Spark Programming

Access Cluster at AHPCC

Access Cluster

- Open a web browser, on the address bar type in
  - hpc-portal2.hpc.uark.edu
  - On the popup window, enter your uark ID and password
- When you are on the dashboard page
  - Click on Clusters --> _Karpinski Shell Access to open the shell
  - Click on Files --> /karpinski/uarkID to open the FTP for uploading/downloading files
- Off-campus access to cluster
  - Install and run GlobalProtect vpn first
  - https://its.uark.edu/network-access/vpn/index.php

Allocate Cluster Node

- In the shell, type in spark-node4me.sh to allocate a cluster node for 2 hours to program.
- Once you finish using the cluster node, you can type in exit to release the node.
- The shell supports “Ctrl-V” for pasting. In the shell, when you select some texts, they are automatically copied.

Load Module

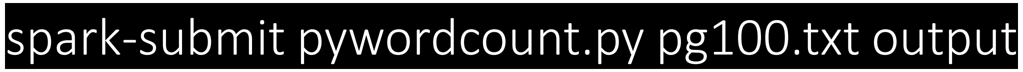
- Load modules
    - module load spark/2.3.0

- (Optional) Check available modules
    - module avail [java/python/spark/hadoop]

Submitting Python Applications

Submitting Python Applications

- Write your Python code and save as **.py
- Upload the Python code and input files to the cluster
- In shell type in spark-submit **.py [path_to_inputfile]



```
spark-submit pywordcount.py pg100.txt output
```

Submitting Java Applications

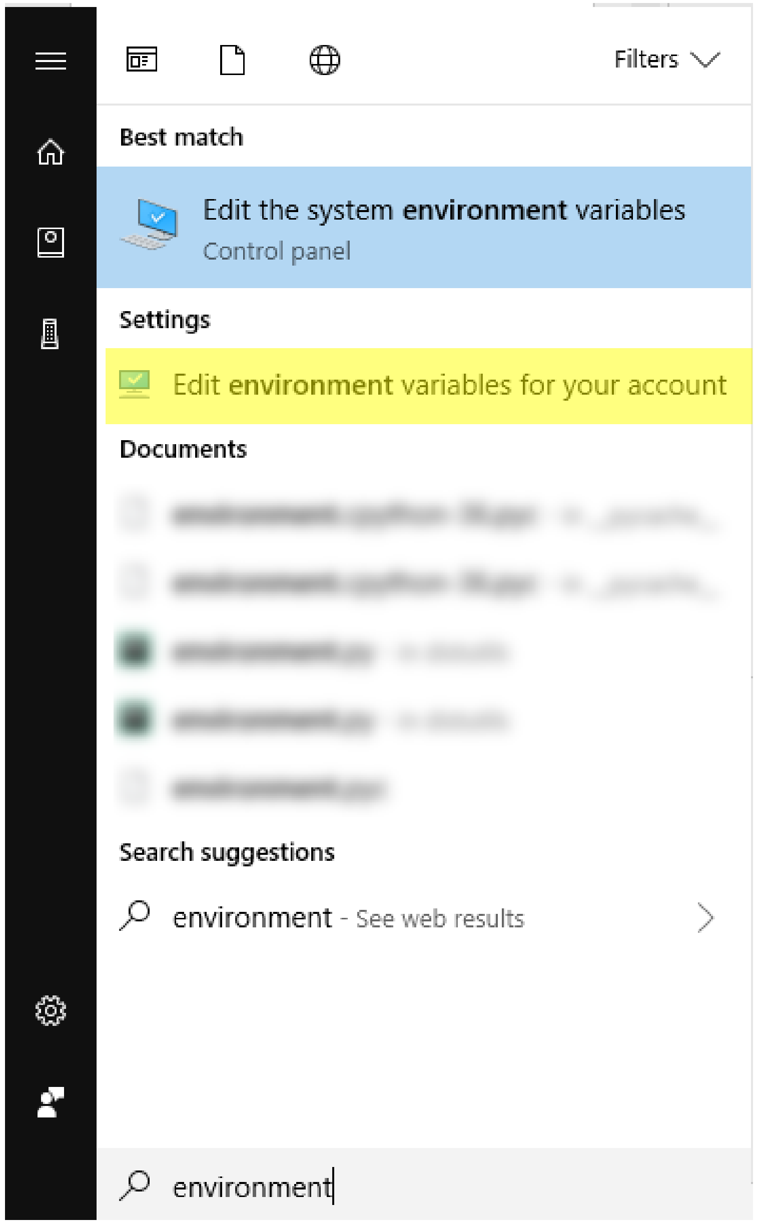
Overview

- Install/download Java, Spark, Eclipse, and Maven on your PC

1. Create Maven project in Eclipse
2. Write your java code in the project
3. Build Maven package
4. Upload the Maven package and input files to the cluster
5. Submit the application to spark in the shell

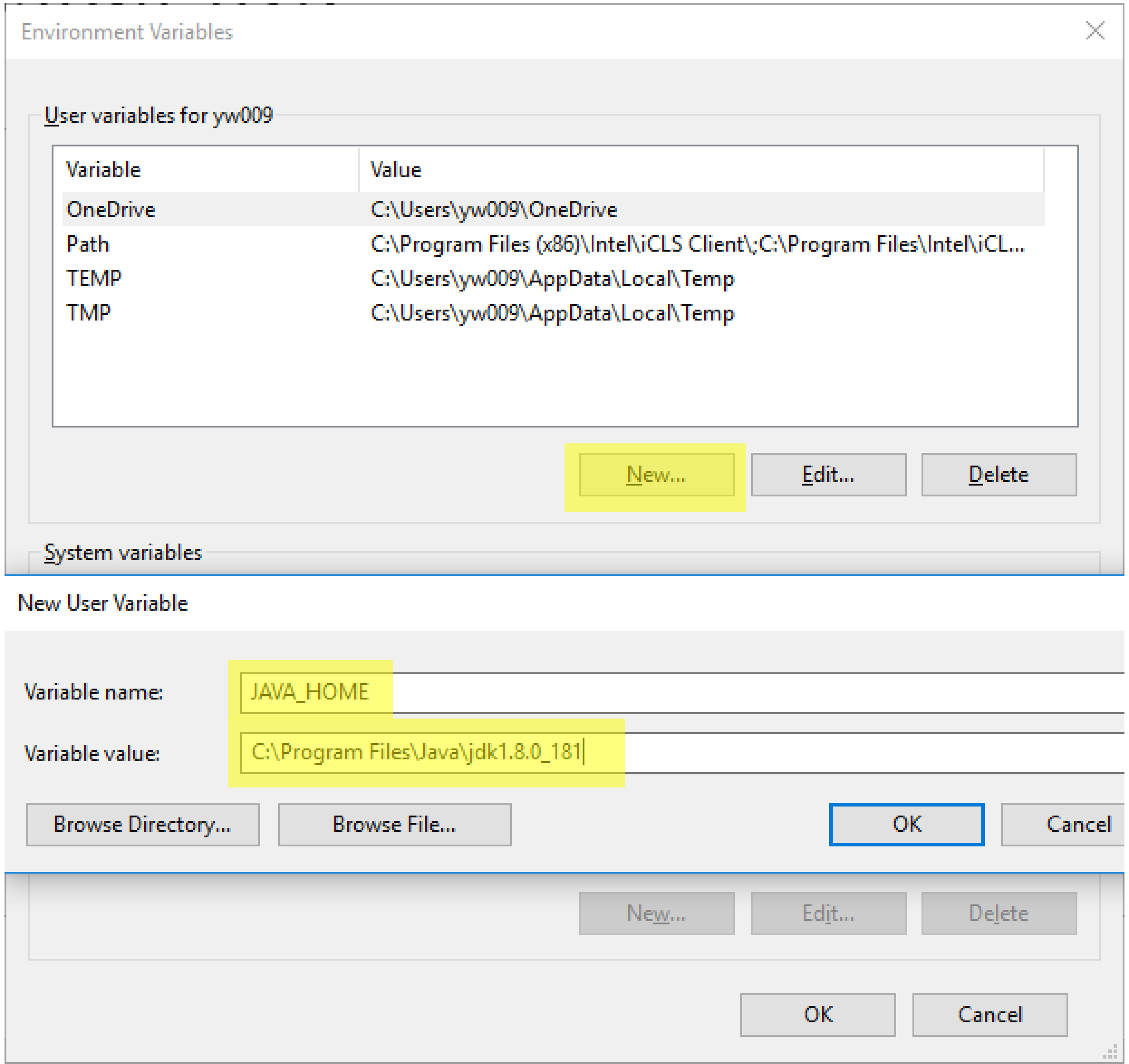
Steps of Set Environment Variables

- Open Environment Variables
  - Open Start and type “environment”,
  - Select “Edit environment variables for your account”. (see highlight of the figure)


Filters
Best match
Edit the system environment variables
Control panel
Settings
Edit environment variables for your account
Documents
Search suggestions
environment - See web results
environment

Steps of Set Environment Variables

- Set Java_HOME
  1. Click on the “New…” button
  2. In the Pop-up window, Type “JAVA_HOME” and your jdk path



Environment Variables

User variables for yw009

Variable	Value
OneDrive	C:\Users\yw009\OneDrive
Path	C:\Program Files (x86)\Intel\iCLS Client\;C:\Program Files\Intel\iCL...
TEMP	C:\Users\yw009\AppData\Local\Temp
TMP	C:\Users\yw009\AppData\Local\Temp

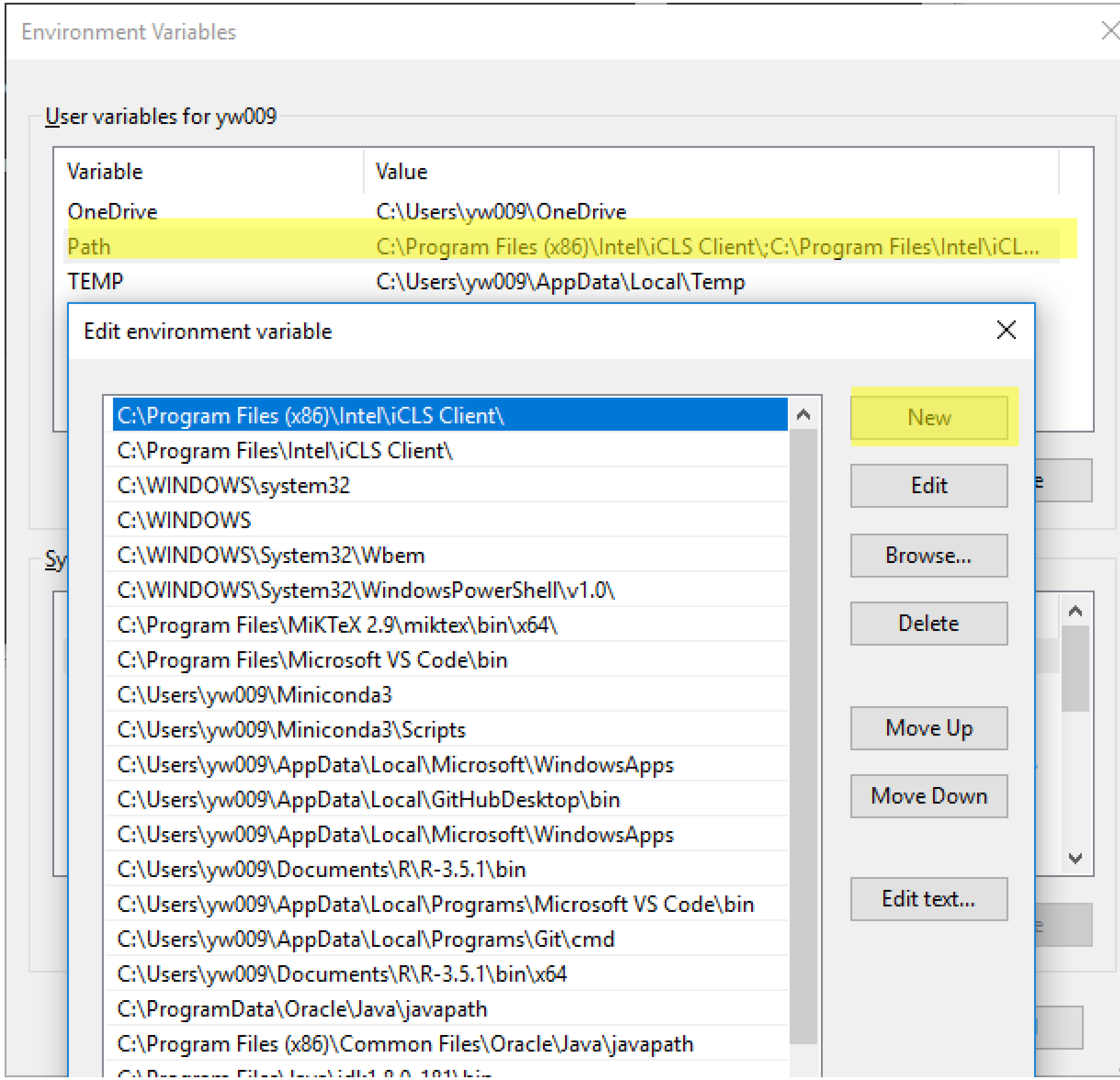
New User Variable

Variable name: JAVA_HOME

Variable value: C:\Program Files\Java\jdk1.8.0_181

Steps of Set Environment Variables

- Add to path
- Steps
  1. Double-click on the Path
  2. In the Pop-up window, click on the “New” button, type “%JAVA_HOME%\bin”



Environment Variables

User variables for yw009

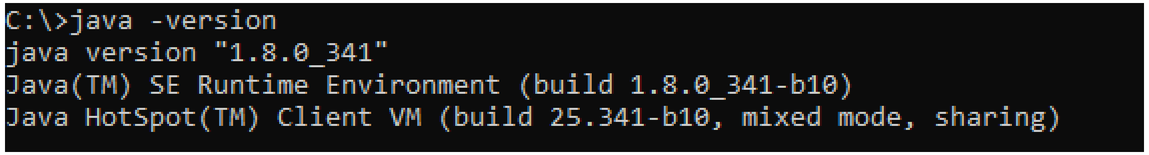
Variable	Value
OneDrive	C:\Users\yw009\OneDrive
Path	C:\Program Files (x86)\Intel\iCLS Client\;C:\Program Files\Intel\iCL...
TEMP	C:\Users\yw009\AppData\Local\Temp

Edit environment variable

C:\Program Files (x86)\Intel\iCLS Client\
C:\Program Files\Intel\iCLS Client\
C:\WINDOWS\system32
C:\WINDOWS
C:\WINDOWS\System32\Wbem
C:\WINDOWS\System32\WindowsPowerShell\v1.0\
C:\Program Files\MiKTeX 2.9\miktex\bin\x64\
C:\Program Files\Microsoft VS Code\bin
C:\Users\yw009\Miniconda3
C:\Users\yw009\Miniconda3\Scripts
C:\Users\yw009\AppData\Local\Microsoft\WindowsApps
C:\Users\yw009\AppData\Local\GitHubDesktop\bin
C:\Users\yw009\AppData\Local\Microsoft\WindowsApps
C:\Users\yw009\Documents\R\R-3.5.1\bin
C:\Users\yw009\AppData\Local\Programs\Microsoft VS Code\bin
C:\Users\yw009\AppData\Local\Programs\Git\cmd
C:\Users\yw009\Documents\R\R-3.5.1\bin\x64
C:\ProgramData\Oracle\Java\javapath
C:\Program Files (x86)\Common Files\Oracle\Java\javapath

Install Java 8

- Download Java 8 from the link: https://www.java.com/download/ie_manual.jsp and install it.
- Set environmental variables:
  - User variable:
    - Variable: JAVA_HOME;
    - Value: C:\Program Files\Java\jdk1.8.0_341
  - System variable:
    - Variable: PATH
    - Value: %JAVA_HOME%\bin
- Check on cmd, see below:



```
C:\>java -version
java version "1.8.0_341"
Java(TM) SE Runtime Environment (build 1.8.0_341-b10)
Java HotSpot(TM) Client VM (build 25.341-b10, mixed mode, sharing)
```

Download Spark 2.3.0

- Download it from the following link: https://archive.apache.org/dist/spark/ and extract it into a folder such as C:\spark
- Set environmental variables:
  - User variable:
    - Variable: SPARK_HOME;
    - Value: C:\spark\spark-2.3.0-bin-hadoop2.7
  - System variable:
    - Variable: PATH
    - Value: %SPARK_HOME%\bin

Install Eclipse

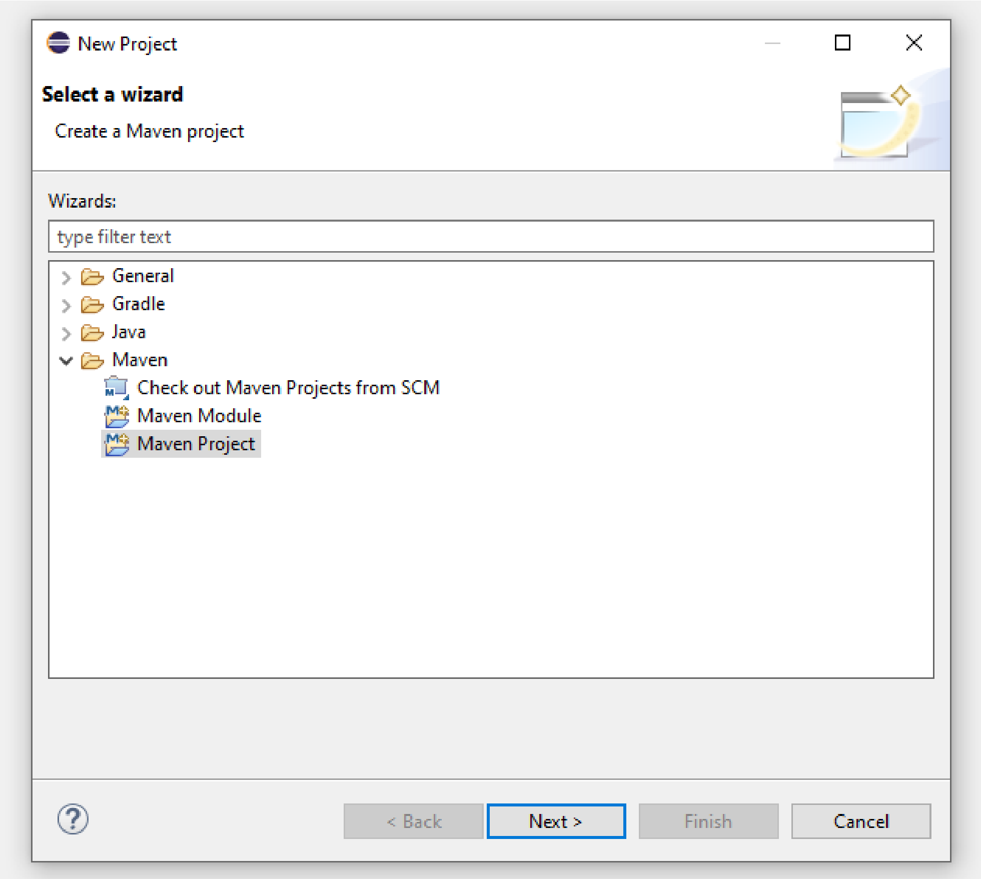
- Download it from the link: https://eclipse.org/downloads/ and install it for Java.

Download Maven 3.3

- Download Apache-Maven-3.3.9 from the link: http://apache.mivzakim.net/maven/maven-3/3.3.9/binaries/apache-maven-3.3.9-bin.zip and extract it into a folder such as C:\apache-maven-3.3.9
- Set environmental variables:
  - User variable:
    - Variable: MAVEN_HOME;
    - Value: C:\apache-maven-3.3.9
  - System variable:
    - Variable: PATH
    - Value: %MAVEN_HOME%\bin

Create First WordCount Project

- Open Eclipse and do File->New->project->Maven Project.

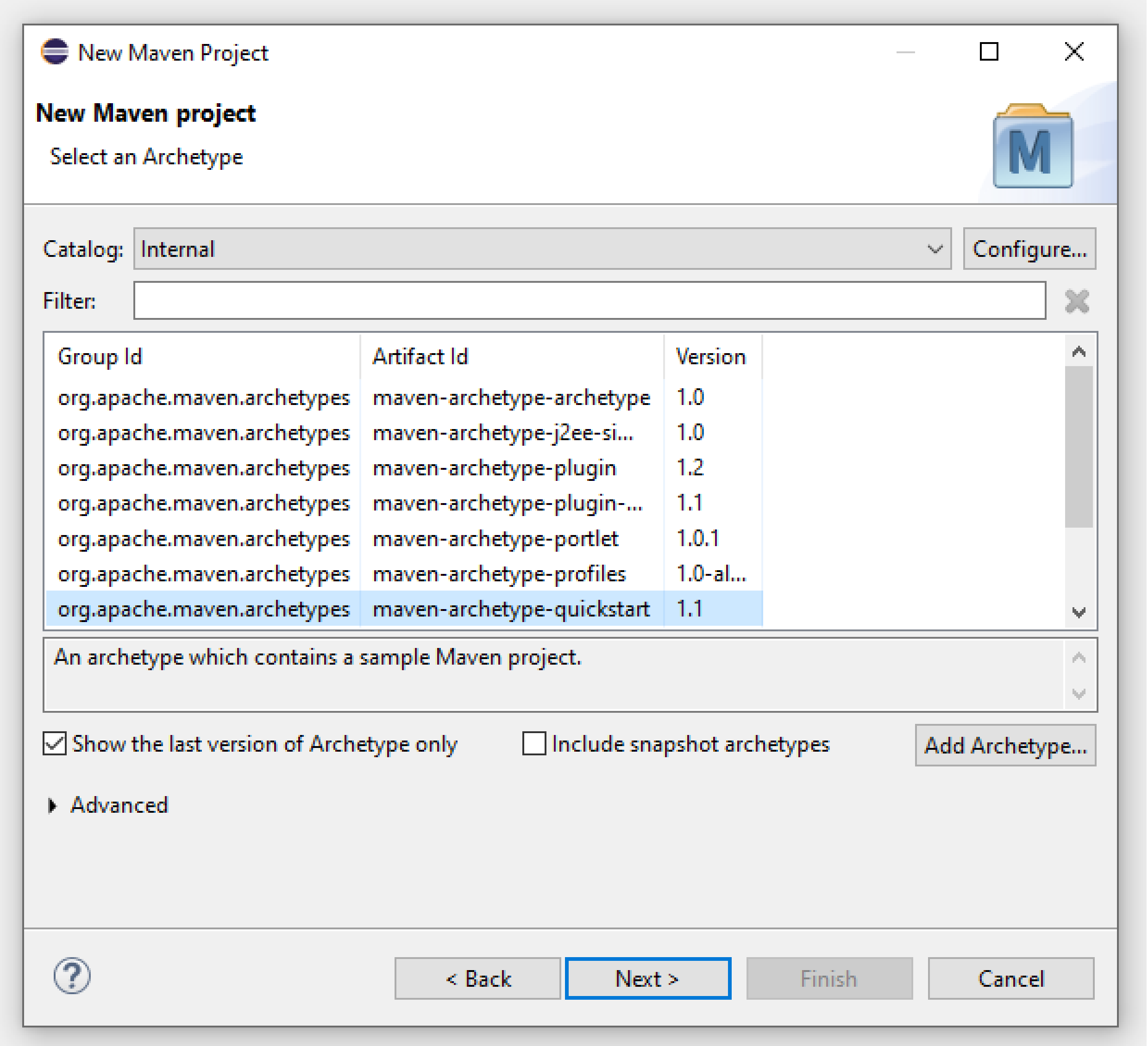


New Project

Select a wizard

Create a Maven project

Wizards: type filter text



New Maven Project

New Maven project

Select an Archetype

Catalog: Internal

Filter:

Group Id	Artifact Id	Version
org.apache.maven.archetypes	maven-archetype-archetype	1.0
org.apache.maven.archetypes	maven-archetype-j2ee-si...	1.0
org.apache.maven.archetypes	maven-archetype-plugin	1.2
org.apache.maven.archetypes	maven-archetype-plugin-...	1.1
org.apache.maven.archetypes	maven-archetype-portlet	1.0.1
org.apache.maven.archetypes	maven-archetype-profiles	1.0-al...
org.apache.maven.archetypes	maven-archetype-quickstart	1.1

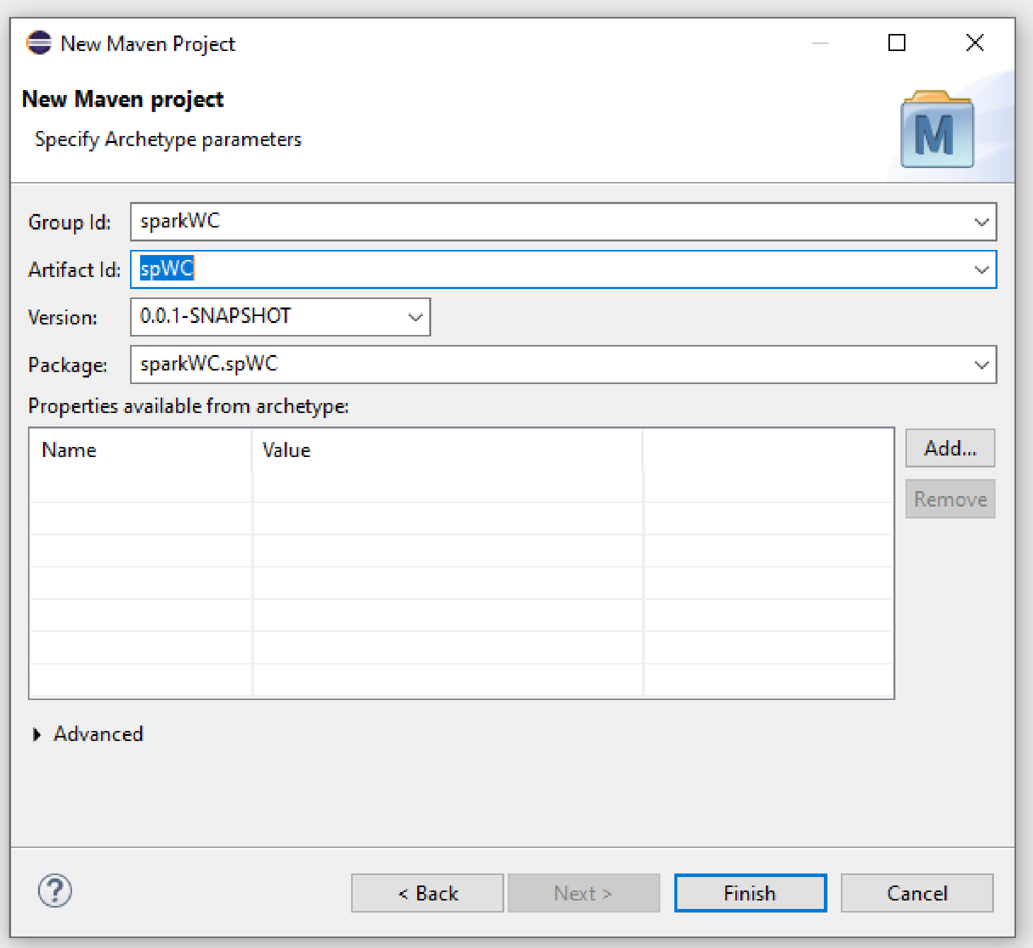
An archetype which contains a sample Maven project.

☑ Show the last version of Archetype only

□ Include snapshot archetypes

Create First WordCount Project

- Enter Group id, Artifact id, and click finish.



New Maven project

Specify Archetype parameters

Group Id: sparkWC

Artifact Id: spWC

Version: 0.0.1-SNAPSHOT

Package: sparkWC.spWC

Properties available from archetype:

Name	Value

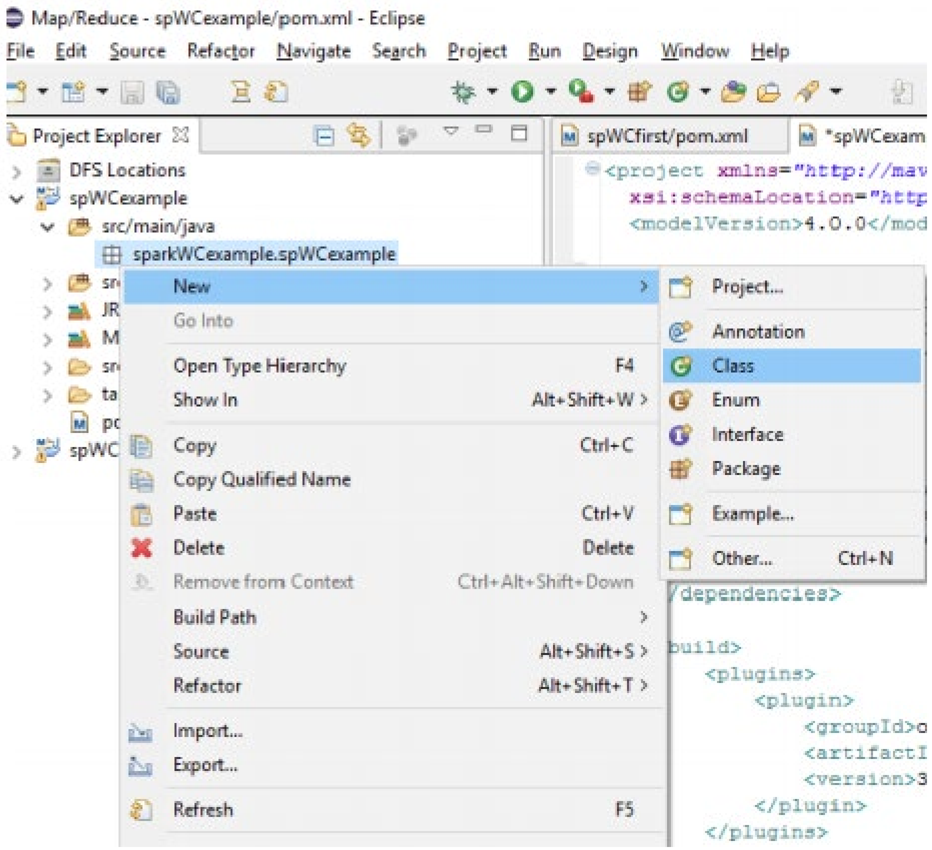
Create First WordCount Project

- Edit pom.xml. Change the <dependencies> part to the following.

```xml
<dependencies>
    <dependency>
        <groupId>org.apache.spark</groupId>
        <artifactId>spark-core_2.11</artifactId>
        <version>2.2.0</version>
        <scope>provided</scope>
    </dependency>
    <dependency>
        <groupId>org.apache.spark</groupId>
        <artifactId>spark-sql_2.11</artifactId>
        <version>2.3.0</version>
    </dependency>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>3.8.1</version>
        <scope>test</scope>
    </dependency>
</dependencies>
```

Create First WordCount Project

- Write your java code in class JavaWordCount



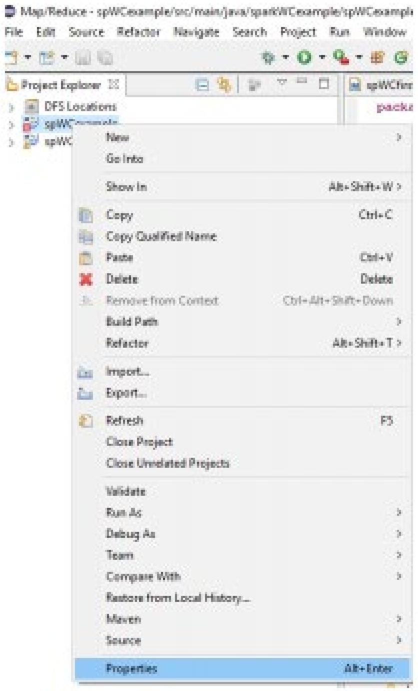
```
<project xmlns="http://mav
  xsi:schemaLocation="http
  <modelVersion>4.0.0</mod
```

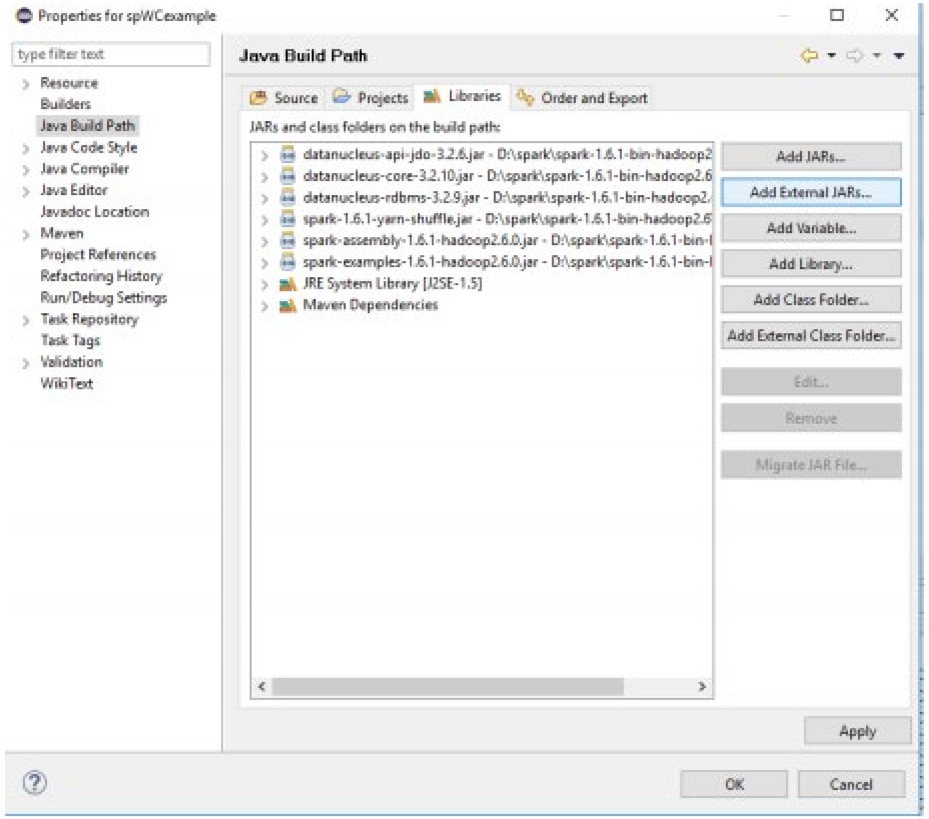
```
/dependencies>

build>
  <plugins>
    <plugin>
      <groupId>o
      <artifactI
      <version>3
    </plugin>
  </plugins>
```

Create First WordCount Project

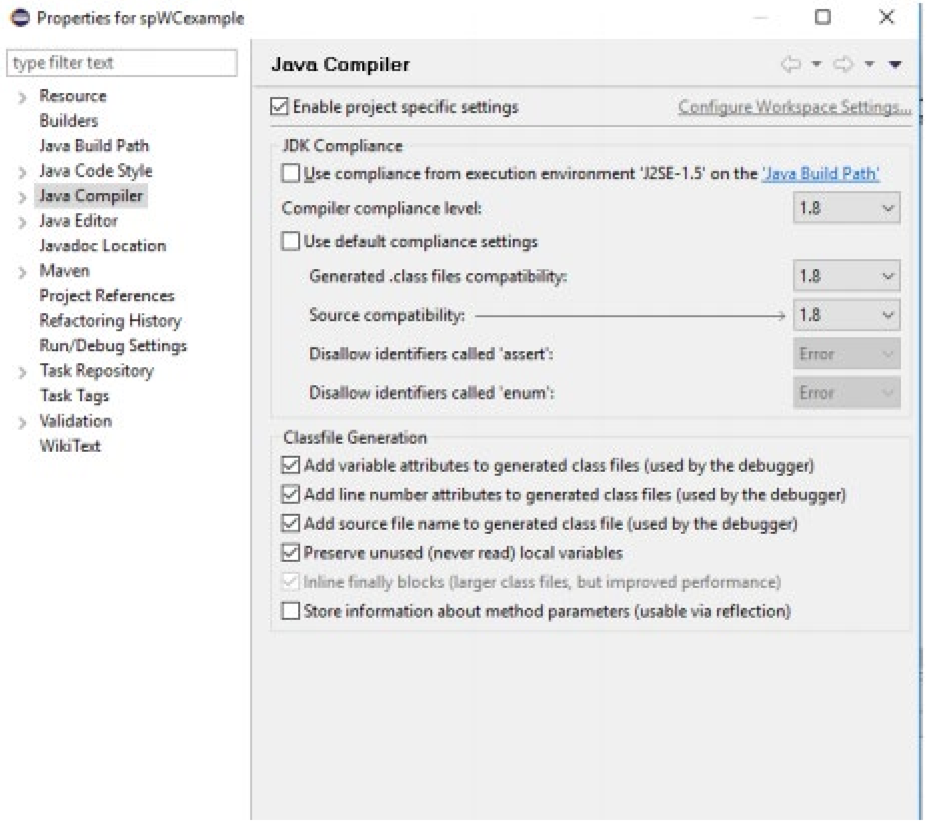
- Add external jar from the location C:\spark\spark-2.3.0-bin-hadoop2.7\jar and set Java 8 for compilation; see below.





JARs and class folders on the build path:

- datanucleus-api-jdo-3.2.6.jar - D:\spark\spark-1.6.1-bin-hadoop2
- datanucleus-core-3.2.10.jar - D:\spark\spark-1.6.1-bin-hadoop2.6
- datanucleus-rdbms-3.2.9.jar - D:\spark\spark-1.6.1-bin-hadoop2.
- spark-1.6.1-yarn-shuffle.jar - D:\spark\spark-1.6.1-bin-hadoop2.6
- spark-assembly-1.6.1-hadoop2.6.0.jar - D:\spark\spark-1.6.1-bin-
- spark-examples-1.6.1-hadoop2.6.0.jar - D:\spark\spark-1.6.1-bin-
- JRE System Library [J2SE-1.5]
- Maven Dependencies



☑ Enable project specific settings

JDK Compliance

☐ Use compliance from execution environment 'J2SE-1.5' on the 'Java Build Path'

Compiler compliance level: 1.8

☐ Use default compliance settings

Generated .class files compatibility: 1.8

Source compatibility: 1.8

Disallow identifiers called 'assert': Error

Disallow identifiers called 'enum': Error

Classfile Generation

☑ Add variable attributes to generated class files (used by the debugger)

☑ Add line number attributes to generated class files (used by the debugger)

☑ Add source file name to generated class file (used by the debugger)

☑ Preserve unused (never read) local variables

☑ Inline finally blocks (larger class files, but improved performance)

☐ Store information about method parameters (usable via reflection)

Create First WordCount Project

- Build the project: Open cmd, go to the following location (where we stored the project, e.g., C:\Users\eclipse-workspace\spWCexample):
  - Type in `mvn package`
  - Will build a Maven package, e.g., **spWCexample-0.0.1-SNAPSHOT.jar**
- Execute the project on the cluster:
  - Upload **spWCexample-0.0.1-SNAPSHOT.jar** and input files to the cluster
  - Open the shell, type in

  ```
  spark-submit --class sparkWCexample.spWCexample.JavaWordCount --master local[2] ./spWCexample-0.0.1-SNAPSHOT.jar ./input.txt ./output
  ```