**Counters Example**

**-----------------**

import java.io.IOException;

import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class **WordCountWithCounter** {

static enum WordsNature { STARTS\_WITH\_DIGIT, STARTS\_WITH\_LETTER, ALL }

/\*\*

\* The map class of WordCount.

\*/

public static class TokenCounterMapper

extends Mapper<Object, Text, Text, IntWritable> {

private final static IntWritable one = new IntWritable(1);

private Text word = new Text();

public void map(Object key, Text value, Context context)

throws IOException, InterruptedException {

StringTokenizer itr = new StringTokenizer(value.toString());

while (itr.hasMoreTokens()) {

word.set(itr.nextToken());

context.write(word, one);

}

}

}

/\*\*

\* The reducer class of WordCount

\*/

public static class TokenCounterReducer

extends Reducer<Text, IntWritable, Text, IntWritable> {

public void reduce(Text key, Iterable<IntWritable> values, Context context)

throws IOException, InterruptedException {

int sum = 0;

String token = key.toString();

if( StringUtils.startsWithDigit(token) ){

context.getCounter(WordsNature.STARTS\_WITH\_DIGIT).increment(1);

}

else if( StringUtils.startsWithLetter(token) ){

context.getCounter(WordsNature.STARTS\_WITH\_LETTER).increment(1);

}

context.getCounter(WordsNature.ALL).increment(1);

for (IntWritable value : values) {

sum += value.get();

}

context.write(key, new IntWritable(sum));

}

}

/\*\*

\* The main entry point.

\*/

public static void main(String[] args) throws Exception {

Configuration conf = new Configuration();

Job job = new Job(conf, "Example Hadoop 0.20.1 WordCount");

//job.setJarByClass(WordCountOld.class);

job.setJarByClass(WordCountWithCounter.class);

job.setMapperClass(TokenCounterMapper.class);

job.setReducerClass(TokenCounterReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(job, new Path(args[0]));

FileOutputFormat.setOutputPath(job, new Path(args[1]));

System.exit(job.waitForCompletion(true) ? 0 : 1);

}

}

public class **StringUtils** {

public static boolean startsWithDigit(String s){

if( s == null || s.length() == 0 )

return false;

return Character.isDigit(s.charAt(0));

}

public static boolean startsWithLetter(String s){

if( s == null || s.length() == 0 )

return false;

return Character.isLetter(s.charAt(0));

}

}