

## Ruby master - Bug #15910

### \$. doesn't hold the linenumber anymore when reading a CSV-file using the CSV class

06/10/2019 11:15 AM - rovf (Ronald Fischer)

<b>Status:</b> Rejected	
<b>Priority:</b> Normal	
<b>Assignee:</b> kou (Kouhei Sutou)	
<b>Target version:</b>	
<b>ruby -v:</b> ruby 2.6.3p62 (2019-04-16 revision 67580) [x86_64-cygwin]	<b>Backport:</b> 2.4: UNKNOWN, 2.5: UNKNOWN, 2.6: UNKNOWN
<b>Description</b>	
Example program:	
<pre>require 'csv' IFS=';' CSV_OPTIONS = { col_sep: IFS, external_encoding: Encoding::ISO_8859_1, internal_encoding: Encoding::UTF_8 } CSV.new(\$stdin, CSV_OPTIONS).each do    row    puts "::::line #{\$.} row=#{row}" end%</pre>	
With this program, I could dump a semicolon-delimited CSV file, with line numbers. Now the line number always displays as 0. Has the implementation of CSV changed?	
See also <a href="https://stackoverflow.com/questions/56524941/is-special-variable-gone-from-ruby">https://stackoverflow.com/questions/56524941/is-special-variable-gone-from-ruby</a>	

### History

#### #1 - 06/10/2019 08:40 PM - jeremyevans0 (Jeremy Evans)

- Assignee set to kou (Kouhei Sutou)

I can confirm the behavior change. I bisected it to <https://github.com/ruby/csv/commit/8505ff0900a4789dce0740ab23e92de1fb89196e>. I would not consider this change a bug, since I don't think the csv library was ever documented as treating \$. as the line number of the current CSV file. I would consider it relying on internal behavior. Restoring backwards compatibility in regards to \$. looks like it may hurt performance. Assigning to kou, csv maintainer, to make a decision.

#### #2 - 06/10/2019 08:41 PM - kou (Kouhei Sutou)

- Status changed from Open to Rejected

CSV implementation was changed to improve performance. The new implementation reads data as chunk (\$stdin.gets(nil, 4096)) instead of lines (\$stdin.gets) from input. \$. doesn't work with reading data as chunk.

It's backward incompatible but I don't want to keep the previous behavior. Because it blocks performance improvement.

You can use CSV#lineno as alternative:

```
require 'csv'
IFS=';'
CSV_OPTIONS = { col_sep: IFS, external_encoding: Encoding::ISO_8859_1, internal_encoding: Encoding::UTF_8 }
csv = CSV.new($stdin, CSV_OPTIONS)
csv.each do |row|
  puts "::::line #{csv.lineno} row=#{row}"
end
```

CSV#lineno reports logical line number. If your CSV data has a record that includes new lines:

```
a;"with
new
line"
b;c
```

CSV#lineno reports 1 for a;"with\nnew\nline" row and 2 for b;c row.

If your CSV data doesn't have these rows, you can use CSV#lineno as alternative of \$..