

JAVA INTERFACE TO HDFS

In this lab, we will use Java to interact with HDFS. You will require **Eclipse IDE for Java EE**, which can be downloaded from:

<http://www.eclipse.org/downloads/eclipse-packages/>

The best strategy is to compile your code on your local computer and create a jar file. You will then upload this jar file to the Hadoop cluster and run it there.

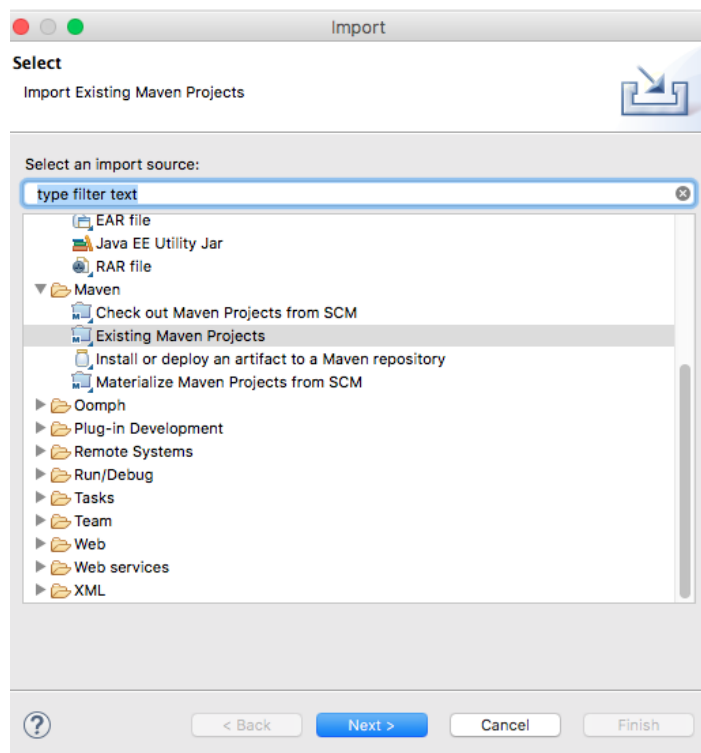
Steps:

1. You can download the sample Maven project from:

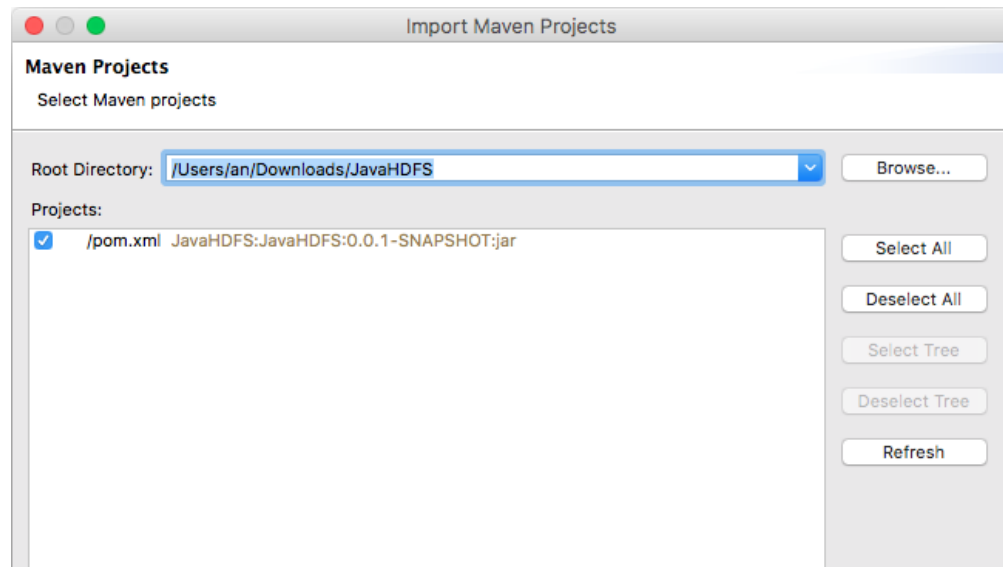
<http://www.utdallas.edu/~axn112530/cs6350/lab2/JavaHDFS.zip>

Unzip the file and remember the location.

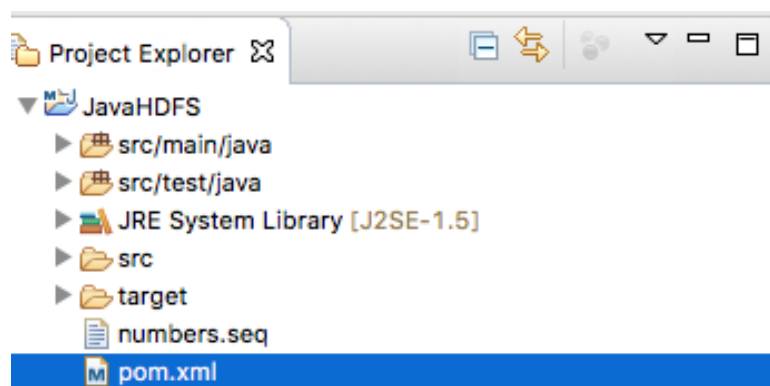
2. Import the project into Eclipse for Java EE. For this, go to File -> Import and choose "Existing Maven Projects"



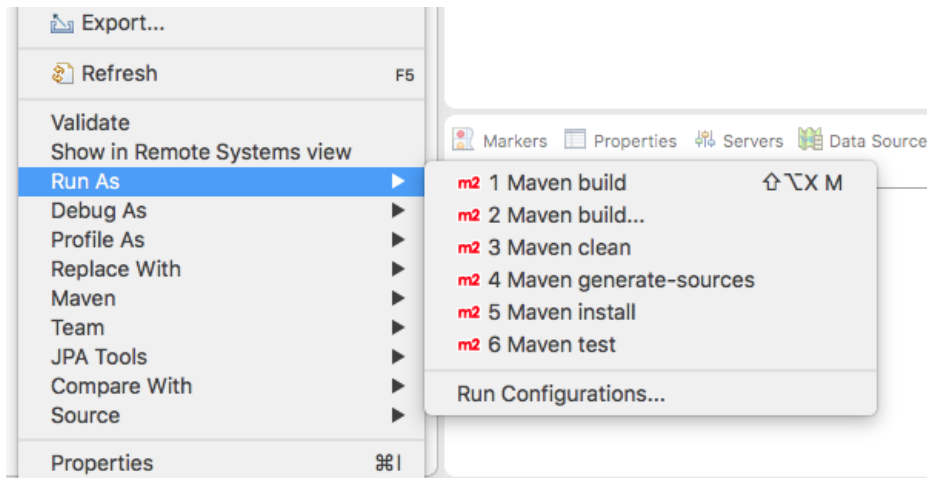
Hit next and navigate to the folder where you downloaded the JavaHDFS project in step 1. You should see a screen like below:



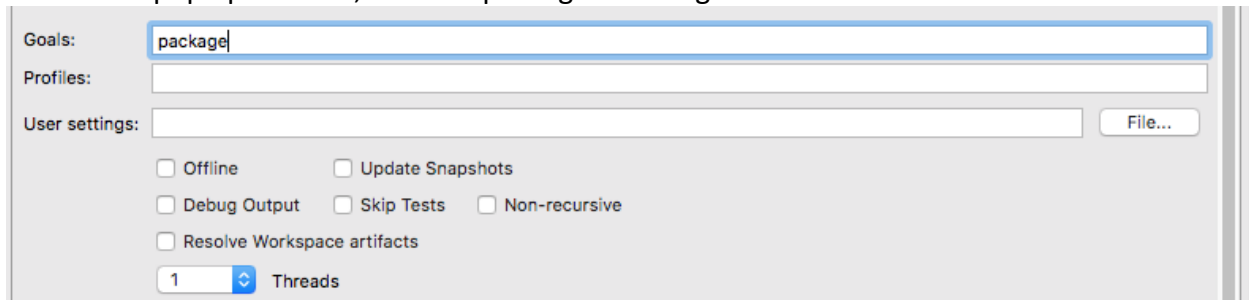
3. If you imported successfully, you should see the following in the Project Explorer window:



To build your project, right click on the pom.xml file and choose "Run As" -> "Maven build"

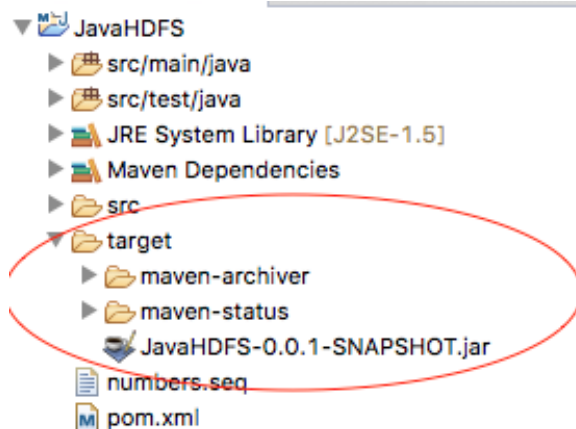


In the next pop up window, choose "package" as the goal:



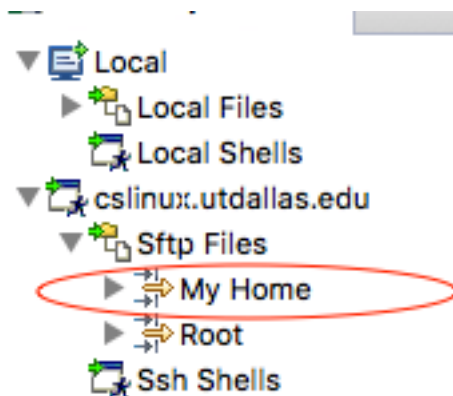
Click "Apply" and then "Run".

4. If it builds successfully, you should get a jar file in the **target** folder of your project.



5. Change to the **Remote System Explorer (RSE)** perspective by clicking Window -> Perspective -> Open Perspective -> Remote System Explorer.

6. Connect to the UTD cluster and then copy the jar file that you obtained in step 4 to the cluster. You can drag and drop (or right click and copy and then paste) the jar file to the "My Home" directory of the cluster.



7. Then, SSH into the cs6360.utdallas.edu node and run the command to execute the jar file. The command will be something like:

```
hadoop jar FileName.jar PackageName.ClassName arguments
```

where

FileName.jar is the name of your jar file

PackageName.ClassName is the full namespace and class name of the class you want to run
arguments is the arguments for your Java program

For example,

```
hadoop jar JavaHDFS-0.0.1-SNAPSHOT.jar JavaHDFS.JavaHDFS.FileCopyWithProgress  
mytest.txt hdfs://cshadoop1/user/axn112530/mytest.txt
```

Another example, if you want to display the contents of HDFS file

hdfs://cshadoop1/user/YourNetID/mytest.txt, then you would run the following command:

```
hadoop jar JavaHDFS-0.0.1-SNAPSHOT.jar JavaHDFS.JavaHDFS.URLCat  
hdfs://cshadoop1/user/YourNetID/mytest.txt
```