Building Spark App using Eclipse - Scala IDE integration & SBT

By
Nagamalleswara Rao Pilli
534705

- Spark Overview
- Eclipse – Scala IDE integration
- Creating Spark Project
- Creating SBT
- Submitting Spark Application
- Conclusion

Spark Overview

Apache Spark is an open source processing engine built around speed, ease of use, and analytics. If you want to process large amounts of data in faster way, existing Map Reduce paradigm/program cannot provide, Spark is the alternative. Spark run programs up to 100 times faster than Hadoop MapReduce in memory, or 10 times faster on disk.

Spark supports Java, Scala, Python, R APIs for ease of development. Spark combines SQL, streaming and complex analytics together seamlessly in the same application to handle a wide range of data processing scenarios. Spark runs on top of Hadoop, Mesos, Standalone, or in the cloud. It can access diverse data sources such as HDFS, Cassandra, HBase, or S3.

Note : Spark is a processing model and it's **not** a replacement of Hadoop.

Eclipse – Scala IDE integration

Scala is native language for Spark programming. Scala is a programming language designed to express common programming patterns in a concise, well-designed, and type-safe way. It smoothly integrates features of object-oriented and functional languages.

- Scala is object-oriented
- Scala is functional
- Scala is statically typed
- Scala is extensible

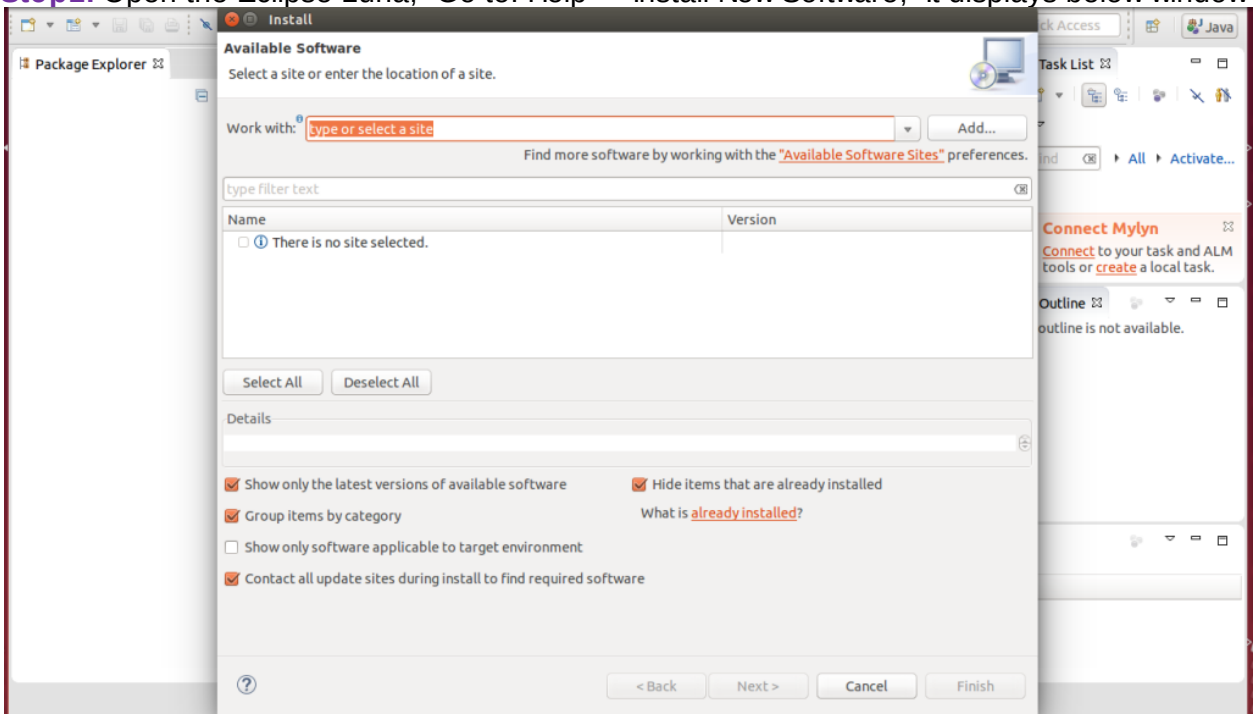
Technical Specification

Eclipse Luna
Java 1.7
Hadoop-2.4.1
Spark-1.5.2-bin-hadoop2.4
Scala-2.10.4
SBT 0.13.6

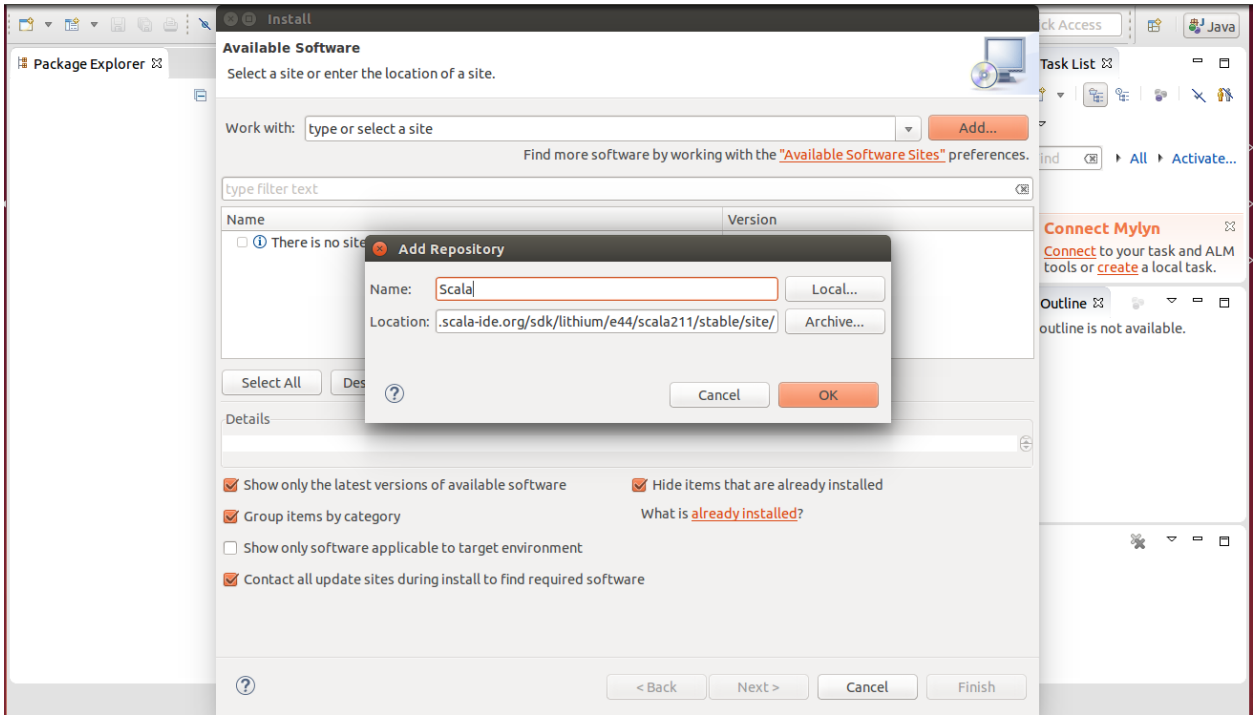
Eclipse – Scala IDE integration steps

Follow below sequence of steps for the integration.

Step1: Open the Eclipse Luna, Go to: Help -> Install New Software, It displays below window

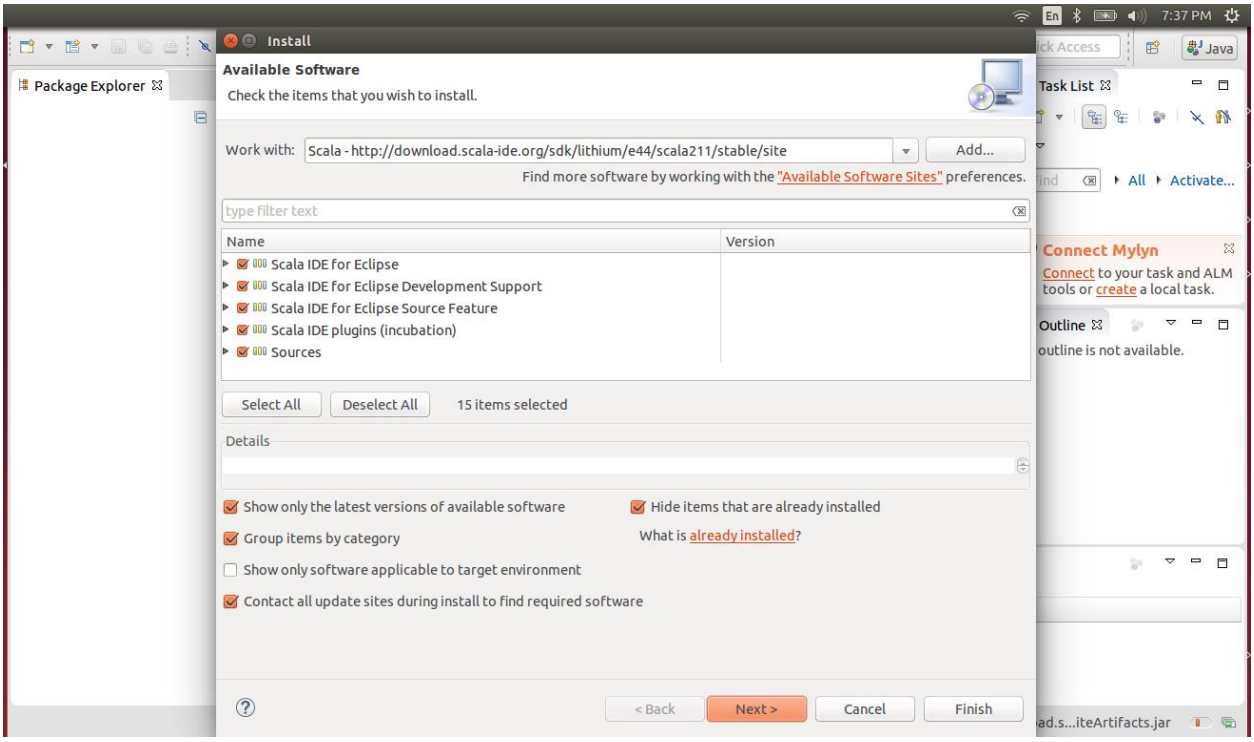


Step2: select Add button, it displays below pop-up window

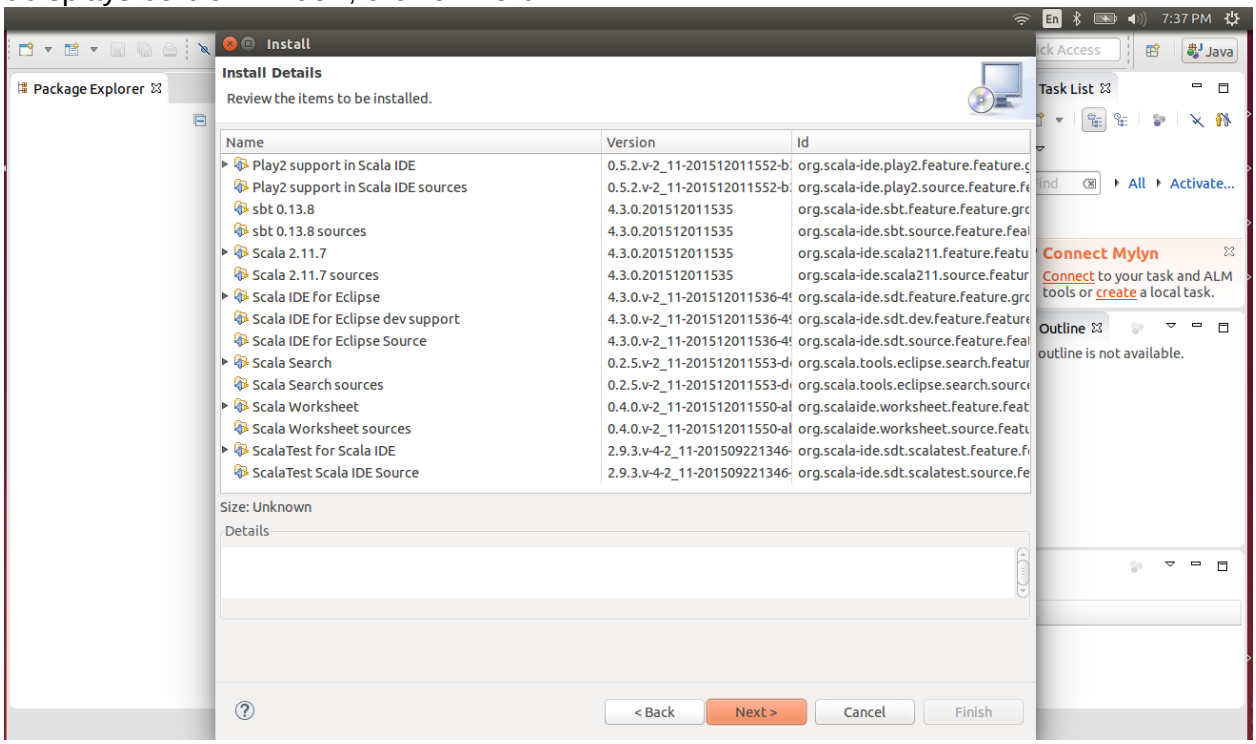


Specify below details and click on Ok
In Name Text box : Scala
In Location : http://download.scala-ide.org/sdk/helium/e44/scala210/stable/site/

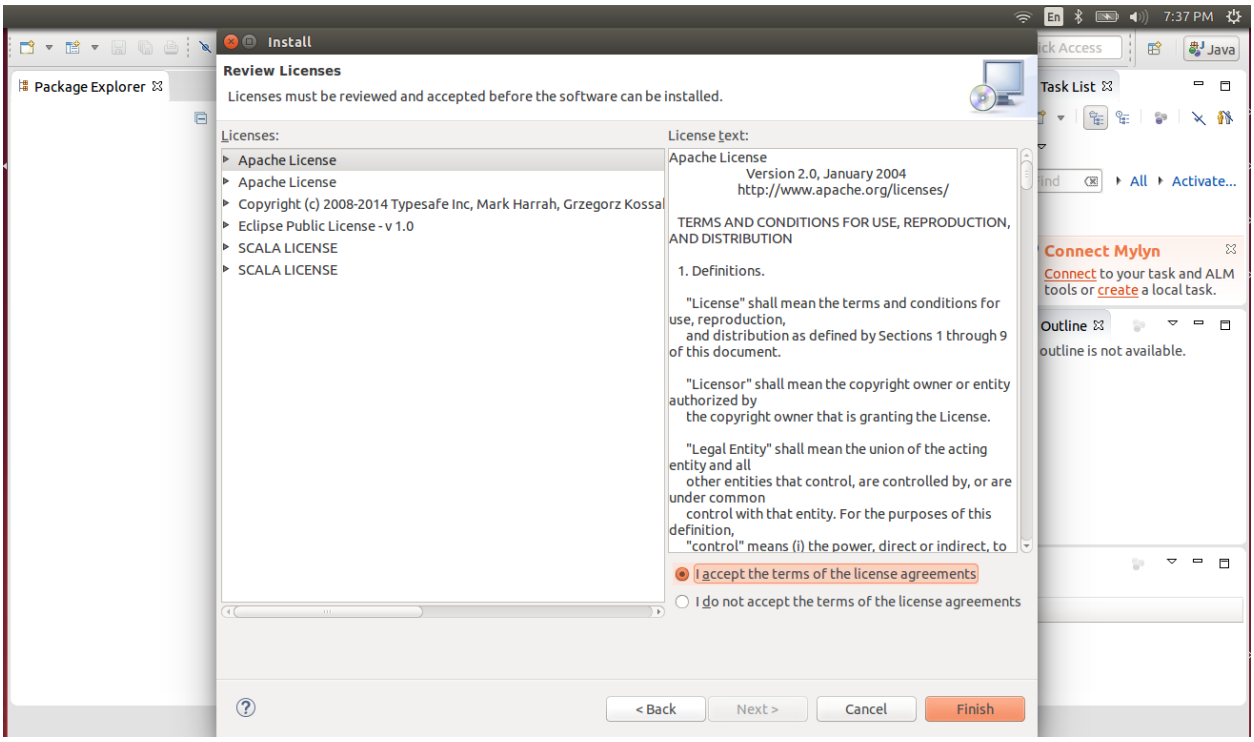
Step 3: It displays below window, then select All -> Next



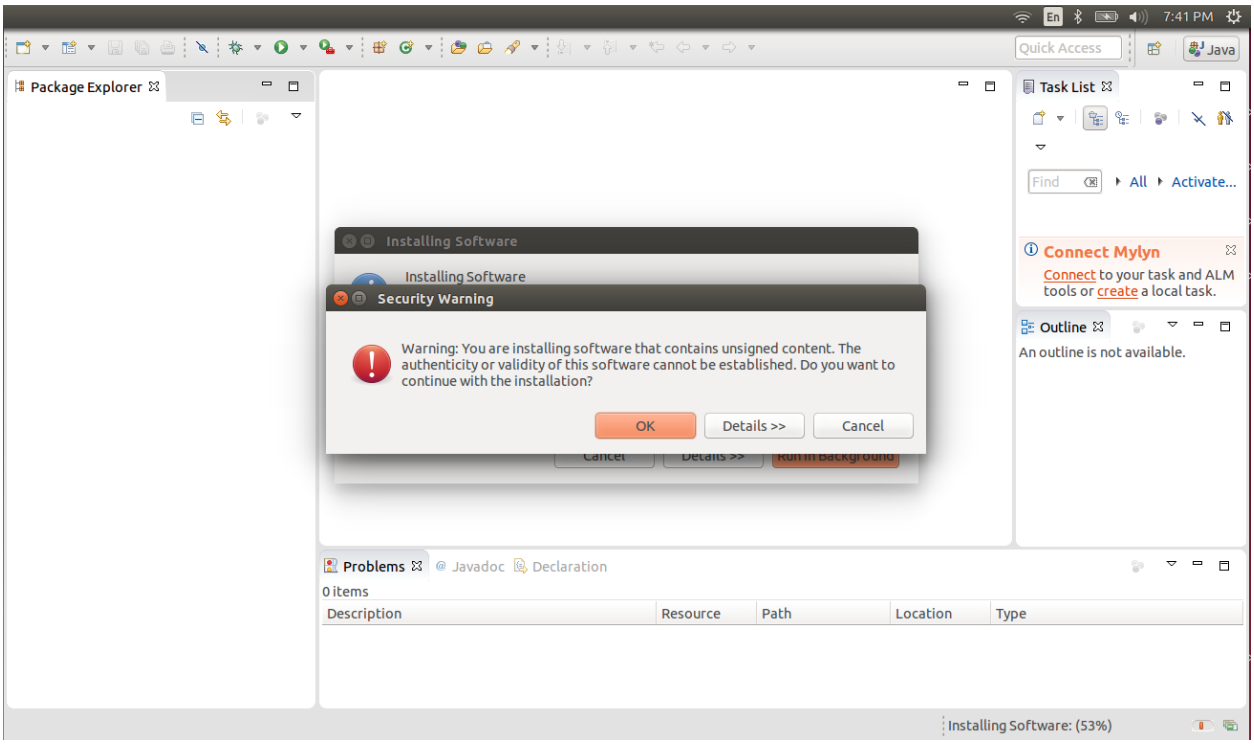
It displays be blow window, click on Next



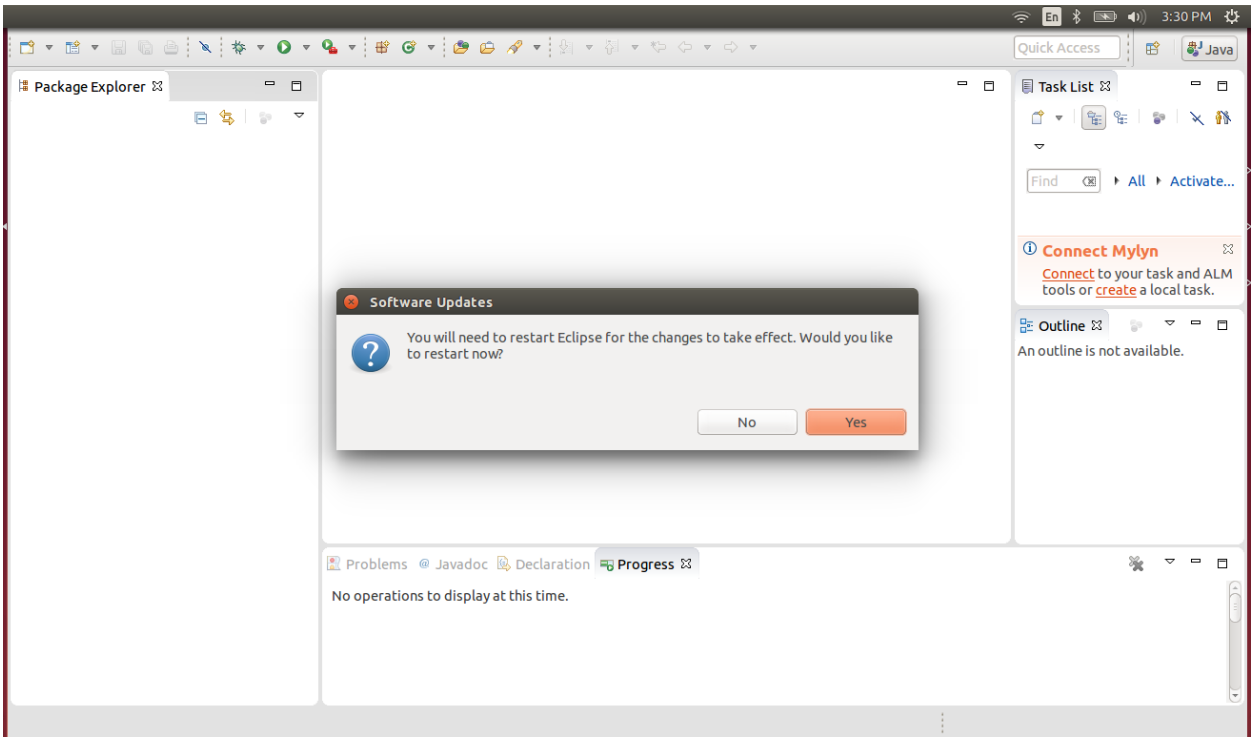
Step 4: it displays below window, accept the terms of the license agreements and click on finish button, it take couple of minute to download the Scala IDE Plug-in.



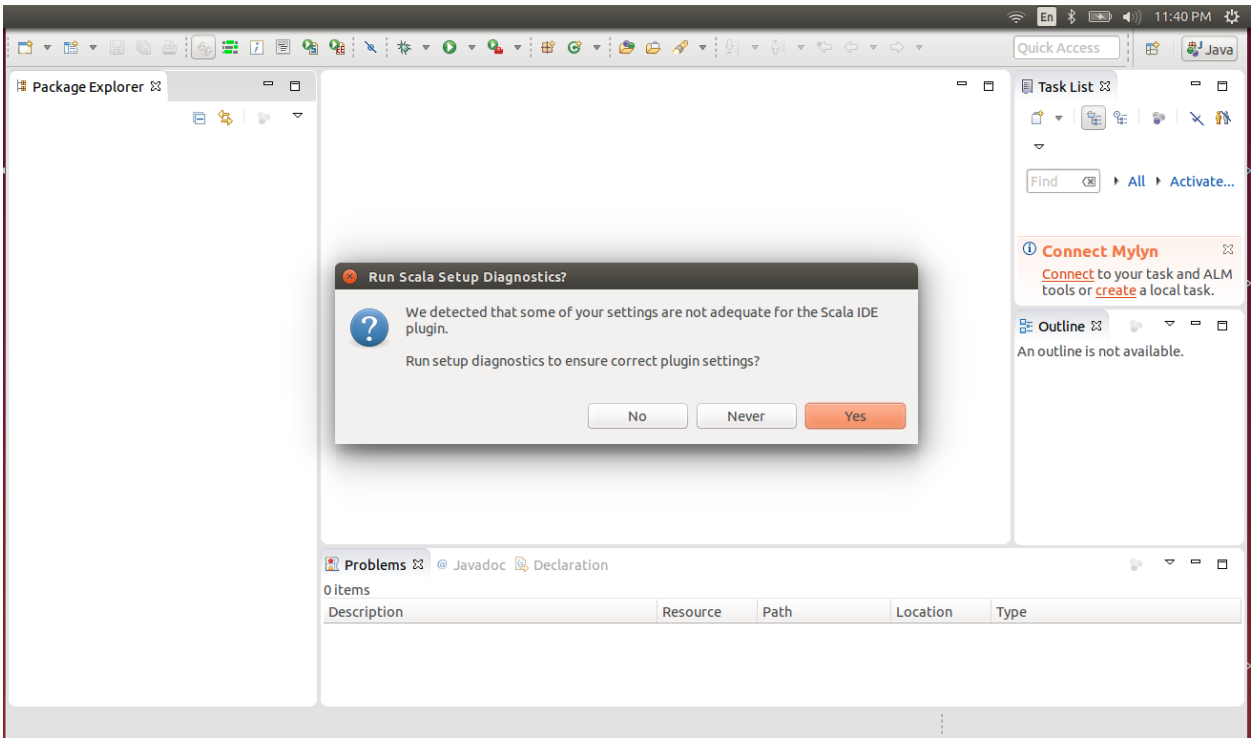
Step 5: During downloading if it displays below pup-up window with warnings, then click on Ok.



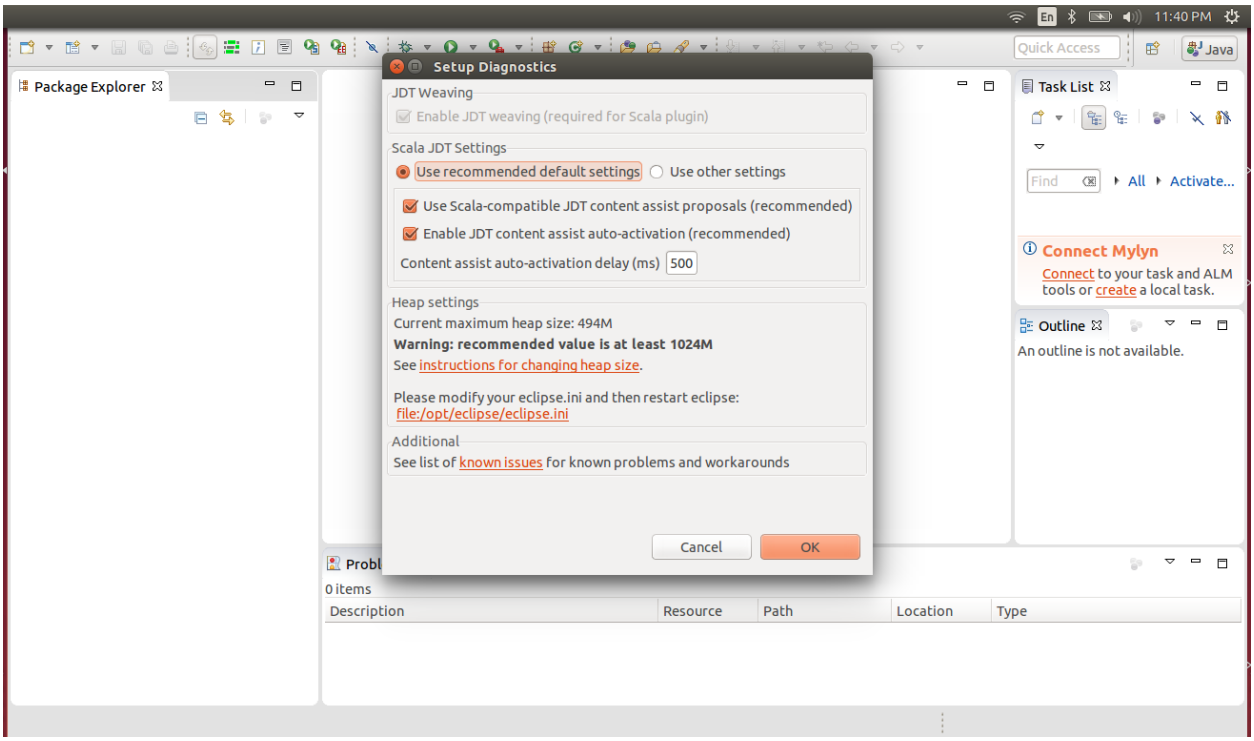
Step 6: Once Download is completed, it's asking for restarting of eclipse. i.e., it displays below pop-up window, click on Yes.



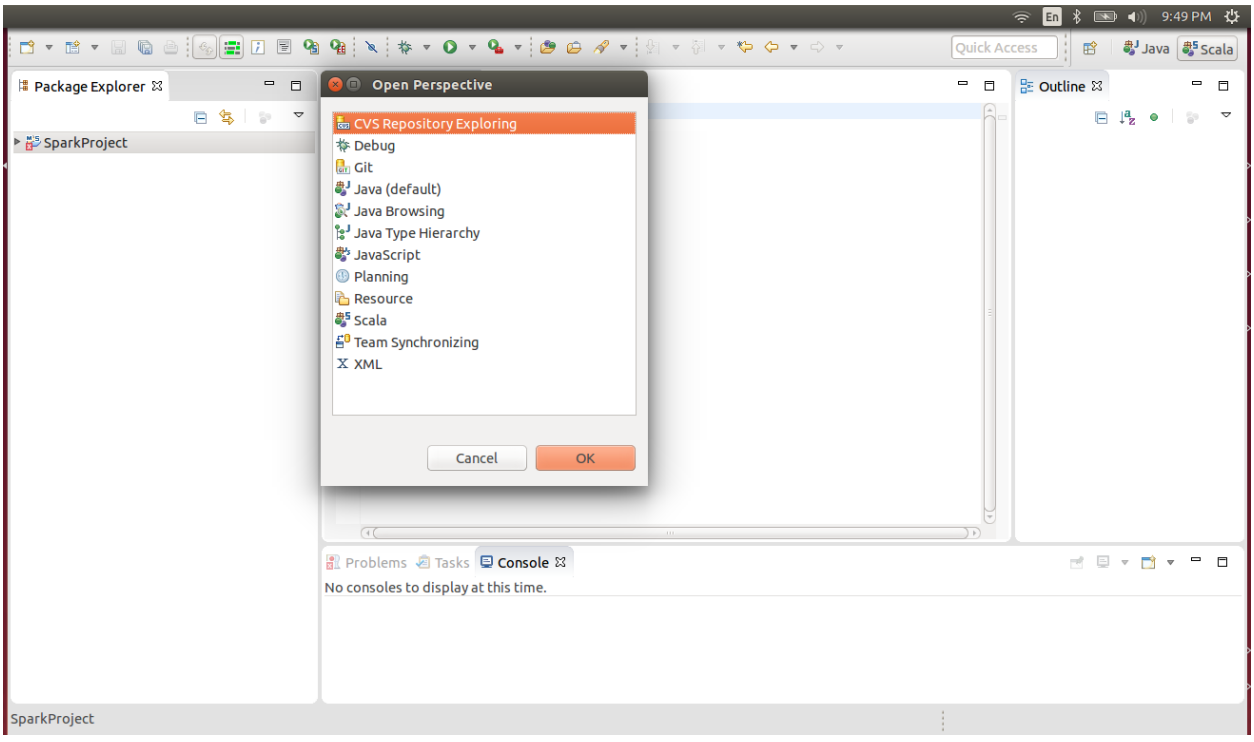
Step 7: Once Eclipse is restarted, it will display below window, click on Yes.



Step 8: Select " User recommended default setting " radio button and click on Ok



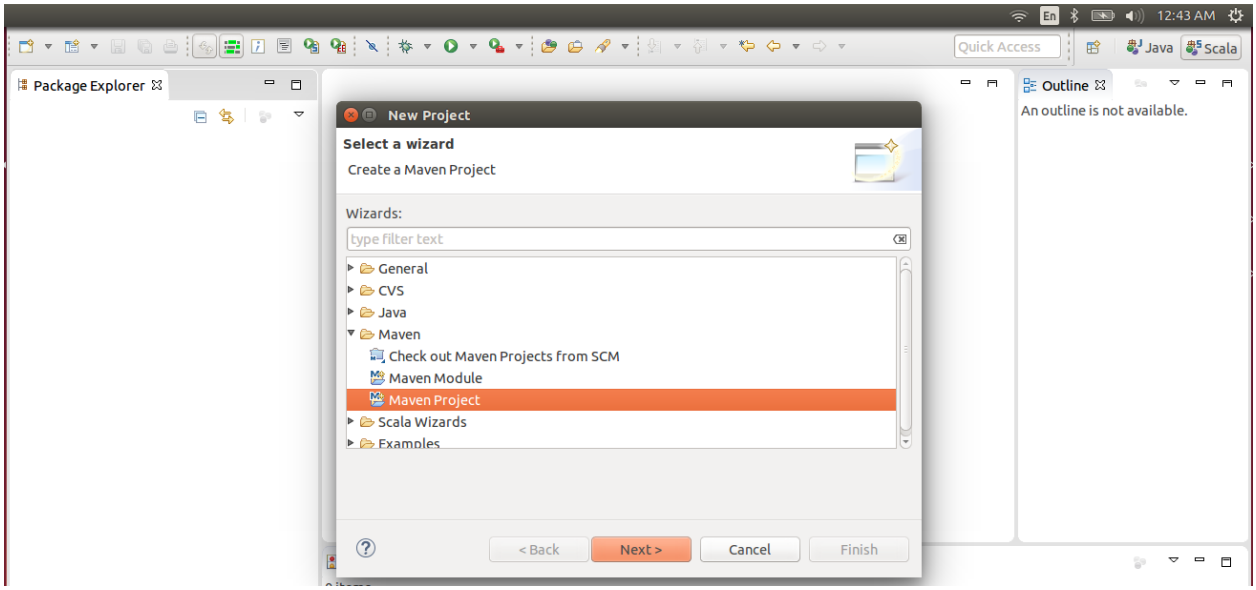
Now, Scala plug-in is integrated with eclipse. It displays Scala on right most corner. If it's not displayed, select Open Perspective, it will displays below window, then select Scala



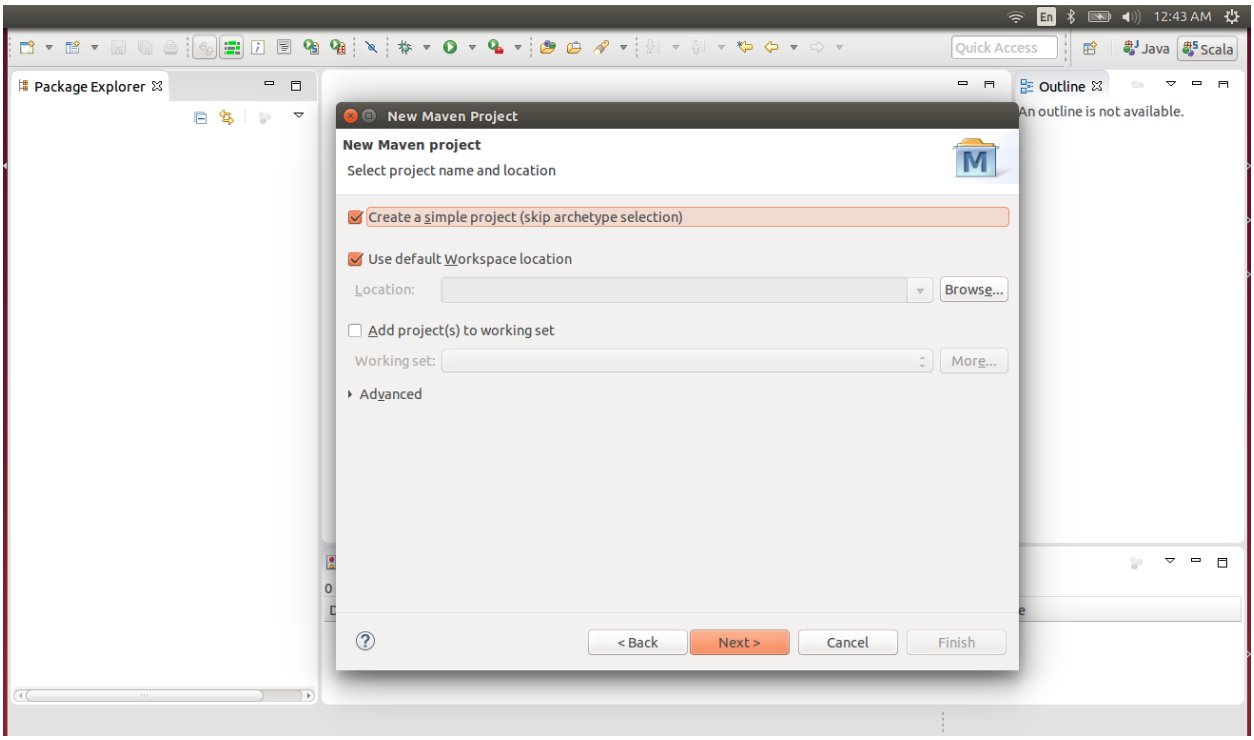
Creating Spark Project

Follow below steps to create Spark Project

Step 1 : Go to: File-> New -> Project -> Maven project and create a maven project.

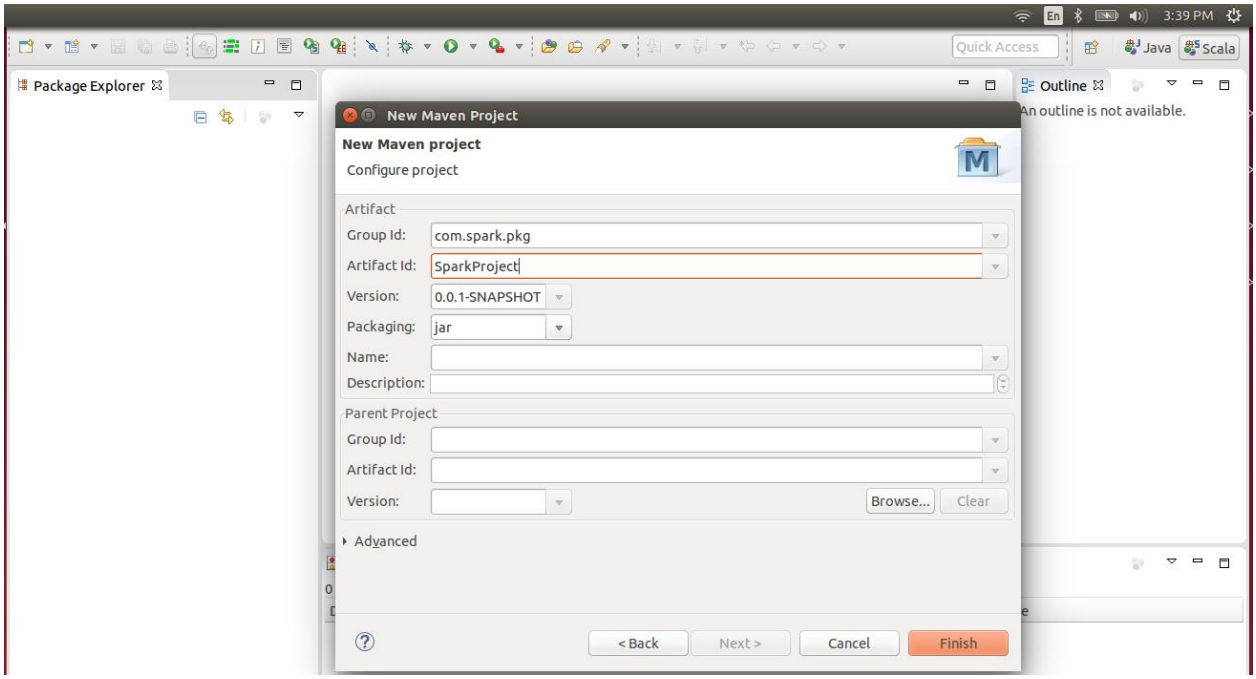


Step2 : Select "*Create a simple project skip (skip archetype selection)*" check box, click on Next



Step 3 : Specify the Group Id and Artifact Id & click finish.

Group Id = com.spark.pkg - Package name
Artifact Id = SparkProject - Project Name

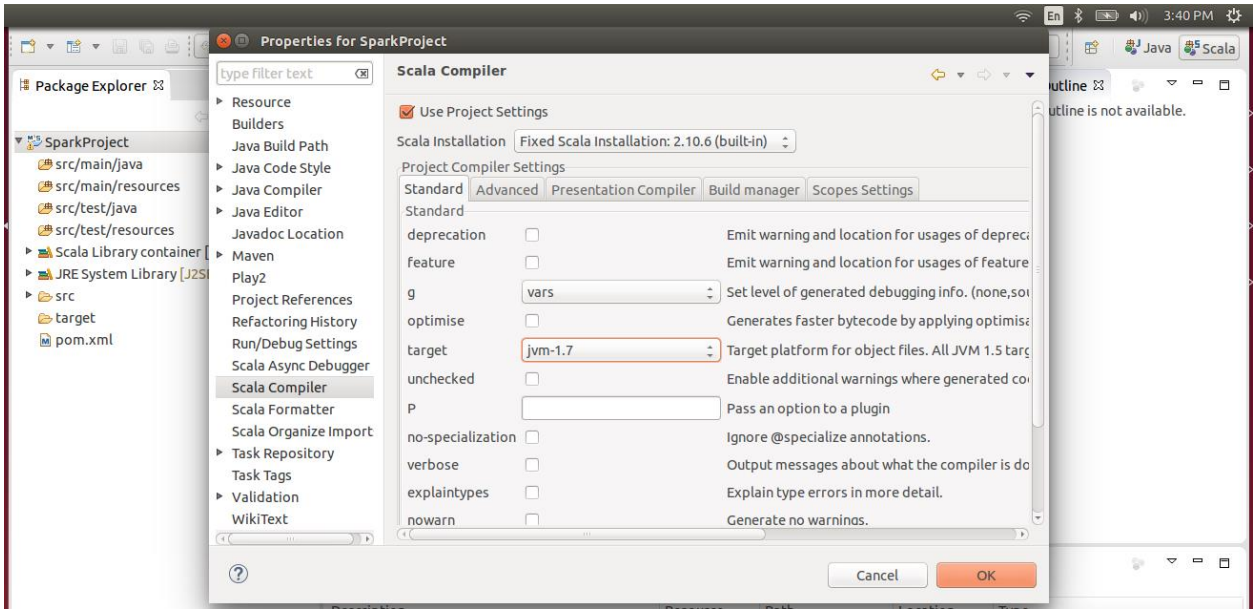


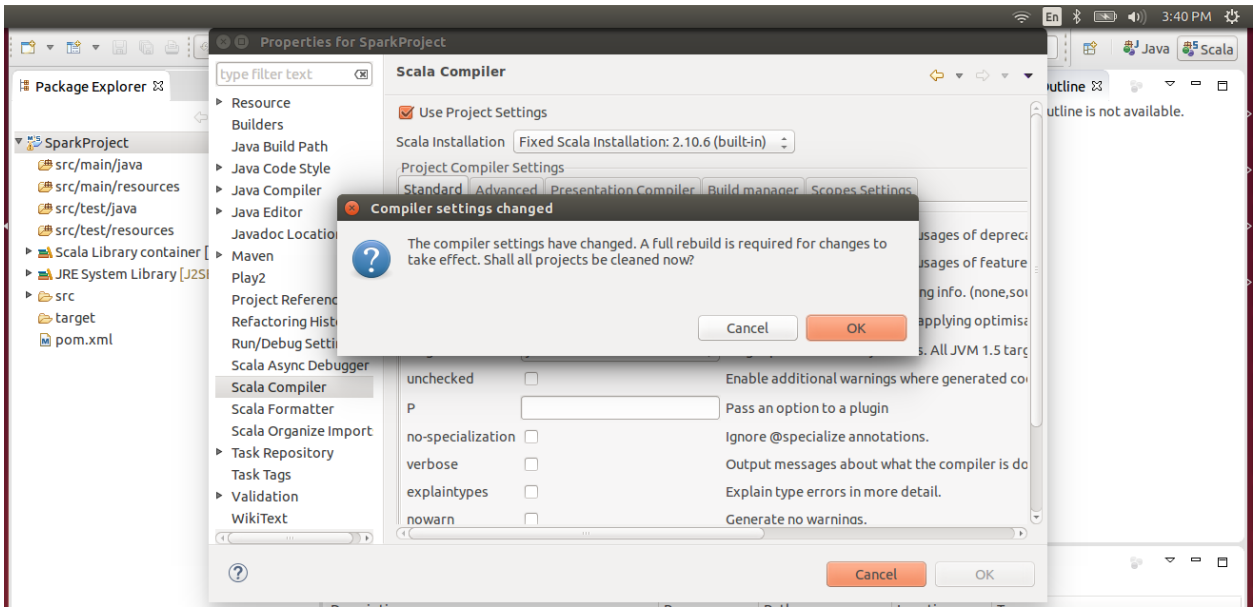
Step 4 : Add Scala Nature to above created project (i.e, SparkProject)
Right click on SparkProject -> configure -> Add Scala Nature.

Step 5 : Update Scala compiler version for Spark
Scala IDE by default uses latest version(2.11) of Scala compiler, however Spark uses version 2.10. So we need to update appropriate version for IDE.

Right click on SparkProject -> Go to properties -> select Scala compiler -> update Scala installation version to 2.10.6

By default Scala compiler settings are disabled, enable those setting by selecting "*Use Projects Settings*" check box and select Fixed Scala Installation: 2.10.6 (built-in) as below and click on Ok.

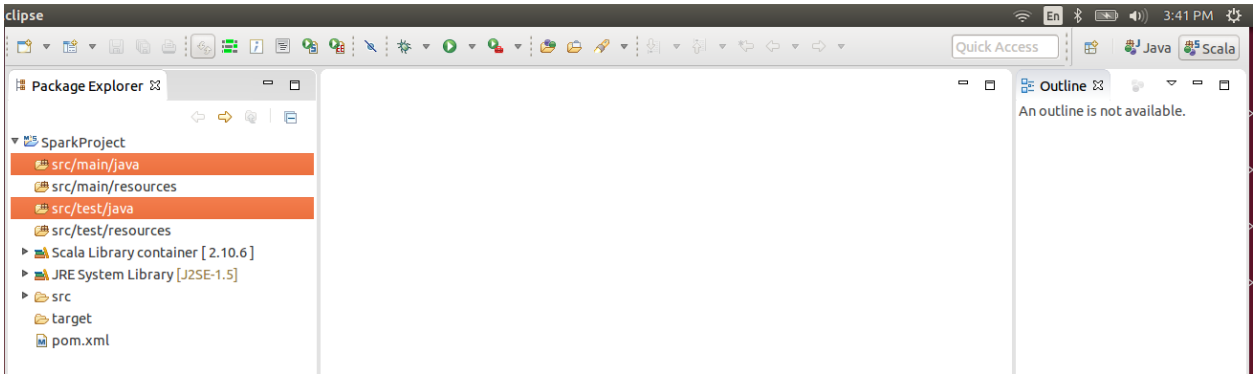

Properties for SparkProject
Scala Compiler
Use Project Settings
Scala Installation
Fixed Scala Installation: 2.10.6 (built-in)
Project Compiler Settings
Standard
Advanced
Presentation Compiler
Build manager
Scopes Settings
deprecation
feature
g
vars
optimise
target
jvm-1.7
unchecked
P
no-specialization
verbose
explaintypes
nowarn
Cancel
OK
Package Explorer
SparkProject
src/main/java
src/main/resources
src/test/java
src/test/resources
target
pom.xml

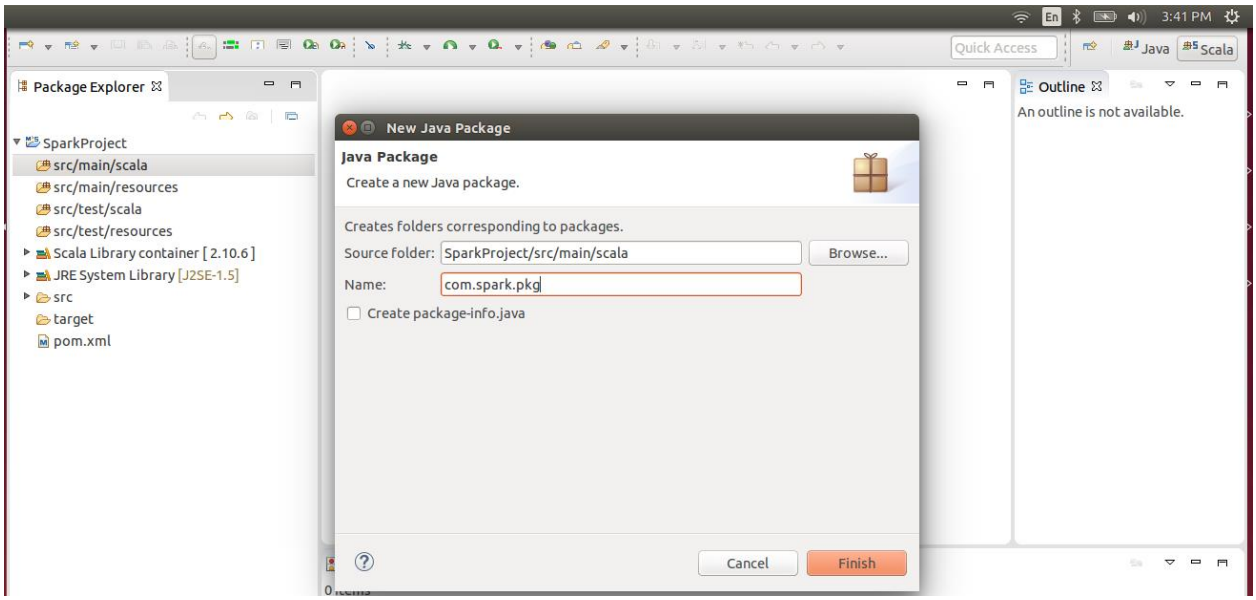

Properties for SparkProject
Scala Compiler
Use Project Settings
Scala Installation
Fixed Scala Installation: 2.10.6 (built-in)
Compiler settings changed
The compiler settings have changed. A full rebuild is required for changes to take effect. Shall all projects be cleaned now?
Cancel
OK
Package Explorer
SparkProject

Step 6 : Update source folder src/main/**java** to src/main/**scala** and src/test/**java** to src/test/**scala**

(Right click -> Refactor -> Rename to scala) or simply press F2 and change it.

and create package com.spark.pkg under src/main/scala.



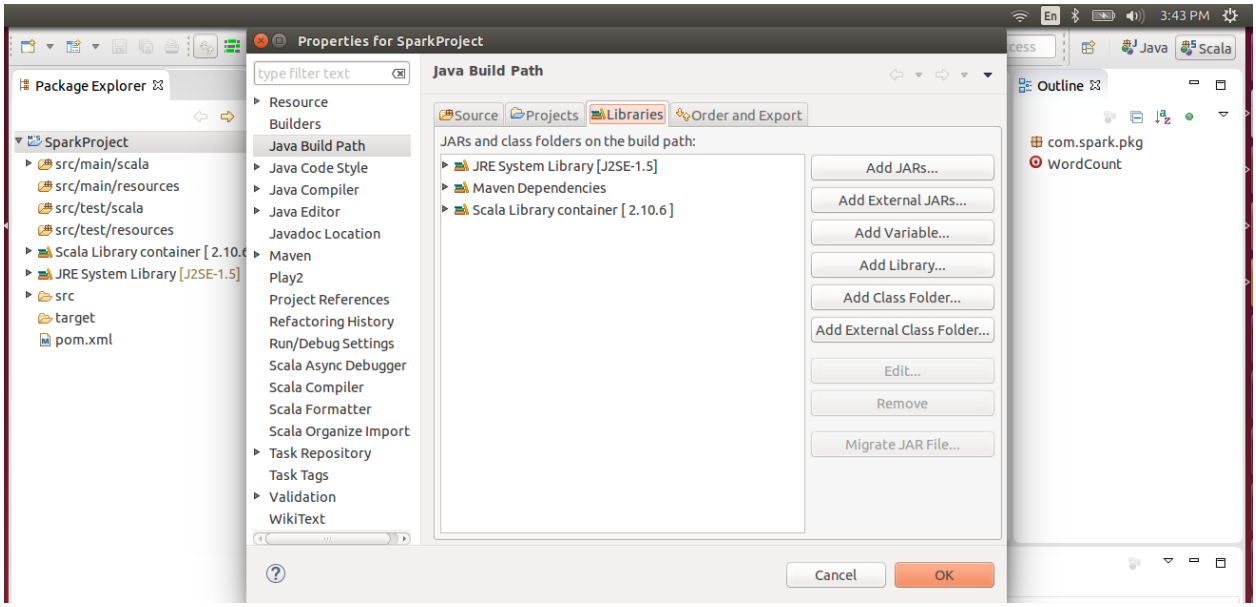


Step 7: Add all supporting Jars

Right click on SparkProject -> Build Path -> Configure BuildPath

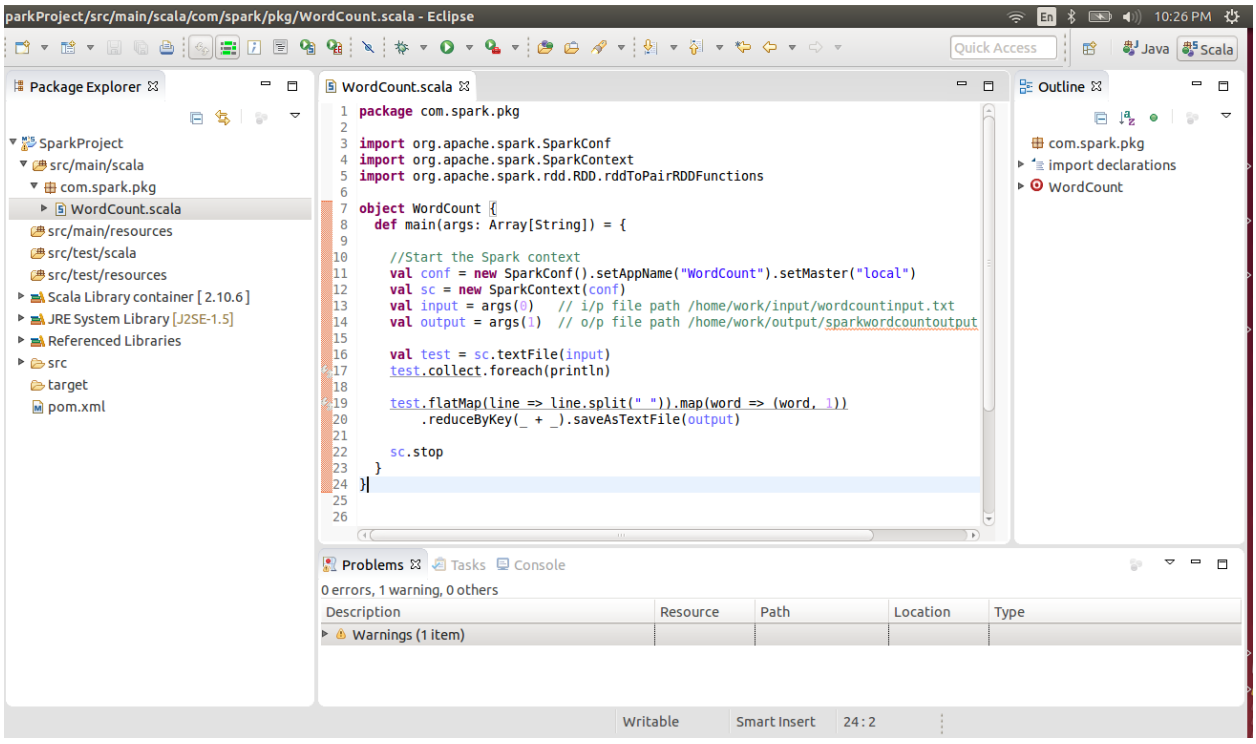
It displays new window,
select Libraries tab
select Add External JARs -> it displays browse window select supporting jars and click on Ok

Supporting external jars are
1) jars from spark i.e., spark-1.5.2-bin-hadoop2.4/lib
2) jars from scala-2.10.4/lib



Step 8: Create new Scala Object WordCount.scala under com.spark.pkg package.

Write the below code



Step 9: Execute wordcount program from eclipse

Right click on WordCount.scala - > Run as -> Scala application. It will create an output directory Sparkwordcountoutput and it will contain two file : part-00000 and _SUCCESS.

Note: Need to provide input and output file paths.

Sample output in part-00000 is :-

```
[redacted]@[redacted]:~$ cat /home/[redacted]/sparkwordcountoutput/part-00000
(scala,2)
(spark,4)
(is,3)
(native,1)
(bright,1)
(source,1)
(processing,1)
(language,1)
(future,1)
(for,1)
(and,1)
(engine,1)
(has,1)
(open,1)
```

Creating SBT

SBT is small Build Tool

Above eclipse program is stored in workspace/SparkProject folder. Initially this folder contains below files

```
[redacted]@[redacted]:~/workspace/SparkProject$ ls
pom.xml  project  src  target
```

Preparing SBT for SparkProject in order to run in Cluster.

Step1: Create new file built.sbt with following content and place it in workspace/SparkProject folder.

```
[redacted]@[redacted]:~$ cat workspace/SparkProject/build.sbt
name := "Build Project"

version := "1.0"

scalaVersion := "2.10.4"

libraryDependencies += "org.apache.spark" %% "spark-core" % "1.5.1"

libraryDependencies += "org.apache.spark" %% "spark-sql" % "1.5.1"

libraryDependencies += "org.apache.spark" %% "spark-hive" % "1.5.1"

libraryDependencies += "org.apache.spark" %% "spark-streaming" % "1.5.1"

libraryDependencies += "org.apache.spark" %% "spark-mllib" % "1.5.1"

libraryDependencies += "org.apache.spark" %% "spark-graphx" % "1.5.1"

libraryDependencies += "org.apache.hadoop" % "hadoop-client" % "2.4.1"
```

Step 2: Sbt package
First time, it will take couple minutes to download the supporting dependencies. For successful creation of SBT, it will display success message.

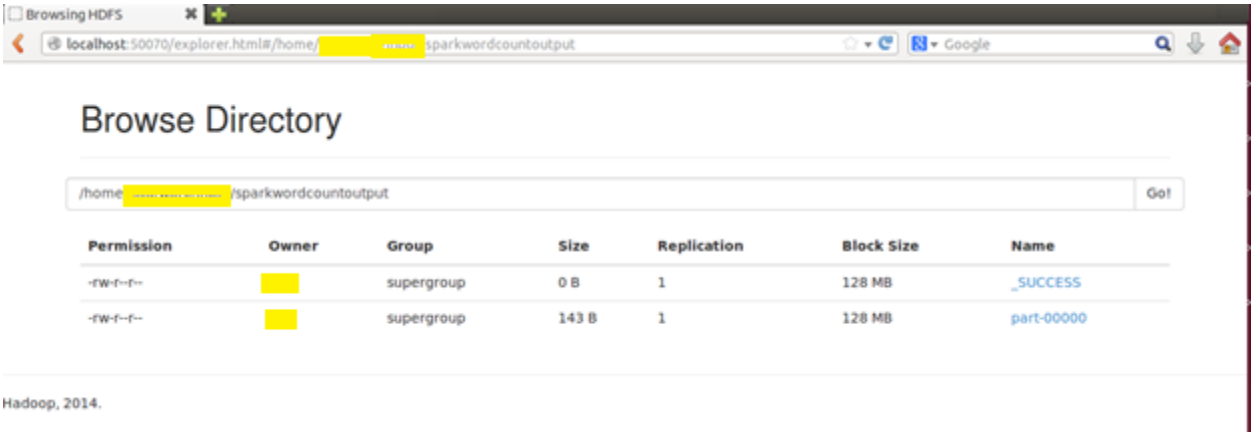
Once it's successfully created, it will create new folder/files in /Workspace/SparkProjct directory. The Jar which is highlighted in red color, used to run the spark application.

```
[redacted]@[redacted]:~/workspace/SparkProject$ sbt package
[info] Set current project to Build Project (in build file:/home/[redacted]/workspace/SparkProject/)
[success] [redacted]
[redacted]@[redacted]:~/workspace/SparkProject$ ls
build.sbt  pom.xml  project  src  target
[redacted]@[redacted]:~/workspace/SparkProject$ cd target
[redacted]@[redacted]:~/workspace/SparkProject/target$ ls
classes  resolution-cache  scala-2.10  streams  test-classes
[redacted]@[redacted]:~/workspace/SparkProject/target$ cd scala-2.10
[redacted]@[redacted]:~/workspace/SparkProject/target/scala-2.10$ ls
build-project_2.10-1.0.jar  classes
[redacted]@[redacted]:~/workspace/SparkProject/target/scala-2.10$
```

Submitting Spark Application in local & Yarn

Step 1: Submit the spark app by specifying –master Loca[4] parameter

xxxxx@xxxxx:~/spark-1.5.2-bin-hadoop2.4$ bin/spark-submit --class com.spark.pkg.WordCount **--master local[4]** /home/xxx/workspace/SparkProject/target/scala-2.10/build-project_2.10-1.0.jar /home/xxx/input/wordcountinput.txt /home/xxx/output/sparkwordcountoutput



Step 2: Submit the spark app in yarn

xxx@xxx:~/spark-1.5.2-bin-hadoop2.4$ bin/spark-submit --class com.spark.pkg.WordCount **--masteryarn --deploy-mode client** /home/xxx/workspace/SparkProject/target/scala-2.10/build-project_2.10-1.0.jar /home/xxx/input/wordcountinput.txt /home/xxx/output/sparkwordcountoutputyarn

Conclusion

From this white paper, you have learned the following:

* Fair understanding on Eclipse - Scala IDE integration.
* What dependencies needs to add, in order to run spark application in eclipse.
* Preparing SBT for SparkProject in order to run in Cluster.

Thanks for reading, I hope this white paper helps you on basic understanding/setup of integrations as well as running of simple spark application.