SVM protocol handler

Parses data in the LibSVM Format.

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).
- **readfirstline**: Should the first line of the file be ignored (e.g. because of header information) (default: true)
- **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
- **lastLine/maxLines**: Stop processing after n elements are read
- **svm.delimiter**: The delimiter for splitting the input (Default:).
- **svm.textDelimiter**: The delimiter for strings (Default: ').
- **addlinenumber**: Adds the line number (starting with 0) to the beginning of the line. Remember to add a proper attribute to the schema! (default false).
- **svm.floatingFormatter**:If used for writing, each double/float value will be formatted using this formatter (default null).
- **svm.numberFormatter**: If used for writing, each number other than double/float value will be formatted using this formatter (default null).
- **svm.trim**: Removes leading and trailing whitespaces in each element (default false)

Example

PQL

CSV Protocol Handler

```
input = ACCESS({source='SVM', wrapper='GenericPush',
transport='File',protocol='SVM',
  dataHandler='Tuple',options=[['delimiter',' '],['textDelimiter',"'"],['readfirstline','true'],
['delay','100']],
  schema=[
    ['symbol','String'],
    ['points','Double'] ]
})
```

CQL

CSV Protocol Handler

```
CREATE STREAM svm (symbol String, points Double)
    WRAPPER 'GenericPush'
    PROTOCOL 'SVM'
    TRANSPORT 'File'
    DATAHANDLER 'Tuple'
    OPTIONS ( delimiter' ' ', 'textDelimiter' "'", 'readfirstline' 'true', 'delay' '100')
```