

## Steps to execute MapReduce programs on Hadoop:

1. Create a directory and write Driver.java, MapperCode.java, ReducerCode.java (names may not be the same for the files), make sure the Driver.java has the correct class names of Mapper and Reducer Class. Make sure all java files have a package in them Eg:

```
package Test;
```

2. Create Manifest.txt file and write

```
Main-Class:[package-name].[DriverClassName]
```

(Driver class is the class with the main function)

```
Main-Class: (blankspace)Test.Driver
```

(here Test is the package name and Driver is the class name driver)

3. Now lets configure the classpath

Open the folder where Hadoop folder is there, right-click and open properties

Copy the path, eg: /home/rit/

In the directory where files are stored, open terminal on that path run the command

From now all the commands should be executed in the terminal in the same directory where the java files there.

```
export HADOOP_HOME="[path hadoop]/hadoop2/"
```

4. Find the correct names of the jars

a. hadoop-mapreduce-client-core-[version].jar

b. hadoop-mapreduce-client-common-[version].jar

c. hadoop-common-[version].jar

and replace them with the correct version names in share/hadoop/mapreduce directory

```
export CLASSPATH="$HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-client-core-[version].jar:$HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-client-common-[version].jar:$HADOOP_HOME/share/hadoop/common/hadoop-common-[version].jar:$HADOOP_HOME/lib/*"
```

5. Compile the Driver, MapperCode and ReducerCode files by running the command

```
javac -d . Driver.java MapperCode.java ReducerCode.java
```

Will the create .class files in the directory same as the name of the package

6. Now create a jar file:

```
jar cfm [package_name].jar Manifest.txt Test/*.class
```

Jar file is finally created now run the same jar in hadoop system

Running mapreduce:

```
$HADOOP_HOME/bin/hadoop jar [jarfile].jar [input_file/input_folder] output
```

Note: Make sure that there is no folder named output before

If it exists then delete it using the command: **rm -r output**

Commands:

```
export HADOOP_HOME="/home/rit/hadoop2/"
export CLASSPATH="$HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-
client-core-2.7.3.jar:$HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-client-
common-2.7.3.jar:$HADOOP_HOME/share/hadoop/common/hadoop-common-2.7.3.jar:~/
MapReduceTutorial/SalesCountry/*:$HADOOP_HOME/lib/*"
```

```
javac -d . Driver.java MapperCode.java ReducerCode.java
```

Create manifest.txt

Main-Class: Test.Driver

Packagename.Classname

```
jar cfm Test.jar Manifest.txt Test/*.class
```

Steps Temp programs

1. export HADOOP\_HOME="/home/msrit/hadoop2"
2. export CLASSPATH="\$HADOOP\_HOME/share/hadoop/mapreduce/hadoop-mapreduce-client-core-3.2.2.jar:\$HADOOP\_HOME/share/hadoop/mapreduce/hadoop-mapreduce-client-common-3.2.2.jar:\$HADOOP\_HOME/share/hadoop/common/hadoop-common-3.2.2.jar:\$HADOOP\_HOME/lib/\*"
3. javac -d . driver.java reducer.java mapper.java
4. create Manifest.txt file  
command:  
**gedit Manifest.txt**  
add: **Main-Class: temp.driver**
5. jar cfm lab1.jar Manifest.txt temp/\*.class
6. \$HADOOP\_HOME/bin/hadoop jar lab1.jar tem.txt output

```
at org.apache.hadoop.util.RunJar.main(RunJar.java:236)
(base) msrit@msrit-ThinkCentre-M720t:~/Documents/LAB1$ javac -d . driver.java reducer.java mapper.java
(base) msrit@msrit-ThinkCentre-M720t:~/Documents/LAB1$ jar cfm lab1.jar Manifest.txt temp/*.class
(base) msrit@msrit-ThinkCentre-M720t:~/Documents/LAB1$ $HADOOP_HOME/bin/hadoop jar lab1.jar tem.txt
output
```

Tue 10:27

Documents

LAB1



driver.java



Manifest.  
txt



mapper.  
java



reducer.  
java



tem.txt