWordMapper.class);
		//Setting the reducer for the job
		job.setReducerClass(NewWordReducer.class);
		//Setting the Input Format for the job
		job.setInputFormatClass(TextInputFormat.class);
		//Setting the Output Format for the job
		job.setOutputFormatClass(TextOutputFormat.class);
		//Adding a path which will act as a input for MR job. args[0]  means it will use the first argument written on terminal  as input path
		FileInputFormat.addInputPath(job, new Path(args[0]));
		//Setting the path to a directory where MR job will dump the  output. args[1] means it will use the second argument written on terminal as output path
		FileOutputFormat.setOutputPath(job,new Path(args[1]));
		//Submitting the job to the cluster and waiting for its completion
		job.waitForCompletion(true);


	}


}
```

NewWordMapper.java

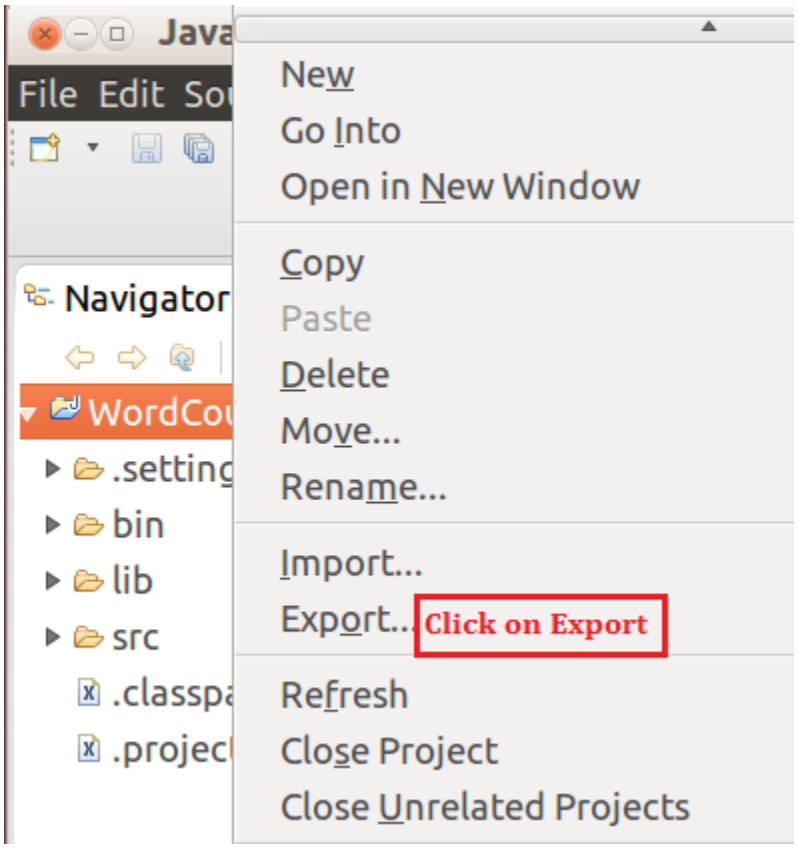
```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
public class NewWordMapper extends Mapper<LongWritable, Text, Text, IntWritable>

{
```

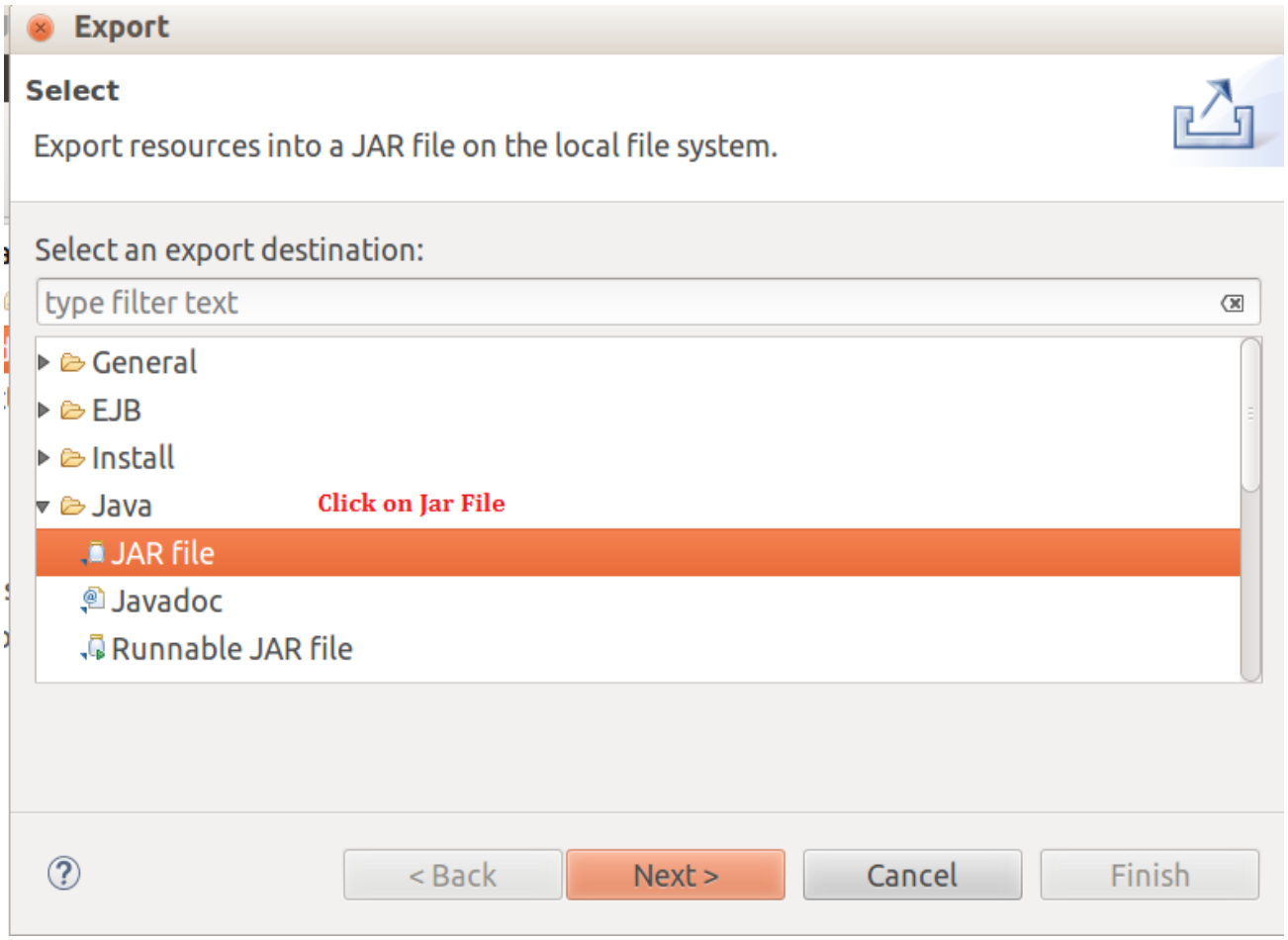
```
        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();
        public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException

        {
                String line = value.toString();
                StringTokenizer tokenizer = new StringTokenizer(line);
                while(tokenizer.hasMoreTokens())
                {
                        word.set(tokenizer.nextToken());
                        context.write(word, one);
                }
        }
}
```

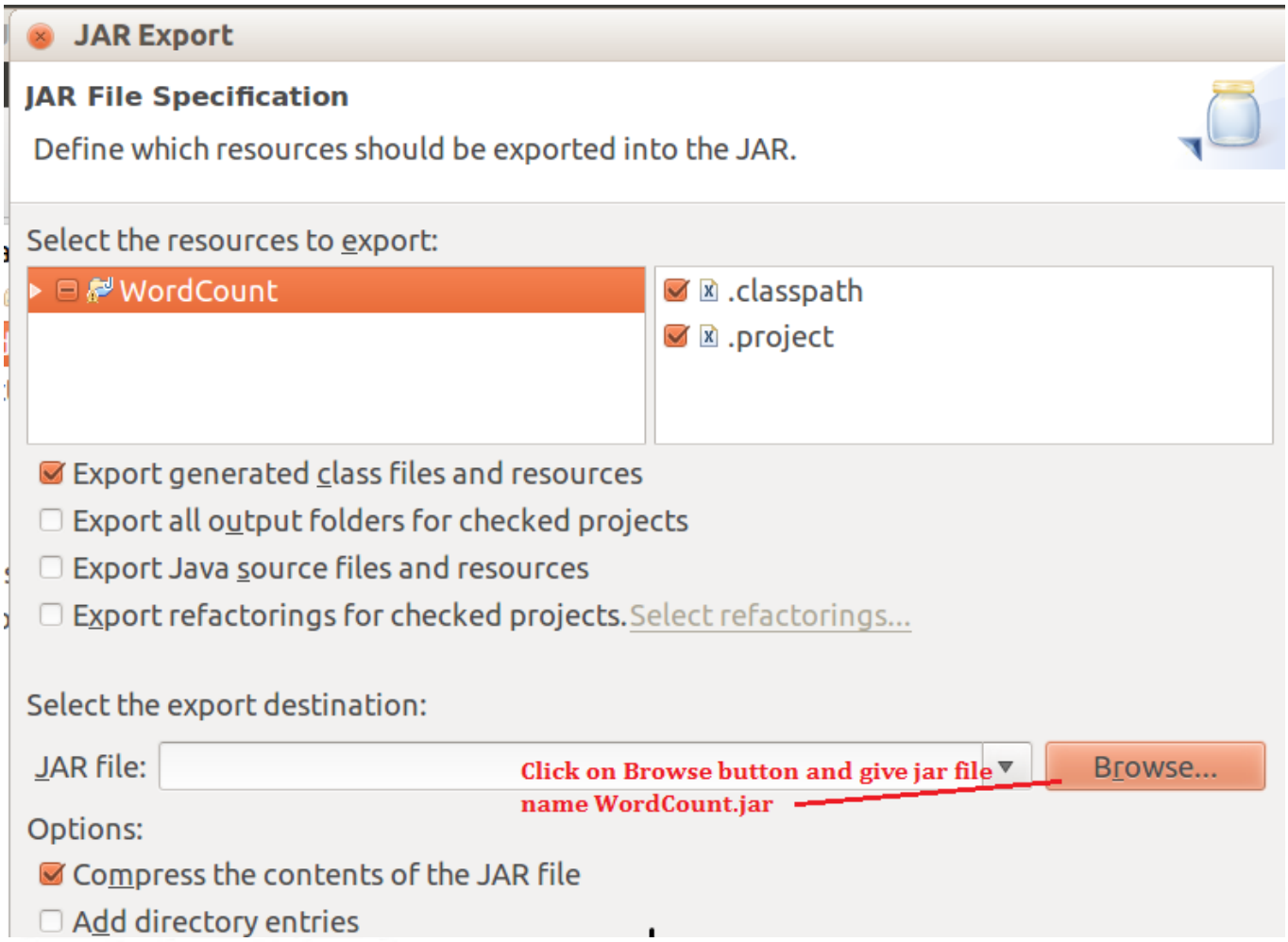
NewWordReducer.java

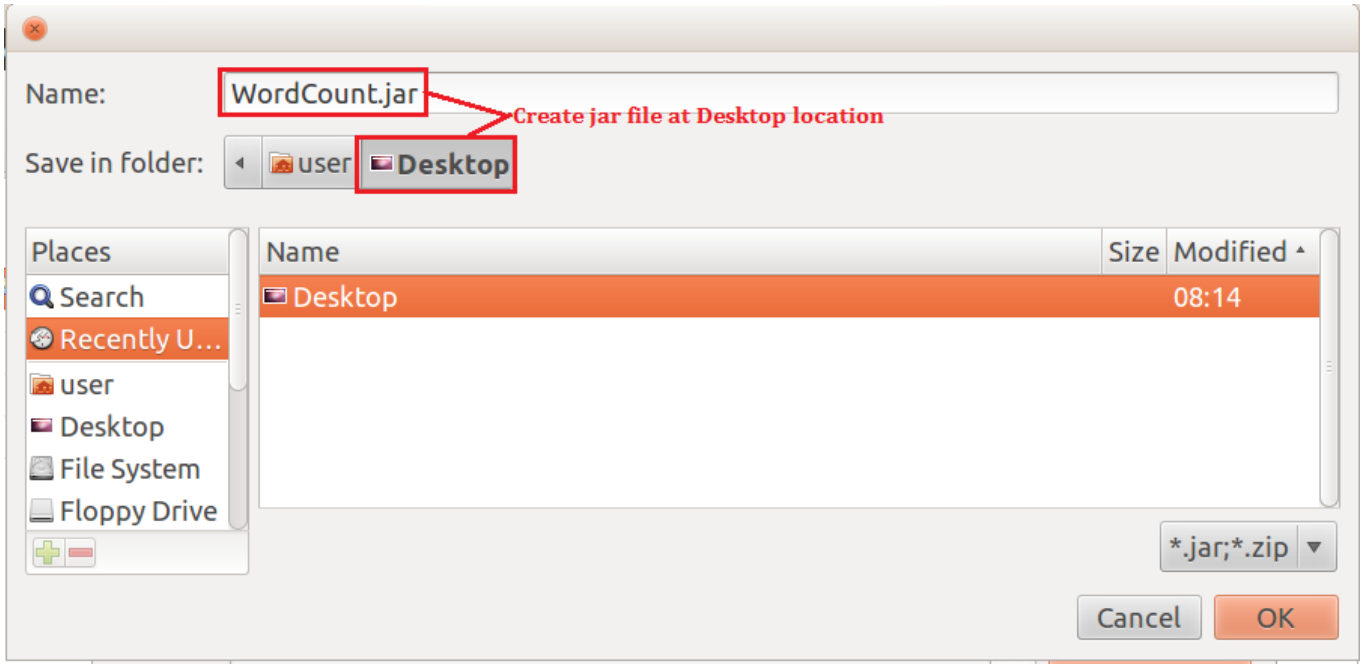
```
import java.io.IOException;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

public class NewWordReducer extends Reducer<Text, IntWritable, Text, IntWritable>

{

        public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException,
InterruptedException

        {       int count = 0;

                for(IntWritable val : values)

                {               count += val.get();

                }

                context.write(key, new IntWritable(count));

        }

}
```

Step 9: Create Jar file Right click on WordCountProject->Export->java->jar file->Browse->give jar WordCount.jar filename ->OK->finish.


File Edit
Navigator
WordCou
.setting
bin
lib
src
.classpa
.projec
New
Go Into
Open in New Window
Copy
Paste
Delete
Move...
Rename...
Import...
Export...
Click on Export
Refresh
Close Project
Close Unrelated Projects


Export
Select
Export resources into a JAR file on the local file system.
Select an export destination:
type filter text
General
EJB
Install
Java
Click on Jar File
JAR file
Javadoc
Runnable JAR file
< Back
Next >
Cancel
Finish


JAR Export
JAR File Specification
Define which resources should be exported into the JAR.
Select the resources to export:
WordCount
.classpath
.project
Export generated class files and resources
Export all output folders for checked projects
Export Java source files and resources
Export refactorings for checked projects. Select refactorings...
Select the export destination:
JAR file:
Browse...
Click on Browse button and give jar file name WordCount.jar
Options:
Compress the contents of the JAR file
Add directory entries


Name:
WordCount.jar
Create jar file at Desktop location
Save in folder:
user
Desktop
Places
Search
Recently U...
user
Desktop
File System
Floppy Drive
Name
Size
Modified
Desktop
08:14
.jar;.zip
Cancel
OK

Step 10: Create txt file name is inputfile

hi how are you

how is your job

how is your family

how is your brother

how is your sister

what is the time now

what is the strength of hadoop

Step 11 : create directory inside hdfs name is /home/user/input

```
user@ubuntu: ~
user@ubuntu:~$ hadoop fs -mkdir -p /home/user/input
OpenJDK Client VM warning: You have loaded library /home/user/hadoop-2.6.0/lib/n
ative/libhadoop.so.1.0.0 which might have disabled stack guard. The VM will try
to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>',
or link it with '-z noexecstack'.
16/06/16 13:06:43 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
user@ubuntu:~$
```

Step 12: move inputfile.txt in hdfs /home/user/input director.

```
user@ubuntu: ~                    input file location        hdfs dir location
user@ubuntu:~$ hadoop fs -put '/home/user/Desktop/inputfile' /home/user/input
OpenJDK Client VM warning: You have loaded library /home/user/hadoop-2.6.0/lib/n
ative/libhadoop.so.1.0.0 which might have disabled stack guard. The VM will try
to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>',
or link it with '-z noexecstack'.
16/06/16 13:11:04 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
```

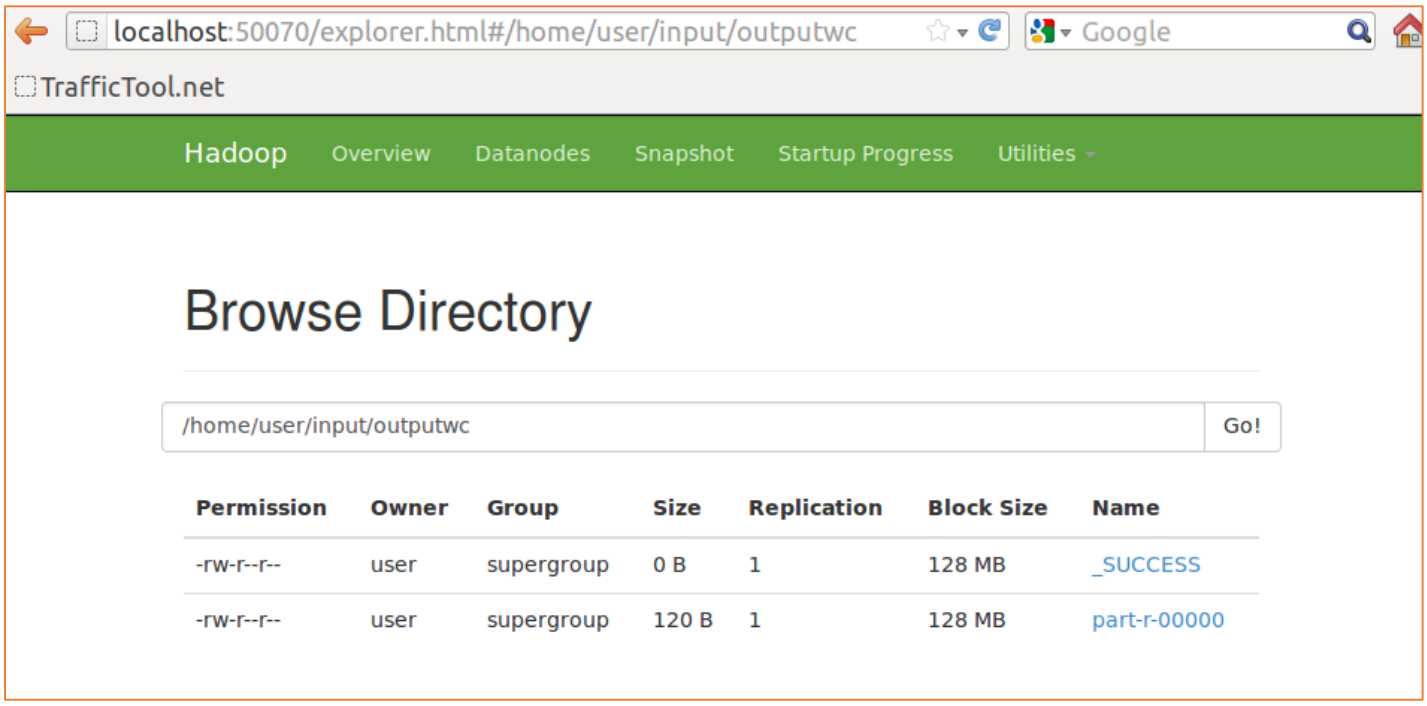
Step 13: Run WordCount.jar file in HDFS .

```
user@ubuntu: ~ WordCount.jar file location at localfile system   Main Class name   inputfile locatio in HDFS
user@ubuntu:~$ hadoop jar '/home/user/Desktop/WordCount.jar' NewWordCount /home/user/i
nput/inputfile /home/user/input/outputwc
                                output directory location where part-r-00000
                                and -SUCCESS file create
```

Step 14: Console final output.

```
user@ubuntu: ~
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=477
                CPU time spent (ms)=10180
                Physical memory (bytes) snapshot=231100416
                Virtual memory (bytes) snapshot=806215680
                Total committed heap usage (bytes)=137433088
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=142
        File Output Format Counters
                Bytes Written=120
```

Step 15: http://localhost:500070 Browser output.



Step 16: out file part-r-00000

```
user@ubuntu:~$ hadoop fs -cat /home/user/input/outputwc/part-r-00000
OpenJDK Client VM warning: You have loaded library /home/user/hadoop-2.6.0/lib/native/l
ibhadoop.so.1.0.0 which might have disabled stack guard. The VM will try to fix the sta
ck guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link
 it with '-z noexecstack'.
16/06/16 13:33:43 WARN util.NativeCodeLoader: Unable to load native-hadoop library for
your platform... using builtin-java classes where applicable
are     1
brother 1
family  1
hadoop  1
hi      1
how     5
is      6
job     1
now     1
of      1
sister  1
strength        1
the     2
time    1
what    2
you     1
your    4
```