**Explain in brief**

**● Sequence File Format**

**● NLine Input Format**

**● DB Input Format**

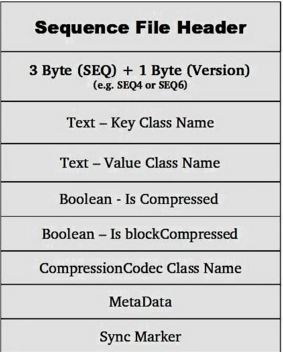
**● DB Output Format**

**Sequence File Format:**

[SequenceFile](http://hadoop.apache.org/core/docs/current/api/org/apache/hadoop/io/SequenceFile.html) is a flat file consisting of binary key/value pairs. It is extensively used in [MapReduce](https://wiki.apache.org/hadoop/MapReduce) as input/output formats. It is also worth noting that, internally, the temporary outputs of maps are stored using SequenceFile.

Based on the compression Sequence file formats are classified into three types. They are:

1. Uncompressed key/value records.
2. Record compressed key/value records - only 'values' are compressed here.
3. Block compressed key/value records - both keys and values are collected in 'blocks' separately and compressed. The size of the 'block' is configurable.

****

A sequence file consists of a header followed by one or more methods. All the three formats uses the same header structure and is shown below.

First 3 bytes of a sequence file are “SEQ”, which denotes that the file is sequence file and followed by a 1 byte representing the actual version number (Eg: SEQ4 or SEQ6).

Sync marker denotes the end of the header. The Sync marker permits seeking to a random point in a file, which is required to be able to efficiently split large files for parallel processing by Mapreduce.

Metadata is secondary key-value list that can be written during the initialization of sequence file writer.

**NLine Input Format:**

NLineInputFormat which splits N lines of input as one split. In many "pleasantly" parallel applications, each process/mapper processes the same input file, but with computations are controlled by different parameters.

With NLineInputFormat each mapper receives fixed number of lines of input, unlike TextInputFormat and KeyValueTextInputFormat. The number of lines of input to each mapper can be controlled by setting the property, mapreduce.input.lineinputformat.linespermap in API. The default value is 1.

Example:

conf.setInt(NLineInputFormat.LINES\_PER\_MAP, 1000); // sets N value to 1000

NLineInputFormat is used in applications that take a small amount of input data and run an extensive (that is, CPU-intensive) computation for it, then emit their output.

And you can create a “seed” input file that lists the data sources, one per line. Then each mapper is allocated a single data source, and it loads the data from that source into HDFS.

**DB Input Format:**

The DBInputFormat is an InputFormat class that allows you to read data from a database. An InputFormat is Hadoop’s formalization of a data source it can mean files formatted in a particular way, data read from a database, etc. DBInputFormat provides a simple method of scanning entire tables from a database, as well as the means to read from arbitrary SQL queries performed against the database.

To use the DBInputFormat, you’ll need to configure your job. The following example shows how to connect to a MySQL database and load from a table:

CREATE TABLE employees ( employee\_id INTEGER NOT NULL PRIMARY KEY, name VARCHAR(32) NOT NULL);

JobConf conf = new JobConf(getConf(), MyDriver.class); conf.setInputFormat(DBInputFormat.class);

DBConfiguration.configureDB(conf,“com.mysql.jdbc.Driver”,“jdbc:mysql://localhost/mydatabase”);

String [] fields = { “employee\_id”, "name"};

DBInputFormat.setInput(conf, MyRecord.class, “employees”, null /\* conditions \*/, “employee\_id”, fields);

// set Mapper, etc., and call JobClient.runJob(conf);

**DB Output Format:**

A companion class, DBOutputFormat, will allow you to write results back to a database. When setting up the job,

conf.setOutputFormat(DBOutputFormat.class);

This method then defines how the results will be written back to the database. Its three arguments are the ***JobConf*** object for the job, a string defining the name of the table to write to, and an array of strings defining the fields of the table to populate.

DBOutputFormat.setOutput(job, "employees", "employee\_id", "name");