Apache Maven Usage - Hadoop

From the history GNU ‘make utility’ has been a proven successful build tool to help developers manage project involving multiple source files. Target/Goal defined in the project specific makefile when utilized by utility ‘make’ intelligently performs compilation and more.

During the development of java world, apache ant played similar role as ‘make’ utility.

In the row when dependencies of jars containing many more library packages was becoming a normal scenario, apache maven hosted repositories and using the project object model (.pom) let developers define the library with artifactID and version, etc in the project to automatically download and manage dependencies.

Hence, maven is known for its managing dependencies automatically which relieves programmer from setting up classpath with external jars manually.

You must specify a valid lifecycle phase or a goal in the format <plugin-prefix>:<goal> or <plugin-group-id>:<plugin-artifact-id>[:<plugin-version>]:<goal>. Available lifecycle phases are: validate, initialize, generate-sources, process-sources, generate-resources, process-resources, compile, process-classes, generate-test-sources, process-test-sources, generate-test-resources, process-test-resources, test-compile, process-test-classes, test, prepare-package, package, pre-integration-test, integration-test, post-integration-test, verify, install, deploy, pre-clean, clean, post-clean, pre-site, site, post-site, site-deploy.

P.s. Know that eclipse like IDE support maven based development. Sometimes GUI menu item may read ‘build’ but internally it is mapped to ‘compile’

Hello World - Simple Java Program with maven command line

Below is the recorded script of sample maven demo program which also talks about test driven development. Hello World of Java.

****
Script started on Sat Aug 1 09:40:46 2020
[?1034hbash-3.2$ tree
.
├── build
├── mvndemoscript
├── pom.xml
└── src
    ├── main
    │   └── java
    │   └── com
    │   └── example
    │   ├── Greeter.java
    │   └── HelloWorld.java
    ├── resources
    │   └── META-INF
    │   └── MANIFEST.MF
    └── test
        └── java
            └── com
                └── example
                    └── GreeterTest.java

12 directories, 6 files
bash-3.2$ mvn test
[[1;34mINFO[m] Scanning for projects...
[[1;34mINFO[m]
[[1;34mINFO[m] [1m------------------------< [0;36mMavens:demo-maven[0;1m
>-------------------------[m
[[1;34mINFO[m] [1mBuilding demo-maven 0.0.1-SNAPSHOT[m
[[1;34mINFO[m] [1m--------------------------------[jar]---------------------------------[m
[[1;34mINFO[m]
[[1;34mINFO[m] [1m--- [0;32mmaven-resources-plugin:2.6:resources[m
[1m(default-resources)[m @ [36mdemo-maven[0;1m ---[m
[[1;33mWARNING[m] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!

```
[[1;34mINFO[m] skip non existing resourceDirectory
/Users/JigarPandya/Desktop/MavenDemoJavaApplication/src/main/resources
[[1;34mINFO[m]
[[1;34mINFO[m] [1m--- [0;32mmaven-compiler-plugin:3.1:compile[m [1m(default-compile)[m @
[36mdemo-maven[0;1m ---[m
[[1;34mINFO[m] Changes detected - recompiling the module!
[[1;33mWARNING[m] File encoding has not been set, using platform encoding UTF-8, i.e. build
is platform dependent!
[[1;34mINFO[m] Compiling 2 source files to
/Users/JigarPandya/Desktop/MavenDemoJavaApplication/build/classes
[[1;34mINFO[m]
[[1;34mINFO[m] [1m--- [0;32mmaven-resources-plugin:2.6:testResources[m
[1m(default-testResources)[m @ [36mdemo-maven[0;1m ---[m
[[1;33mWARNING[m] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e.
build is platform dependent!
[[1;34mINFO[m] skip non existing resourceDirectory
/Users/JigarPandya/Desktop/MavenDemoJavaApplication/src/test/resources
[[1;34mINFO[m]
[[1;34mINFO[m] [1m--- [0;32mmaven-compiler-plugin:3.1:testCompile[m
[1m(default-testCompile)[m @ [36mdemo-maven[0;1m ---[m
[[1;34mINFO[m] Changes detected - recompiling the module!
[[1;33mWARNING[m] File encoding has not been set, using platform encoding UTF-8, i.e. build
is platform dependent!
[[1;34mINFO[m] Compiling 1 source file to
/Users/JigarPandya/Desktop/MavenDemoJavaApplication/target/test-classes
[[1;34mINFO[m]
[[1;34mINFO[m] [1m--- [0;32mmaven-surefire-plugin:2.12.4:test[m [1m(default-test)[m @
[36mdemo-maven[0;1m ---[m
[[1;34mINFO[m] Surefire report directory:
/Users/JigarPandya/Desktop/MavenDemoJavaApplication/target/surefire-reports

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running GreeterTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.246 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[[1;34mINFO[m] [1m------------------------------------------------------------------------[m
[[1;34mINFO[m] [1;32mBUILD SUCCESS[m
```

```
[[1;34mINFO[m] [1m------------------------------------------------------------------------[m
[[1;34mINFO[m] Total time:  9.942 s
[[1;34mINFO[m] Finished at: 2020-08-01T09:41:15+05:30
[[1;34mINFO[m] [1m------------------------------------------------------------------------[m
bash-3.2$ tree
.
├── build
│   └── classes
│       └── com
│           └── example
│               ├── Greeter.class
│               └── HelloWorld.class
├── mvndemoscript
├── pom.xml
├── src
│   ├── main
│   │   └── java
│   │       └── com
│   │           └── example
│   │               ├── Greeter.java
│   │               └── HelloWorld.java
│   ├── resources
│   │   └── META-INF
│   │       └── MANIFEST.MF
│   └── test
│       └── java
│           └── com
│               └── example
│                   └── GreeterTest.java
└── target
    ├── generated-sources
    │   └── annotations
    ├── generated-test-sources
    │   └── test-annotations
    ├── maven-status
    │   └── maven-compiler-plugin
    │       ├── compile
    │       │   └── default-compile
    │       │       ├── createdFiles.lst
    │       │       └── inputFiles.lst
    │       └── testCompile
    │           └── default-testCompile
    │               ├── createdFiles.lst
```

```
│   └── inputFiles.lst
├── surefire-reports
│   ├── GreeterTest.txt
│   └── TEST-GreeterTest.xml
└── test-classes
    └── GreeterTest.class

28 directories, 15 files
bash-3.2$ mvn install
[[1;34mINFO[m] Scanning for projects...
[[1;34mINFO[m]
[[1;34mINFO[m] [1m------------------------< [0;36mMavens:demo-maven[0;1m
>-------------------------[m
[[1;34mINFO[m] [1mBuilding demo-maven 0.0.1-SNAPSHOT[m
[[1;34mINFO[m] [1m--------------------------------[ jar ]---------------------------------[m
[[1;34mINFO[m]
[[1;34mINFO[m] [1m--- [0;32mmaven-resources-plugin:2.6:resources[m
[1m(default-resources)[m @ [36mdemo-maven[0;1m ---[m
[[1;33mWARNING[m] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e.
build is platform dependent!
[[1;34mINFO[m] skip non existing resourceDirectory
/Users/JigarPandya/Desktop/MavenDemoJavaApplication/src/main/resources
[[1;34mINFO[m]
[[1;34mINFO[m] [1m--- [0;32mmaven-compiler-plugin:3.1:compile[m [1m(default-compile)[m @
[36mdemo-maven[0;1m ---[m
[[1;34mINFO[m] Nothing to compile - all classes are up to date
[[1;34mINFO[m]
[[1;34mINFO[m] [1m--- [0;32mmaven-resources-plugin:2.6:testResources[m
[1m(default-testResources)[m @ [36mdemo-maven[0;1m ---[m
[[1;33mWARNING[m] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e.
build is platform dependent!
[[1;34mINFO[m] skip non existing resourceDirectory
/Users/JigarPandya/Desktop/MavenDemoJavaApplication/src/test/resources
[[1;34mINFO[m]
[[1;34mINFO[m] [1m--- [0;32mmaven-compiler-plugin:3.1:testCompile[m
[1m(default-testCompile)[m @ [36mdemo-maven[0;1m ---[m
[[1;34mINFO[m] Changes detected - recompiling the module!
[[1;33mWARNING[m] File encoding has not been set, using platform encoding UTF-8, i.e. build
is platform dependent!
[[1;34mINFO[m] Compiling 1 source file to
/Users/JigarPandya/Desktop/MavenDemoJavaApplication/target/test-classes
[[1;34mINFO[m]
```

```
[[1;34mINFO[m] [1m--- [0;32mmaven-surefire-plugin:2.12.4:test[m [1m(default-test)[m @
[36mdemo-maven[0;1m ---[m
[[1;34mINFO[m] Surefire report directory:
/Users/JigarPandya/Desktop/MavenDemoJavaApplication/target/surefire-reports

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running GreeterTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.187 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[[1;34mINFO[m]
[[1;34mINFO[m] [1m--- [0;32mmaven-jar-plugin:3.1.0:jar[m [1m(default-jar)[m @
[36mdemo-maven[0;1m ---[m
[[1;34mINFO[m] Building jar:
/Users/JigarPandya/Desktop/MavenDemoJavaApplication/target/demo-maven-0.0.1-SNAPSHO
T.jar
[[1;34mINFO[m]
[[1;34mINFO[m] [1m--- [0;32mmaven-install-plugin:2.4:install[m [1m(default-install)[m @
[36mdemo-maven[0;1m ---[m
[[1;34mINFO[m] Installing
/Users/JigarPandya/Desktop/MavenDemoJavaApplication/target/demo-maven-0.0.1-SNAPSHO
T.jar to
/Users/JigarPandya/.m2/repository/Mavens/demo-maven/0.0.1-SNAPSHOT/demo-maven-0.0.1-
SNAPSHOT.jar
[[1;34mINFO[m] Installing /Users/JigarPandya/Desktop/MavenDemoJavaApplication/pom.xml to
/Users/JigarPandya/.m2/repository/Mavens/demo-maven/0.0.1-SNAPSHOT/demo-maven-0.0.1-
SNAPSHOT.pom
[[1;34mINFO[m] [1m------------------------------------------------------------------------[m
[[1;34mINFO[m] [1;32mBUILD SUCCESS[m
[[1;34mINFO[m] [1m------------------------------------------------------------------------[m
[[1;34mINFO[m] Total time:  8.533 s
[[1;34mINFO[m] Finished at: 2020-08-01T09:41:35+05:30
[[1;34mINFO[m] [1m------------------------------------------------------------------------[m
bash-3.2$ tree
.
├── build
│   └── classes
│       └── com
```

```
│           └── example
│               ├── Greeter.class
│               └── HelloWorld.class
├── mvndemoscript
├── pom.xml
├── src
│   ├── main
│   │   └── java
│   │       └── com
│   │           └── example
│   │               ├── Greeter.java
│   │               └── HelloWorld.java
│   ├── resources
│   │   └── META-INF
│   │       └── MANIFEST.MF
│   └── test
│       └── java
│           └── com
│               └── example
│                   └── GreeterTest.java
└── target
    ├── demo-maven-0.0.1-SNAPSHOT.jar
    ├── generated-sources
    │   └── annotations
    ├── generated-test-sources
    │   └── test-annotations
    ├── maven-archiver
    │   └── pom.properties
    ├── maven-status
    │   └── maven-compiler-plugin
    │       ├── compile
    │       │   └── default-compile
    │       │       ├── createdFiles.lst
    │       │       └── inputFiles.lst
    │       └── testCompile
    │           └── default-testCompile
    │               ├── createdFiles.lst
    │               └── inputFiles.lst
    ├── surefire-reports
    │   ├── GreeterTest.txt
    │   └── TEST-GreeterTest.xml
    └── test-classes
        └── GreeterTest.class
```

```
29 directories, 17 files
bash-3.2$ m[Kjava -jar target/demo-maven-0.0.1-SNAPSHOT.jar HelloWorld
Hello World of Java.
Demonstrating simple maven build...
bash-3.2$ mvn clean
[[1;34mINFO[m] Scanning for projects...
[[1;34mINFO[m]
[[1;34mINFO[m] [1m------------------------< [0;36mMavens:demo-maven[0;1m
>-------------------------[m
[[1;34mINFO[m] [1mBuilding demo-maven 0.0.1-SNAPSHOT[m
[[1;34mINFO[m] [1m--------------------------------[ jar ]---------------------------------[m
[[1;34mINFO[m]
[[1;34mINFO[m] [1m--- [0;32mmaven-clean-plugin:2.5:clean[m [1m(default-clean)[m @
[36mdemo-maven[0;1m ---[m
[[1;34mINFO[m] Deleting /Users/JigarPandya/Desktop/MavenDemoJavaApplication/target
[[1;34mINFO[m] Deleting
/Users/JigarPandya/Desktop/MavenDemoJavaApplication/build/classes
[[1;34mINFO[m] [1m------------------------------------------------------------------------[m
[[1;34mINFO[m] [1;32mBUILD SUCCESS[m
[[1;34mINFO[m] [1m------------------------------------------------------------------------[m
[[1;34mINFO[m] Total time:  0.891 s
[[1;34mINFO[m] Finished at: 2020-08-01T09:42:19+05:30
[[1;34mINFO[m] [1m------------------------------------------------------------------------[m
bash-3.2$ tree
.
├── build
├── mvndemoscript
├── pom.xml
└── src
    ├── main
    │   └── java
    │       └── com
    │           └── example
    │               ├── Greeter.java
    │               └── HelloWorld.java
    ├── resources
    │   └── META-INF
    │       └── MANIFEST.MF
    └── test
        └── java
            └── com
                └── example
```

```
        └── GreeterTest.java

12 directories, 6 files
bash-3.2$ exit

Script done on Sat Aug  1 09:42:24 2020
**********************************************
```

Hadoop Development using Eclipse and Maven

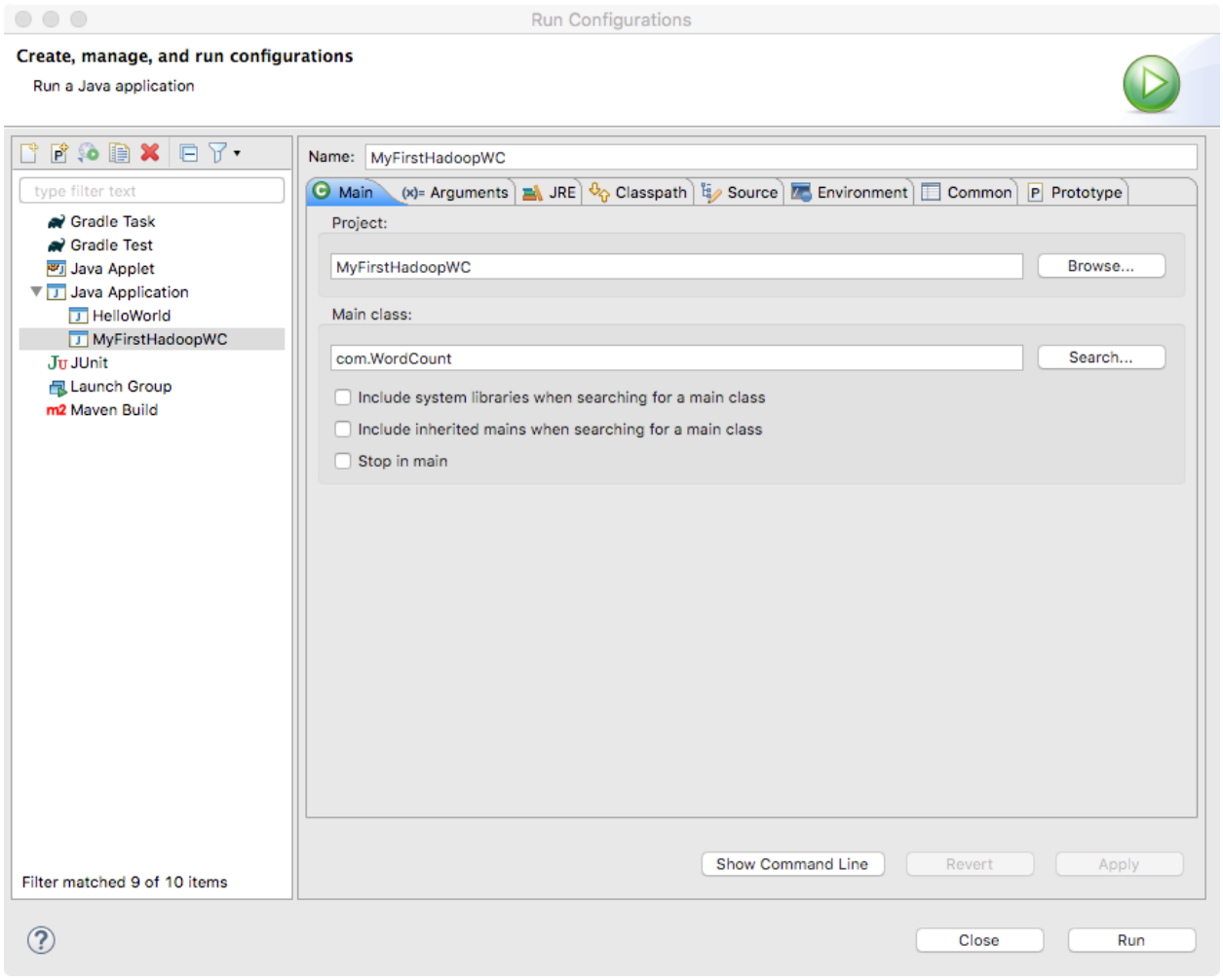
Hadoop Maven based compilation is possible with eclipse like IDE.

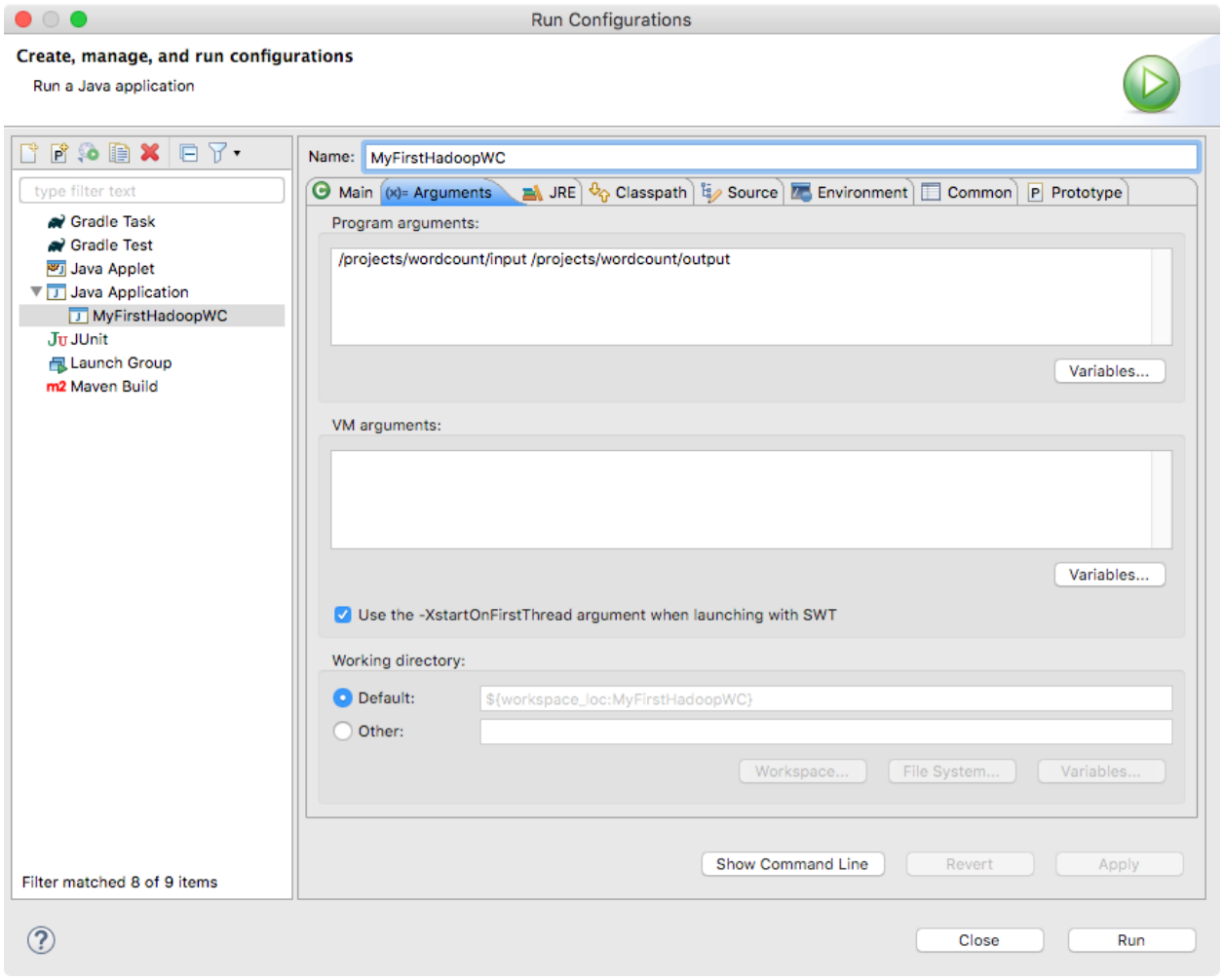
First create a maven project and add a source file in src/main/java

Have entries into pom.xml file.

Optionally may create manifest file to specify the main class name.

Create a run configuration in eclipse. I.e.

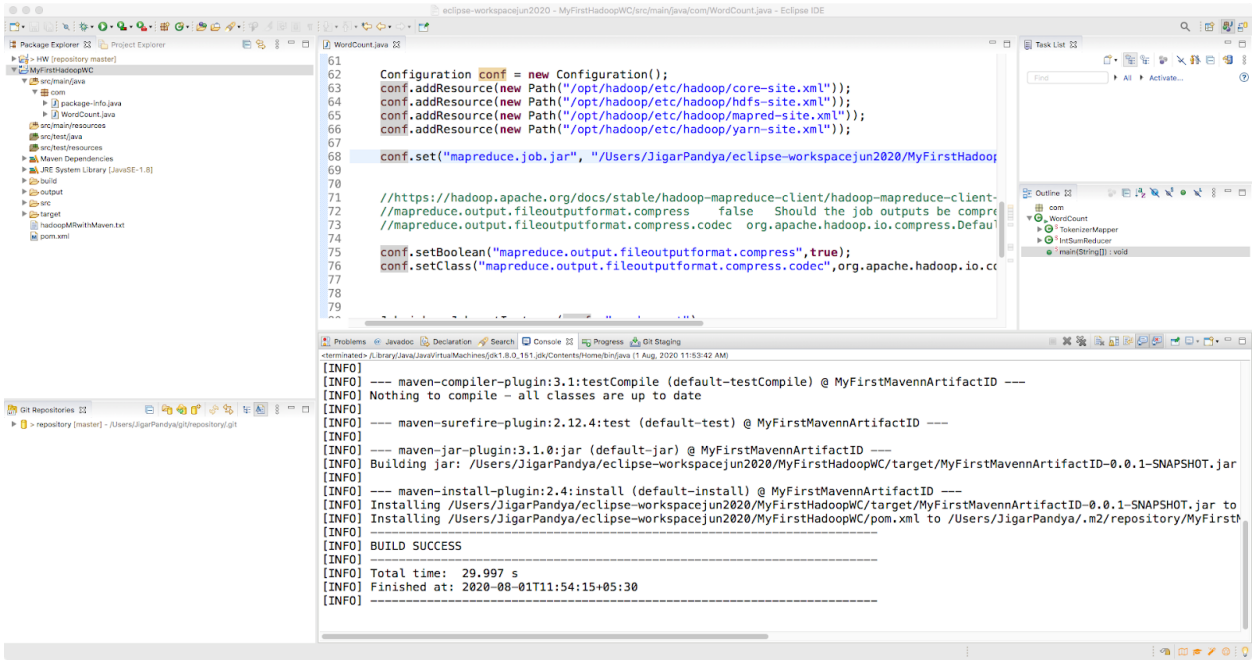

Run Configurations
Create, manage, and run configurations
Run a Java application
type filter text
Gradle Task
Gradle Test
Java Applet
Java Application
HelloWorld
MyFirstHadoopWC
JUnit
Launch Group
Maven Build
Filter matched 9 of 10 items
Name: MyFirstHadoopWC
Main
(x)= Arguments
JRE
Classpath
Source
Environment
Common
Prototype
Project:
MyFirstHadoopWC
Browse...
Main class:
com.WordCount
Search...
Include system libraries when searching for a main class
Include inherited mains when searching for a main class
Stop in main
Show Command Line
Revert
Apply
Close
Run



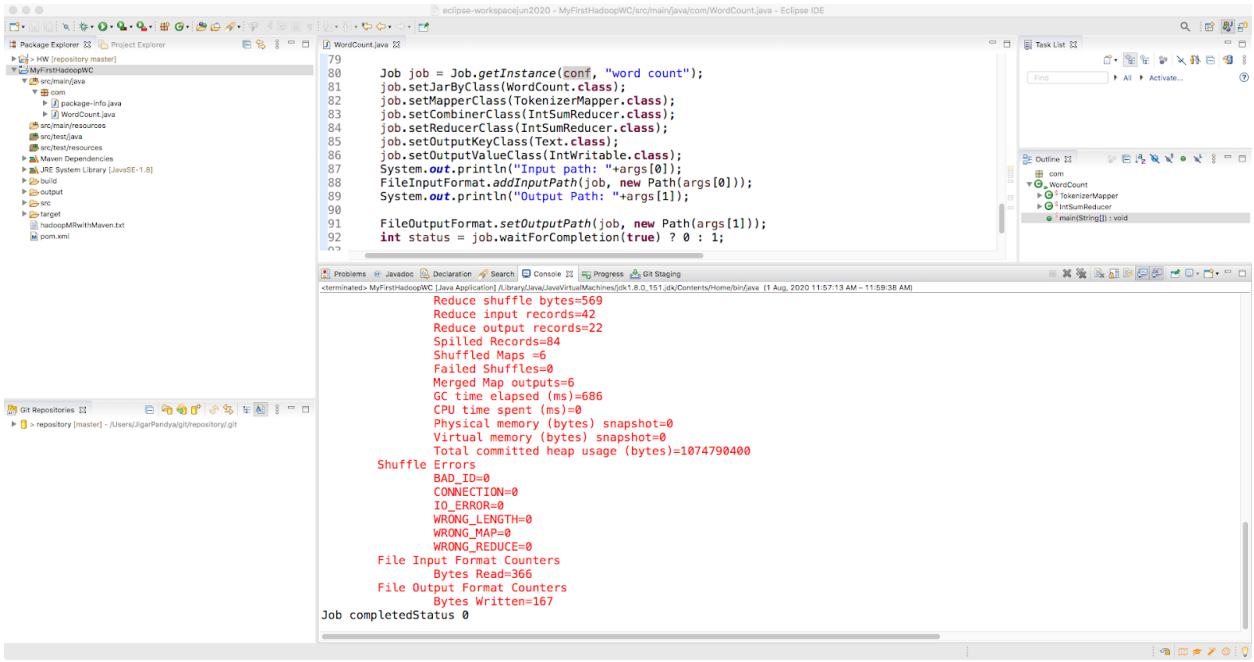
Know that maven runtime downloads dependency jars from web repository and hence internet connection has to be available. May explore local repository option with maven to develop offline.

Eclipse provides menu item on right click of project to act maven targets. I.e. clean, compile (build), , install etc.

As of now, we need a jar created by maven install and the location is specified to be used by run in the program itself so that inner class not found error can be avoided. You may need to uncomment the source code line for configuring jar in the driver program.



Know that maven has validate target but not direct run. Here, we are using eclipse run configurations run menu item or green play symbol to execute the WordCount Map Reduce program. The hadoop components have to be up and running as background processes.



Alternatively, command line is also possible to utilize maven with hadoop

See the steps below:

```
hbash-3.2$ tree
.
├── hadoopMRwithMaven.txt
├── pom.xml
└── src
    ├── main
    │   ├── java
    │   │   └── com
    │   │       ├── WordCount.java
    │   │       └── package-info.java
    │   └── resources
    ├── resources
    │   └── META-INF
    │       └── MANIFEST.MF
    └── test
        ├── java
        └── resources

10 directories, 5 files

//To compile and generate jar
mvn install

//Remove output directory if exists
dfs -rm -r  /projects/wordcount/output


//Confirm the input files are stored.
hdfs dfs -ls /projects/wordcount/input

//Run jar
hadoop jar target/MyFirstMavennArtifactID-0.0.1-SNAPSHOT.jar WordCount
/projects/wordcount/input /projects/wordcount/output

//Download locally the output
```

hdfs dfs -get /projects/wordcount/output .

//Extract and verify the output. Here program was configured to generate compressed output.
bash-3.2$ gunzip part-r-00000.gz
bash-3.2$ ls
_SUCCESS part-r-00000
bash-3.2$ cat part-r-00000
Again. 1
Fourth 4
Hadoop 2
Hadoop. 1
Hello 13
Ita 2
Learning 1
MapReduce 1
World 1
another 1
count?? 2
enables 1
fifth 3
file 2
horizontal 1
line. 1
of 1
programming 1
scalling. 1
third 4
time. 11
with 1

hadoop version
Hadoop 2.8.2

Documented By:
Jigar M. Pandya
https://www.linkedin.com/in/jigar-pandya

Document Last updated: Aug 01, 2020.