What is Counter in MapReduce?

A counter in MapReduce is a mechanism used for collecting statistical information about the MapReduce job. This information could be useful for diagnosis of a problem in MapReduce job processing. Counters are similar to putting a log message in the code for a map or reduce.

Typically, these counters are defined in a program (map or reduce) and are incremented during execution when a particular event or condition (specific to that counter) occurs. A very good application of counters is to track valid and invalid records from an input dataset.

Types of MapReduce Counters

There are basically 2 types of MapReduce Counters

1. **Hadoop Built-In counters:**There are some built-in counters which exist per job. Below are built-in counter groups-
    - **MapReduce Task Counters** - Collects task specific information (e.g., number of input records) during its execution time.
    - **FileSystem Counters** - Collects information like number of bytes read or written by a task
    - **FileInputFormat Counters** - Collects information of a number of bytes read through FileInputFormat
    - **FileOutputFormat Counters** - Collects information of a number of bytes written through FileOutputFormat
    - **Job Counters -** These counters are used by JobTracker. Statistics collected by them include e.g., the number of task launched for a job.
2. **User Defined Counters**

In addition to built-in counters, a user can define his own counters using similar functionalities provided by programming languages. For example, in Java 'enum' are used to define user defined counters.

Counters Example

An example MapClass with Counters to count the number of missing and invalid values. Input data file used in this tutorial Our input data set is a CSV file, SalesJan2009.csv

```
public static class MapClass
            extends MapReduceBase
            implements Mapper<LongWritable, Text, Text, Text>
{
    static enum SalesCounters { MISSING, INVALID };
    public void map ( LongWritable key, Text value,
                 OutputCollector<Text, Text> output,
                 Reporter reporter) throws IOException
    {

        //Input string is split using ',' and stored in 'fields' array
        String fields[] = value.toString().split(",", -20);
        //Value at 4th index is country. It is stored in 'country' variable
        String country = fields[4];

        //Value at 8th index is sales data. It is stored in 'sales' variable
        String sales = fields[8];

        if (country.length() == 0) {
            reporter.incrCounter(SalesCounters.MISSING, 1);
        } else if (sales.startsWith("\"")) {
            reporter.incrCounter(SalesCounters.INVALID, 1);
        } else {
            output.collect(new Text(country), new Text(sales + ",1"));
        }
    }
}
```

Above code snippet shows an example implementation of counters in Map Reduce.

Here, **SalesCounters** is a counter defined using **'enum'**. It is used to count **MISSING** and **INVALID** input records.

In the code snippet, if **'country'** field has zero length then its value is missing and hence corresponding counter **SalesCounters.MISSING** is incremented.

Next, if **'sales'** field starts with a **"** then the record is considered INVALID. This is indicated by incrementing counter **SalesCounters.INVALID.**