Line protocol handler

The Line protocol handler parses simple text lines.

Important: Each line must end with a newline! Especially important when not using a file for input but e.g. TCPServer transport handler or AMQP /RabbitMQ transport handler.

Options

- **delay**: Delay of reading in milliseconds (default 0). DEPRECATED: use `scheduler.delay` instead if not used together with delayeach
- **nanodelay**: Delay of reading in nanoseconds (default 0).
- **delayeach**: The number of lines between a delay is used (default 0).
- **checkdelay**: Delay between reading attempts when the transport stream isn't ready. Useful for polling log files which are currently written to reduce I/O overhead (default 0).
- **readfirstline**: Should the first line of the file be ignored (e.g. because of header information) (default: true), Remark: When reading from File transport handler and using directory, this will only treat the first line of the first file.
- **skipFirstLines**: The number of lines that should skipped. Use this, if one or more lines should be skipped (e.g. in case of a longer header).
- **debug**: If set to true, some additional thinks are available: (default false)
    - **dumpEachLine**: Dumps lines to the console. if set to 1 each line will be dumped
    - **measureEachLine**: Measures the processing time between n elements that are dumped
    - **maxLines**: Stop processing after n elements are read

Example

PQL

Line Protocol Handler

```
input = ACCESS({source='Line', wrapper='GenericPush',
transport='File',protocol='Line',
  dataHandler='Tuple',options=[['filename','test.csv'],['delay','100']],
  schema=[
    ['line','String'] ]
})
```

CQL