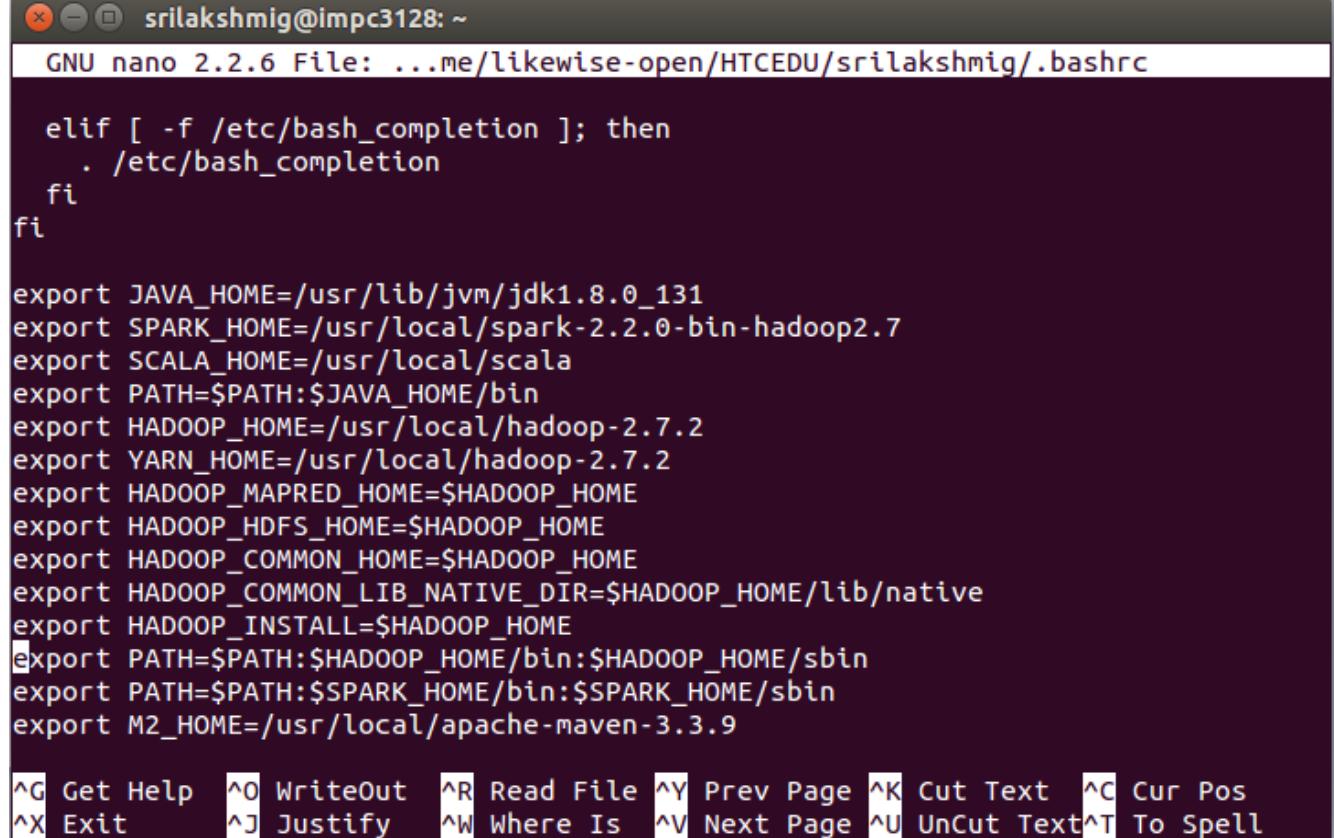


Steps to use Eclipse for Spark Programming

Step 1: Set Spark and Scala HOME in bashrc



```
srilakshmig@impc3128: ~  
GNU nano 2.2.6 File: ...me/likewise-open/HTCEDU/srilakshmig/.bashrc  
  
elif [ -f /etc/bash_completion ]; then  
  . /etc/bash_completion  
fi  
fi  
  
export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_131  
export SPARK_HOME=/usr/local/spark-2.2.0-bin-hadoop2.7  
export SCALA_HOME=/usr/local/scala  
export PATH=$PATH:$JAVA_HOME/bin  
export HADOOP_HOME=/usr/local/hadoop-2.7.2  
export YARN_HOME=/usr/local/hadoop-2.7.2  
export HADOOP_MAPRED_HOME=$HADOOP_HOME  
export HADOOP_HDFS_HOME=$HADOOP_HOME  
export HADOOP_COMMON_HOME=$HADOOP_HOME  
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native  
export HADOOP_INSTALL=$HADOOP_HOME  
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin  
export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin  
export M2_HOME=/usr/local/apache-maven-3.3.9  
  
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos  
^X Exit      ^J Justify   ^W Where Is  ^V Next Page  ^U UnCut Text ^T To Spell
```

Step 2: Download and install Scala Eclipse IDE from the below given link:

<http://scala-ide.org/download/sdk.html>

Note: You need Java 8 as a pre – requisite

(or)

Install the Scala IDE plugin from help → market place in your existing eclipse.

Step 3: Extract it to a location .

Step 4: Edit the eclipse.ini file that is found in the eclipse folder.

Include -vm /usr/lib/jvm/jdk1.8.0_131/jre/bin before -vmargs and ofcourse after startup

```
srilakshmi@impc3128: ~/eclipse
GNU nano 2.2.6 File: eclipse.ini

-startup
plugins/org.eclipse.equinox.launcher_1.3.201.v20161025-1711.jar
--launcher.library
plugins/org.eclipse.equinox.launcher.gtk.linux.x86_64_1.1.401.v20161122-1740
-vm
/usr/lib/jvm/jdk1.8.0_131/jre/bin
-vmargs
-Xmx2G
-Xms200m
-XX:MaxPermSize=384m

[ Read 10 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

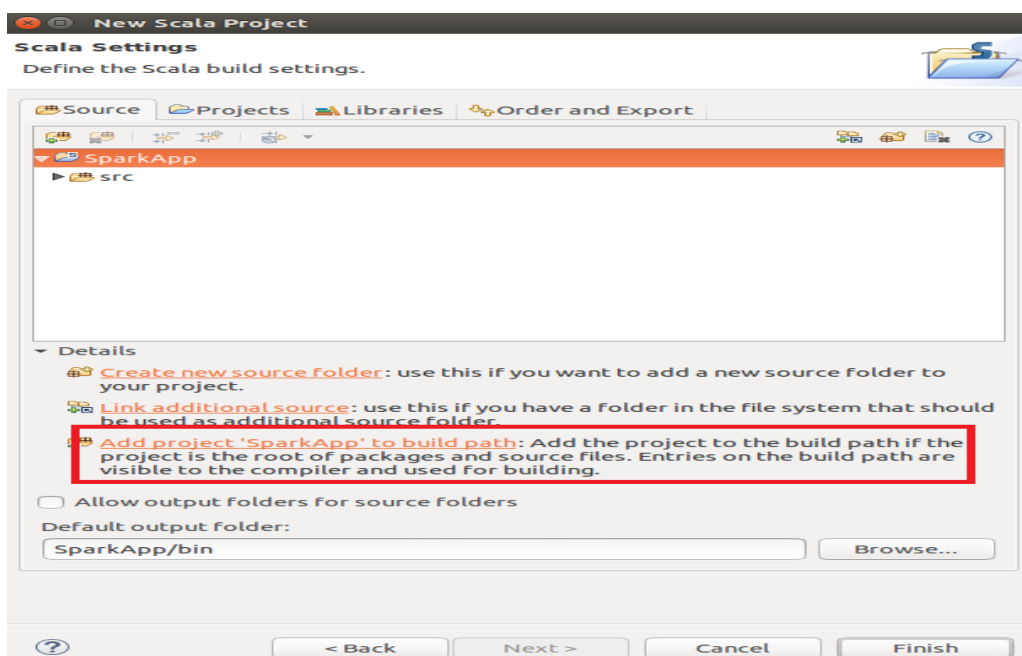
Step 5: Start the eclipse application file

Step 6: Create a new workspace.

Step 7: Create a new Scala project,
File → New → Scala Project

Step 8: Mention the Project Name and click 'Next' Button.

Step 9: Click on “Add Project 'SparkApp' to build path.



Step 10: Click on Finish

Step 11: Expand your project on the package explorer. Select 'src' folder.

Step 12: Create a new package, "com.htc.FindNullValues" under 'src' folder.

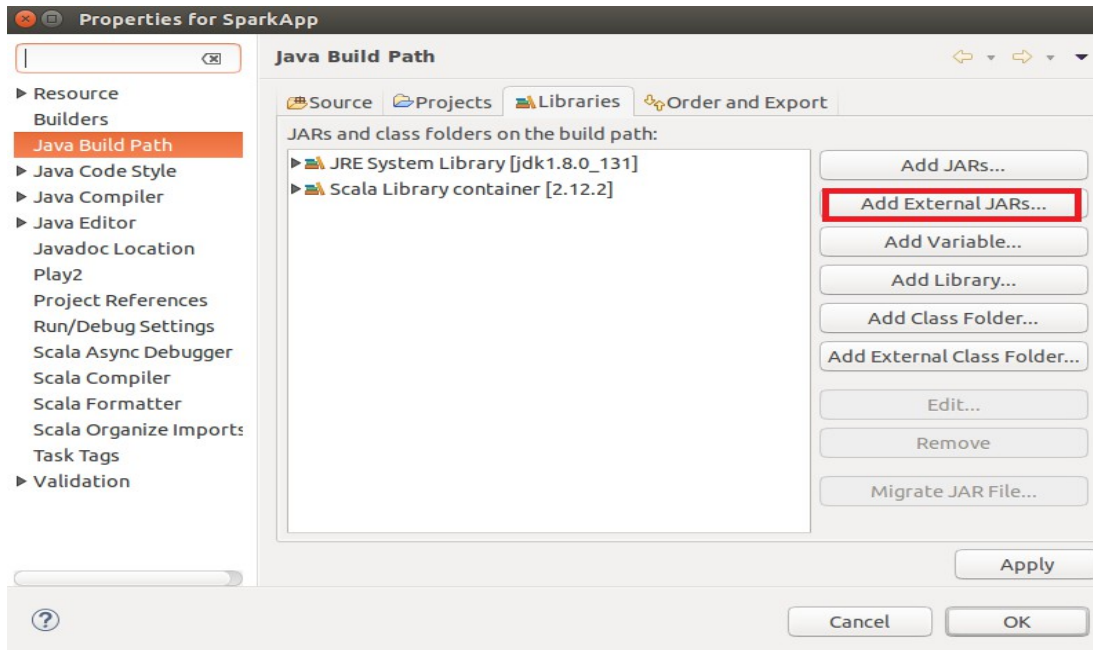
Step 13: Create a new Scala Object inside a package.

Step 14: Import Spark core jar files into your project.

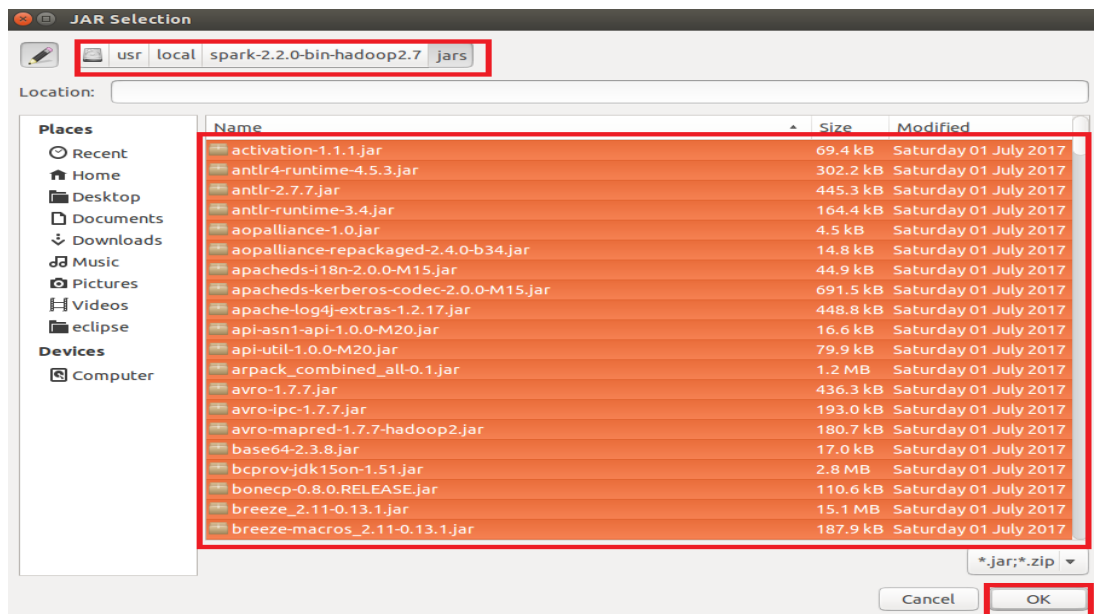
Right click on your project in the package explorer.

Select Build Path → Configure Build Path

Step 15: Select "Add External Jars" from the window that appears:

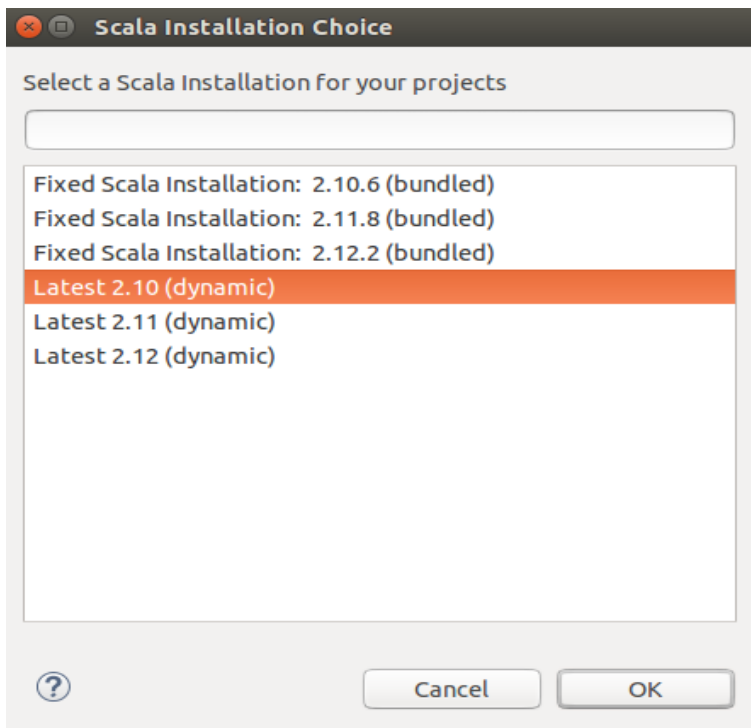


Step 16: Browse for the location where Spark jars are stored. Select All the jars and click on 'Ok' Button.



Step 17: Right click on the project and select Scala → Set Scala Installation

Step 18: Select Latest 2.10 (dynamic) from the list available.



Step 19: Write the following code:

```
package com.htc.FindNullValues
```

```
import org.apache.spark.SparkConf
```

```
import org.apache.spark.SparkContext
```

```
object NullValue {
```

```
  def main(args: Array[String]) {
```

```
    val sc = new SparkContext(new SparkConf().setAppName("Spark Count"))
```

```
    val files = sc.textFile(args(0)).map(_._split(","))
```

```
    def f(x:Array[String]) = {
```

```
      if (x.length > 0)
```

```
        x(3)
```

```
      else
```

```
        "NO NAME"
```

```
    }
```

```
    val names = files.map(f)
```

```
    val wordCounts = names.map(_._1).reduceByKey(_ + _).sortByKey()
```

```
    wordCounts.saveAsTextFile(args(1))
```

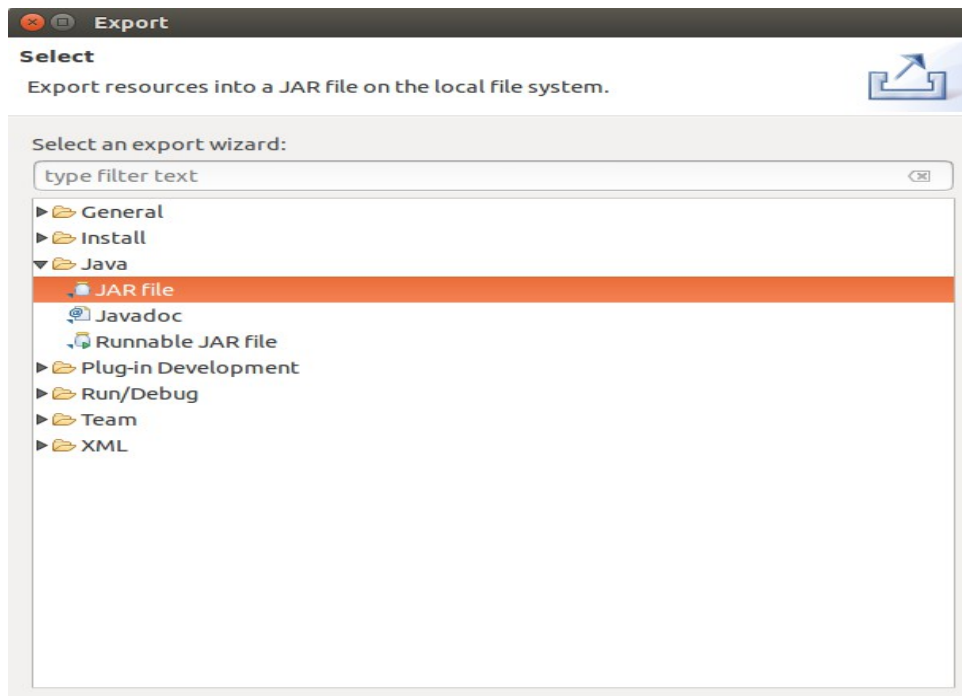
```
  }
```

```
}
```

Step 20:

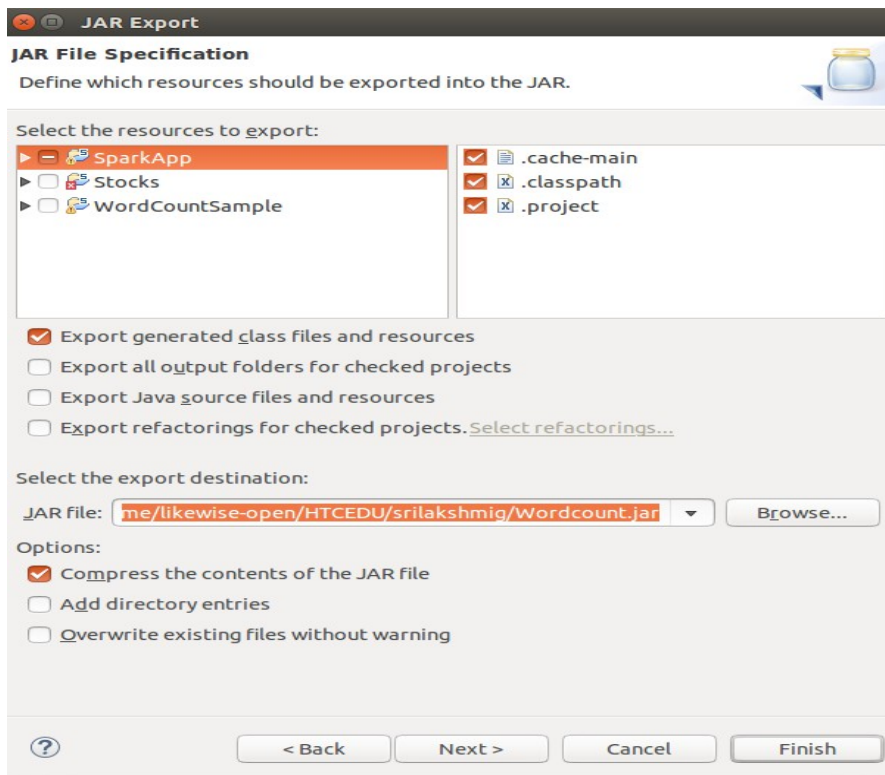
Right click on the project in the package explorer and select export.

Step 21: Select jar file.

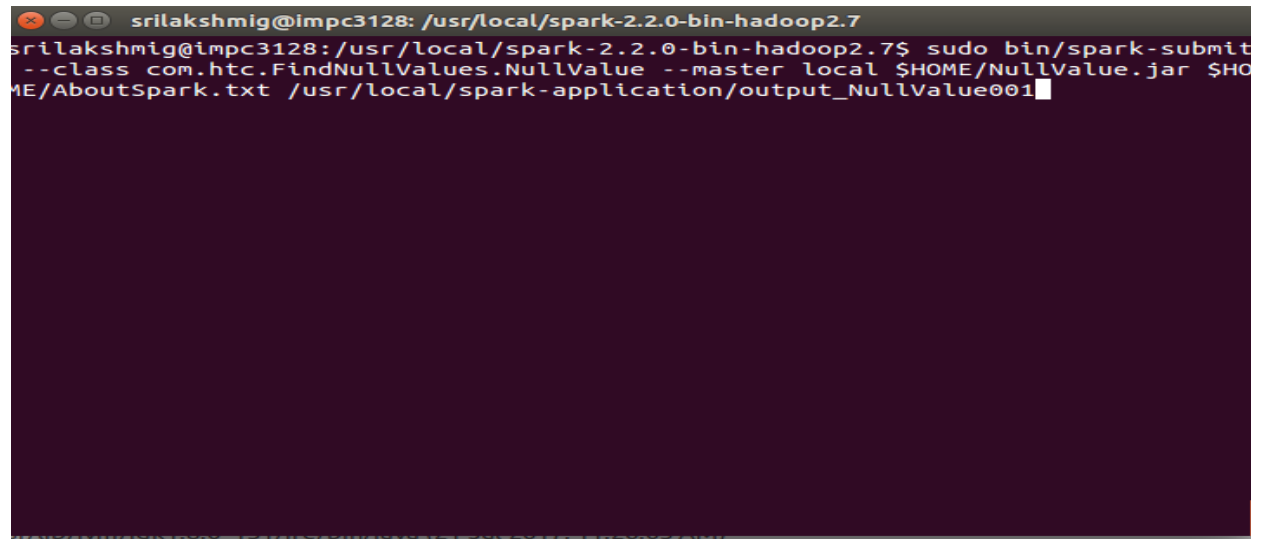


Step 22: Select 'Next' Button.

Step 23: Select the export destination and click 'Finish' button



Step 24: Execute with the following command to submit the job for processing.

A terminal window with a dark background and light-colored text. The window title is 'srilakshmig@impc3128: /usr/local/spark-2.2.0-bin-hadoop2.7'. The command entered is 'sudo bin/spark-submit --class com.htc.FindNullValues.NullValue --master local \$HOME/NullValue.jar \$HOME/AboutSpark.txt /usr/local/spark-application/output_NullValue001'. The cursor is at the end of the command line.

```
srilakshmig@impc3128: /usr/local/spark-2.2.0-bin-hadoop2.7
srilakshmig@impc3128: /usr/local/spark-2.2.0-bin-hadoop2.7$ sudo bin/spark-submit
--class com.htc.FindNullValues.NullValue --master local $HOME/NullValue.jar $HO
ME/AboutSpark.txt /usr/local/spark-application/output_NullValue001
```