# InputFormat<K, V>

逻辑上负责切分输入文件（生成InputSplit列表，每个InputSplit对应到一个map任务），并生成用于读取切分后文件的RecordReader

抽象方法：

1. **public** **abstract** List<InputSplit> getSplits(JobContext context)

生成splits，The split is a logical split of the inputs and the input files are not physically split into chunks. For e.g. a split could be <input-file-path, start, offset> tuple. The InputFormat also creates the [RecordReader](eclipse-javadoc:%E2%98%82=hadoop/src%5C/mapred%3Corg.apache.hadoop.mapreduce%7BInputFormat.java%E2%98%83InputFormat~getSplits~QJobContext;%E2%98%82RecordReader) to read the [InputSplit](eclipse-javadoc:%E2%98%82=hadoop/src%5C/mapred%3Corg.apache.hadoop.mapreduce%7BInputFormat.java%E2%98%83InputFormat~getSplits~QJobContext;%E2%98%82InputSplit).

1. **public** **abstract** RecordReader<K,V> createRecordReader(InputSplit split, TaskAttemptContext context)

为每个split创建record reader，The framework will call [RecordReader.initialize(InputSplit, TaskAttemptContext)](eclipse-javadoc:%E2%98%82=hadoop/src%5C/mapred%3Corg.apache.hadoop.mapreduce%7BInputFormat.java%E2%98%83InputFormat~createRecordReader~QInputSplit;~QTaskAttemptContext;%E2%98%82RecordReader%E2%98%82initialize%E2%98%82InputSplit%E2%98%82TaskAttemptContext) before the split is used.

## FileInputFormat

FileInputFormat is the base class for all file-based InputFormats. This provides a generic implementation of [getSplits(JobContext)](eclipse-javadoc:%E2%98%82=hadoop/src%5C/mapred%3Corg.apache.hadoop.mapreduce.lib.input%7BFileInputFormat.java%E2%98%83FileInputFormat%E2%98%82%E2%98%82getSplits%E2%98%82JobContext). Subclasses of FileInputFormat can also override the [isSplitable(JobContext, Path)](eclipse-javadoc:%E2%98%82=hadoop/src%5C/mapred%3Corg.apache.hadoop.mapreduce.lib.input%7BFileInputFormat.java%E2%98%83FileInputFormat%E2%98%82%E2%98%82isSplitable%E2%98%82JobContext%E2%98%82Path) method to ensure input-files are not split-up and are processed as a whole by [Mapper](eclipse-javadoc:%E2%98%82=hadoop/src%5C/mapred%3Corg.apache.hadoop.mapreduce.lib.input%7BFileInputFormat.java%E2%98%83FileInputFormat%E2%98%82Mapper)s.

方法：

1. **public** List<InputSplit> getSplits(JobContext job)
   1. 获取split的上下界：minSize和maxSize
   2. 获取所有指定的文件List<FileStatus>files = listStatus(job);并遍历之，对每个FileStatus file：
      1. 查询文件的block列表：BlockLocation[] blkLocations = fs.getFileBlockLocations(file, 0, length);
      2. 如果file需要进行split（isSplitable返回true）：
         1. 计算**long** splitSize = computeSplitSize(blockSize, minSize, maxSize);（blockSize为文件的block大小，一般为64M）
         2. 循环计算bytesRemaining/splitSize > *SPLIT\_SLOP*,对每个splitSize大小的数据，建立一个FileSplit（因为使用1.1，所以最后一个split可能大于FileSplit，而这个block的splitHosts，即block所在的host，可能只包含一部分，eg：65M）
      3. 如果isSplitable返回false，即不需要继续切分，则直接返回整个文件的FileSplit

### TextInputFormat

继承自FileInputFormat<LongWritable, Text>，键是文件中的偏移（行首的偏移），值是行的内容

只有一个方法：

**public** RecordReader<LongWritable, Text>createRecordReader(InputSplit split, TaskAttemptContext context)

直接返回**new** LineRecordReader(job, (FileSplit) genericSplit);

### KeyValueTextInputFormat

继承自FileInputFormat<Text, Text>，键值对以分隔符切割

只有一个方法：

**public** RecordReader<LongWritable, Text>createRecordReader(InputSplit split, TaskAttemptContext context)

直接返回**new** KeyValueLineRecordReader(job, (FileSplit) genericSplit);

### NLineInputFormat

继承自FileInputFormat，splits N lines of input as one split.

覆盖了父类的getSplits方法，对每个文件调用*getSplitsForFile*方法按行计算split

方法：

1. **public** RecordReader<LongWritable, Text> createRecordReader(InputSplit genericSplit, TaskAttemptContext context)

返回**new** LineRecordReader(job, (FileSplit) genericSplit);（和TextInputFormat一样）

1. **public** **static** List<FileSplit> getSplitsForFile(FileStatus status, Configuration conf, **int** numLinesPerSplit)

使用LineReader，每读取numLinesPerSplit行就新建一个FileSplit

# InputSplit

InputSplit represents the data to be processed by an individual [Mapper](eclipse-javadoc:%E2%98%82=hadoop/src%5C/mapred%3Corg.apache.hadoop.mapreduce%7BInputSplit.java%E2%98%83InputSplit%E2%98%82Mapper).

Typically, it presents a byte-oriented view on the input and is the responsibility of [RecordReader](eclipse-javadoc:%E2%98%82=hadoop/src%5C/mapred%3Corg.apache.hadoop.mapreduce%7BInputSplit.java%E2%98%83InputSplit%E2%98%82RecordReader) of the job to process this and present a record-oriented view.

抽象方法：

1. **public** **abstract** **long** getLength()

Get the size of the split, so that the input splits can be sorted by size.

1. **public** **abstract** String[] getLocations()

Get the list of nodes by name where the data for the split would be local.

## FileSplit

继承自InputSplit，表示A section of an input file.

属性：

1. **private** Path file;
2. **private** **long** start;
3. **private** **long** length;
4. **private** String[] hosts;

包含block的DN列表

# FileStatus

/\*\* Interface that represents the client side information for a file.

\*/

属性：

1. **private** Path path;
2. **private** **long** length;
3. **private** **boolean** isdir;
4. **private** **short** block\_replication;
5. **private** **long** blocksize;
6. **private** **long** modification\_time;
7. **private** **long** access\_time;
8. **private** FsPermission permission;
9. **private** String owner;
10. **private** String group;

# TaskCompletionEvent

TaskCompletionEvent用于在客户端和JT之间传递task的消息（每个job对应多个task，使用RunningJob的.getTaskCompletionEvents方法），每个TaskCompletionEvent对应一个编号：eventId，在job整个job中自增

属性：

1. **private** **int** eventId;
2. **private** String taskTrackerHttp;
3. **private** **int** taskRunTime; // using int since runtime is the time difference
4. **private** TaskAttemptID taskId;
5. Status status;
6. **boolean** isMap = **false**;
7. **private** **int** idWithinJob;

# ID

这里比较混乱，新旧两个MR包交叉继承

## JobID

eg: job\_200707121733\_0003

An example JobID is : job\_200707121733\_0003 , which represents the third job running at the jobtracker started at 200707121733.

## JVMId

jvm开头

属性：

**boolean** isMap;

JobID jobId;

## TaskID

eg：task\_200707121733\_0003\_m\_000005

An example TaskID is : task\_200707121733\_0003\_m\_000005 , which represents the fifth map task in the third job running at the jobtracker started at 200707121733.

## TaskAttemptID

eg：attempt\_200707121733\_0003\_m\_000005\_0

An example TaskAttemptID is : attempt\_200707121733\_0003\_m\_000005\_0 , which represents the zeroth task attempt for the fifth map task in the third job running at the jobtracker started at 200707121733.