

**NAME**

Lintian::SlidingWindow – Lintian interface to sliding window match

**SYNOPSIS**

```
use Lintian::SlidingWindow;

my $sfd = Lintian::SlidingWindow->new('<', 'someevilfile.c', sub { $_[0] = lc($_); });
my $window;
while ($window = $sfd->readwindow) {
    if (index($window, 'evil') > -1) {
        if ($window =~
            m/software \s++ shall \s++
            be \s++ used \s++ for \s++ good \s*+ ,?+ \s*+
            not \s++ evil/xsim) {
            # do something like : tag 'license-problem-json-evil';
        }
    }
}
```

**DESCRIPTION**

Lintian::SlidingWindow provides a way of matching some pattern, including multi line pattern, without needing to fully load the file in memory.

**CLASS METHODS**

`new(HANDLE[, BLOCKSUB[, BLOCKSIZE]])`

Create a new sliding window by reading from a given HANDLE, which must be open for reading. Optionally run BLOCKSUB against each block. Note that BLOCKSUB should apply transform byte by byte and does not depend of context.

Each window consists of up to two blocks of BLOCKSIZE characters.

**INSTANCE METHODS**

`readwindow`

Return a new block of sliding window. Return undef at end of file.

`blocknumber`

return the number of block read by the instance. Return undef if no block has been read.

**DIAGNOSTICS**

no data type specified

**AUTHOR**

Originally written by Bastien ROUCARIES for Lintian.

**SEE ALSO**

**lintian** (1)