Using ToolRunner and Passing Parameters

|  |  |
| --- | --- |
| **Exercise Dir** | ~/workspace/toolrunner |
| **Eclipse Proj** | toolrunner |
| **Java Files** | AverageReducer.java  (Reducer from AverageWordLength)  LetterMapper.java  (Mapper from AverageWordLength)  AvgWordLength.java  (driver from AverageWordLength) |
| **JAR File** | toolrunner.jar |

# 

**In this Exercise, you will implement a driver using ToolRunner.**

Follow the steps below to start with the AverageWordLength program you wrote in an earlier exercise, and modify the driver to use ToolRunner. Then modify the Mapper to reference a Boolean parameter called caseSensitive; if true, the mapper should treat upper and lower case letters as different; if false or unset, all letters should be converted to lower case.

**Modify the Average Word Length Driver to use Toolrunner**

Copy the Reducer, Mapper and driver code you completed in the “Writing Java

MapReduce Programs” exercise earlier, in the averagewordlength project. (If you

did not complete the exercise, use the code from the solution package.)

**Copying Source Files**

You can use Eclipse to copy a Java source file from one project or package to

another by right-clicking on the file and selecting Copy, then right-clicking the new package and selecting Paste. If the packages have different names (e.g. if you copy from averagewordlength.solution to toolrunner.fixme), Eclipse will automatically change the package directive at the top of the file. If you copy the file using a file browser or the shell, you will have to do that manually.

Modify the AvgWordLength driver to use ToolRunner. Refer to the slides for details.

Implement the run method

Modify main to call run

Jar your solution and test it before continuing; it should continue to function exactly as it did before. Refer to the *Writing a Java MapReduce Program* exercise for how to assemble and test if you need a reminder.

**Modify the Mapper to use a configuration parameter**

Modify the LetterMapper class to:

Override the setup method to get the value of a configuration parameter called caseSensitive, and use it to set a member variable indicating whether to do case sensitive or case insensitive processing.

In the map method, choose whether to do case sensitive processing (leave the letters as-‐is), or insensitive processing (convert all letters to lower-‐case) based on that variable.

**Pass a parameter programmatically**

Modify the driver’s run method to set a Boolean configuration parameter called caseSensitive. (Hint: use the Configuration.setBoolean method.)

Test your code twice, once passing false and once passing true. When set to true, your final output should have both upper and lower case letters; when false, it should have only lower case letters.

Hint: Remember to rebuild your Jar file to test changes to your code.

**Pass a parameter as a runtime parameter**

Comment out the code that sets the parameter programmatically.

HINT: Select the code to comment and then select Source > Toggle Comment.

Test again, this time passing the parameter value using –D on the Hadoop command line, e.g.:

$ yarn jar toolrunner.jar fixme.AvgWordLength

**-DcaseSensitive=true** shakespeare toolrunnerout

Test passing both true and false to confirm the parameter works correctly.

**END**