The CSV parser gives nil for empty fields.

```ruby
require "csv"
CSV.parse('%|","|') #=> [[nil, ""]]
```

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:

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

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,

```ruby
require "csv"
CSV.parse('%|","|, string: true) #=> [['", "']]
```

Related issues:
Related to Ruby master - Bug #11126: CSV field converters doesn't attempt to convert nil value. added
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)
- Status changed from Open to Assigned
- Assignee changed from JEG2 (James Gray) to kou (Kouhei Sutou)

#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 20 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

09/02/2022
CSV.parse(csv_text, nil_value: '')
end

r.compare!
end

<table>
<thead>
<tr>
<th></th>
<th>Warming up</th>
<th>Calculating</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>not convert:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>742.000 i/100ms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>converter:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>620.000 i/100ms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>option:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>672.000 i/100ms</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>not convert</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7.480k (± 1.8%) i/s -</td>
</tr>
<tr>
<td></td>
<td></td>
<td>37.842k in 5.061095s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>converter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6.289k (± 0.5%) i/s -</td>
</tr>
<tr>
<td></td>
<td></td>
<td>31.620k in 5.028042s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>option</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6.697k (± 3.7%) i/s -</td>
</tr>
<tr>
<td></td>
<td></td>
<td>33.600k in 5.025273s</td>
</tr>
</tbody>
</table>

Comparison:

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