

Ruby master - Feature #12839

CSV - Give not nil but empty strings for empty fields

10/14/2016 05:45 AM - 5.5 (5 5)

Status: Closed	
Priority: Normal	
Assignee: kou (Kouhei Sutou)	
Target version:	
Description The CSV parser gives nil for empty fields. <pre>require "csv"</pre> <pre>CSV.parse(% ,"") #=> [[nil, ""]]</pre> The above behavior maybe be suitable for certain programmers, but I hope to get [["", ""]]. So I had used to write the following code reluctantly till Ruby 2.1: <pre>require "csv"</pre> <pre>CSV.parse(% ,"", converters: lambda{ v v ""})</pre> <pre>#=> [["", ""]]</pre> It is wasteful, but certainly works for my purpose. However, because of #11126 , the above code does not work from Ruby 2.2. (Converters are not called for nil) I merely want an option, which makes the CSV parser give empty strings for empty fields. Namely, <pre>require "csv"</pre> <pre>CSV.parse(% ,"", string: true) #=> [["", ""]]</pre>	
Related issues: Related to Ruby master - Bug #11126: CSV field converters doesn't attempt to ... Closed	

History

#1 - 12/21/2016 02:27 PM - shyouhei (Shyouhei Urabe)

- Related to Bug #11126: CSV field converters doesn't attempt to convert nil value. added

#2 - 12/21/2016 02:30 PM - shyouhei (Shyouhei Urabe)

- Assignee set to JEG2 (James Gray)

Currently no active developers are there for CSV.

I heard the distinguish between silent and sound empty cells are intentional. However I have no idea if converter issue is a bug or design.

#3 - 02/26/2018 11:18 AM - hsbt (Hiroshi SHIBATA)

- Assignee changed from JEG2 (James Gray) to kou (Kouhei Sutou)

- Status changed from Open to Assigned

#4 - 03/10/2018 08:04 AM - kou (Kouhei Sutou)

- Status changed from Assigned to Closed

This code works again with the latest csv.

```
require "csv"

CSV.parse(%|, "|", converters: lambda{|v| v || ""})
#=> [{"", ""}]
```

#5 - 04/07/2018 01:09 AM - 5.5 (5 5)

Thank you very much for fixing [#11126](#).

But I think that the status of my ticket ([#12839](#)) should be not Closed but Rejected. Because my hope is that the CSV parser gives empty strings for empty fields.

Using converters slows the speed.

```
gem "benchmark-ips"
require "benchmark/ips"
require "csv"

csv_text = <<EOT
foo,bar,"",baz
hoge,"",temo,""
roo,goo,por,kosh
EOT

conv = ->(s){ s || "" }

Benchmark.ips 20 do |r|
  r.report "without converter" do
    CSV.parse csv_text
  end

  r.report "with converter" do
    CSV.parse csv_text, converters: conv
  end

  r.compare!
end

# Comparison:
#   without converter:    9968.4 i/s
#   with converter:      8590.4 i/s - 1.16x slower
```

#6 - 04/09/2018 03:55 PM - kou (Kouhei Sutou)

I added :nil_value option as a shortcut:

```
require "csv"

p CSV.parse('","a') # => [[nil, "", "a"]]
p CSV.parse('","a', nil_value: "") # => [{"", "", "a"]]
```

But it's not so fast:

```
require "csv"

require "benchmark/ips"

csv_text = <<CSV
foo,bar,,baz
hoge,,temo,
roo,goo,por,kosh
CSV

convert_nil = ->(s) {s || ""}

Benchmark.ips do |r|
  r.report "not convert" do
    CSV.parse(csv_text)
  end

  r.report "converter" do
    CSV.parse(csv_text, converters: convert_nil)
  end

  r.report "option" do
```

```
CSV.parse(csv_text, nil_value: "")
end
```

```
r.compare!
end
```

```
Warming up -----
not convert 742.000 i/100ms
converter 620.000 i/100ms
option 672.000 i/100ms
```

```
Calculating -----
not convert 7.480k (± 1.8%) i/s - 37.842k in 5.061095s
converter 6.289k (± 0.5%) i/s - 31.620k in 5.028042s
option 6.697k (± 3.7%) i/s - 33.600k in 5.025273s
```

```
Comparison:
not convert: 7479.8 i/s
option: 6696.8 i/s - 1.12x slower
converter: 6288.9 i/s - 1.19x slower
```