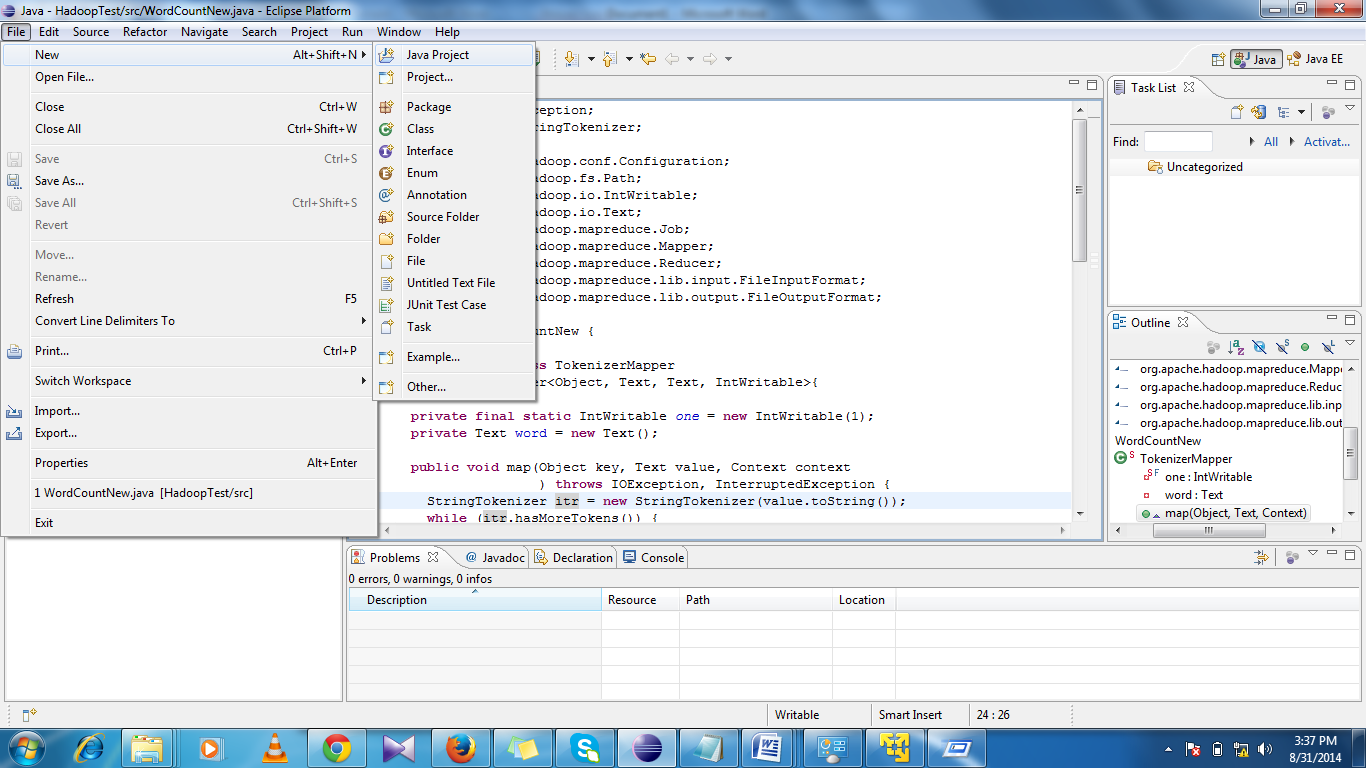
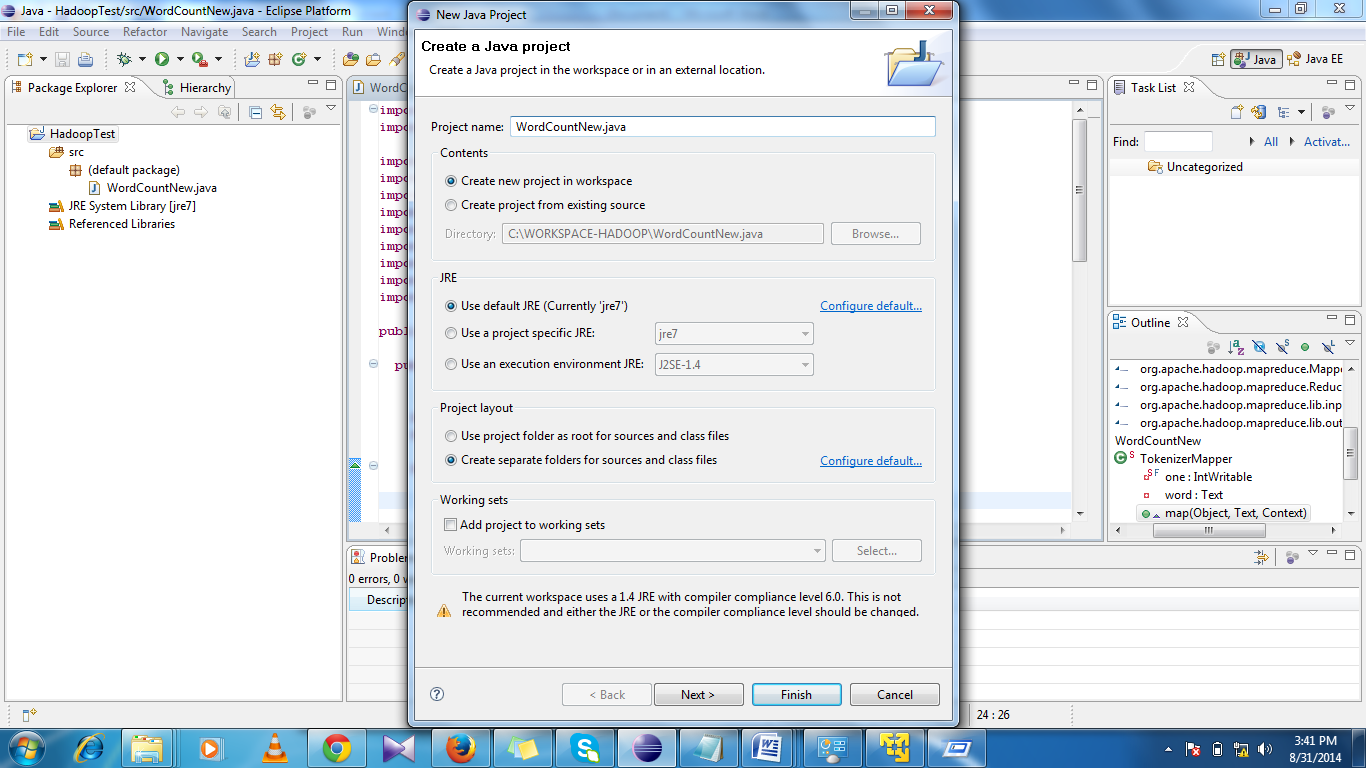
**Creation of jar file and Execution Process in Clustered Environment**

**Process of Creation of Jar file:**

**Step1**: Open Eclipse, click on file, click on New then click on Java Project(if not available click on Other select Java and click on Java project)





**Step2:**

Copy the provided code or write the below code.

**import** java.io.IOException;

**import** java.util.StringTokenizer;

**import** org.apache.hadoop.conf.Configuration;

**import** org.apache.hadoop.fs.Path;

**import** org.apache.hadoop.io.IntWritable;

**import** org.apache.hadoop.io.Text;

**import** org.apache.hadoop.mapreduce.Job;

**import** org.apache.hadoop.mapreduce.Mapper;

**import** org.apache.hadoop.mapreduce.Reducer;

**import** org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

**import** org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

**public** **class** WordCountNew {

**public** **static** **class** TokenizerMapper

**extends** Mapper<Object, Text, Text, IntWritable>{

**private** **final** **static** IntWritable *one* = **new** IntWritable(1);

**private** Text word = **new** Text();

**public** **void** map(Object key, Text value, Context context

) **throws** IOException, InterruptedException {

StringTokenizer itr = **new** StringTokenizer(value.toString());

**while** (itr.hasMoreTokens()) {

word.set(itr.nextToken());

context.write(word, *one*);

}

}

}

**public** **static** **class** IntSumReducer

**extends** Reducer<Text,IntWritable,Text,IntWritable> {

**private** IntWritable result = **new** IntWritable();

**public** **void** reduce(Text key, Iterable<IntWritable> values,

Context context

) **throws** IOException, InterruptedException {

**int** sum = 0;

**for** (IntWritable val : values) {

sum += val.get();

}

result.set(sum);

context.write(key, result);

}

}

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

Configuration conf = **new** Configuration();

Job job = **new** Job(conf, "word count");

job.setJarByClass(WordCountNew.**class**);

job.setMapperClass(TokenizerMapper.**class**);

job.setCombinerClass(IntSumReducer.**class**);

job.setReducerClass(IntSumReducer.**class**);

job.setOutputKeyClass(Text.**class**);

job.setOutputValueClass(IntWritable.**class**);

FileInputFormat.*addInputPath*(job, **new** Path(args[0]));

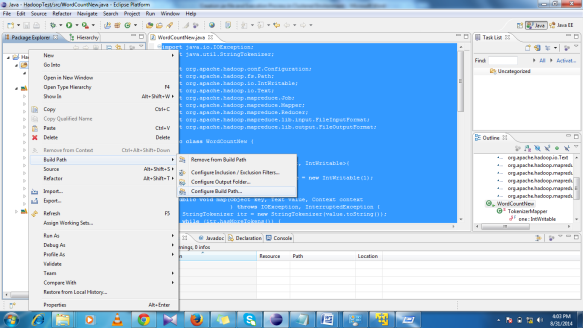
FileOutputFormat.*setOutputPath*(job, **new** Path(args[1]));

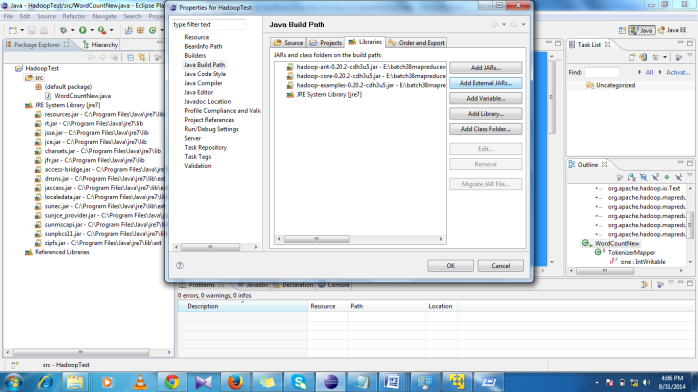
System.*exit*(job.waitForCompletion(**true**) ? 0 : 1);

}

}

**Step3**: Right Click on “src” which is in Package Explorer tab -> Click On “Build Path”->Configure Build Path->Click on Libraries tab->Add External JARs..->Press “Ok” Button.





**Step4**: Step-3: Right Click on “src” ->New->Class->give the name of Class File same as in code->Press “Finish” Button->Paste Code/Write the Code

**Step5**: Again Right Click on “src”->Export->Select JAR File under the Java->give the path as well as Name of JAR File->Press Finish

Button

**Execution Process:**

**Step1:** copy the jar file and input text file(provided) into download folder(provided server shared path).

**Step2**: copy the text file into local folder and copy the jar file into local folder.

**Step3**: Create the new directory in HDFS path (hadoop fs –mkdir MrInput)

**Step4**: copy the text file from LFS to HDFS by using –put or –copyFromLocal.

**Step5**:

root@ubuntu**:/home/chandu# hadoop jar BATCH38-WORDCOUNT.jar WordCountNew /root/user/chandu/MrInput/Input-Big.txt /root/user/chandu/MROutput**

14/08/31 02:50:00 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the same.

14/08/31 02:50:00 INFO input.FileInputFormat: Total input paths to process : 1

14/08/31 02:50:00 WARN snappy.LoadSnappy: Snappy native library is available

14/08/31 02:50:00 INFO util.NativeCodeLoader: Loaded the native-hadoop library

14/08/31 02:50:00 INFO snappy.LoadSnappy: Snappy native library loaded

14/08/31 02:50:00 INFO mapred.JobClient: Running job: job\_201408310158\_0002

14/08/31 02:50:01 INFO mapred.JobClient: map 0% reduce 0%

14/08/31 02:50:07 INFO mapred.JobClient: map 100% reduce 0%

14/08/31 02:50:14 INFO mapred.JobClient: map 100% reduce 33%

14/08/31 02:50:16 INFO mapred.JobClient: map 100% reduce 100%

14/08/31 02:50:16 INFO mapred.JobClient: Job complete: job\_201408310158\_0002

14/08/31 02:50:16 INFO mapred.JobClient: Counters: 26

14/08/31 02:50:16 INFO mapred.JobClient: Job Counters

14/08/31 02:50:16 INFO mapred.JobClient: Launched reduce tasks=1

14/08/31 02:50:16 INFO mapred.JobClient: SLOTS\_MILLIS\_MAPS=5140

14/08/31 02:50:16 INFO mapred.JobClient: Total time spent by all reduces waiting after reserving slots (ms)=0

14/08/31 02:50:16 INFO mapred.JobClient: Total time spent by all maps waiting after reserving slots (ms)=0

14/08/31 02:50:16 INFO mapred.JobClient: Launched map tasks=1

14/08/31 02:50:16 INFO mapred.JobClient: Data-local map tasks=1

14/08/31 02:50:16 INFO mapred.JobClient: SLOTS\_MILLIS\_REDUCES=8545

14/08/31 02:50:16 INFO mapred.JobClient: FileSystemCounters

14/08/31 02:50:16 INFO mapred.JobClient: FILE\_BYTES\_READ=139

14/08/31 02:50:16 INFO mapred.JobClient: HDFS\_BYTES\_READ=153395

14/08/31 02:50:16 INFO mapred.JobClient: FILE\_BYTES\_WRITTEN=119182

14/08/31 02:50:16 INFO mapred.JobClient: HDFS\_BYTES\_WRITTEN=122

14/08/31 02:50:16 INFO mapred.JobClient: Map-Reduce Framework

14/08/31 02:50:16 INFO mapred.JobClient: Map input records=5487

14/08/31 02:50:16 INFO mapred.JobClient: Reduce shuffle bytes=139

14/08/31 02:50:16 INFO mapred.JobClient: Spilled Records=22

14/08/31 02:50:16 INFO mapred.JobClient: Map output bytes=251174

14/08/31 02:50:16 INFO mapred.JobClient: CPU time spent (ms)=2000

14/08/31 02:50:16 INFO mapred.JobClient: Total committed heap usage (bytes)=177016832

14/08/31 02:50:16 INFO mapred.JobClient: Combine input records=25872

14/08/31 02:50:16 INFO mapred.JobClient: SPLIT\_RAW\_BYTES=125

14/08/31 02:50:16 INFO mapred.JobClient: Reduce input records=11

14/08/31 02:50:16 INFO mapred.JobClient: Reduce input groups=11

14/08/31 02:50:16 INFO mapred.JobClient: Combine output records=11

14/08/31 02:50:16 INFO mapred.JobClient: Physical memory (bytes) snapshot=186167296

14/08/31 02:50:16 INFO mapred.JobClient: Reduce output records=11

14/08/31 02:50:16 INFO mapred.JobClient: Virtual memory (bytes) snapshot=749965312

14/08/31 02:50:16 INFO mapred.JobClient: Map output records=25872

Step6:

root@ubuntu:**/home/chandu# hadoop fs -ls /root/user/chandu/MrInput**

Found 1 items

-rw-r--r-- 1 root supergroup 153270 2014-08-31 02:33 /root/user/chandu/MrInput/Input-Big.txt

root@ubuntu:**/home/chandu# hadoop fs -ls /root/user/chandu/MROutput**

Found 3 items

-rw-r--r-- 1 root supergroup 0 2014-08-31 02:50 /root/user/chandu/MROutput/\_SUCCESS

drwxrwxrwx - root supergroup 0 2014-08-31 02:50 /root/user/chandu/MROutput/\_logs

-rw-r--r-- 1 root supergroup 122 2014-08-31 02:50 /root/user/chandu/MROutput/part-r-00000

root@ubuntu:**/home/chandu# hadoop fs -cat /root/user/chandu/MROutput/part-r-00000**

good 4312

hadoop 4312

having 2156

is 4312

knowledge 1078

leader 1078

learn 1078

market 3234

now 2156

people 1078

the 1078

root@ubuntu:/home/chandu#