Ruby master - Bug #4069
String#parse_csv fails to parse "\r" character embedded string
11/18/2010 12:21 PM - phasis68 (Heesob Park)

Status: Rejected
Priority: Normal
Assignee: JEG2 (James Gray)
Target version: 2.0.0
ruby -v: ruby 1.9.3dev (2010-11-18 trunk 29823) [i386-mswin32_90]

Description
=begin
C:\work>ruby -rcsv -ve 'p ["aa\rbb"].to_csv.parse_csv'
ruby 1.9.3dev (2010-11-18 trunk 29823) [i386-mswin32_90]
c:/usr/lib/ruby/1.9.1/csv.rb:1914:in block in shift': Unclosed quoted field on line 1. (CSV::MalformedCSVError)
from c:/usr/lib/ruby/1.9.1/csv.rb:1831:in loop'
from c:/usr/lib/ruby/1.9.1/csv.rb:1831:in shift'
from c:/usr/lib/ruby/1.9.1/csv.rb:1390:in parse_line'
from c:/usr/lib/ruby/1.9.1/csv.rb:2341:in parse_csv'
from -e:1:in'
=end

History
#1 - 11/19/2010 05:54 AM - ender672 (Timothy Elliott)
=begin
["aa\rbb"].to_csv results in the string "\"aa\rbb\n"

When you don't specify a row separator the ruby CSV library makes a guess by searching for the first occurrence of \r or \n.

In the case of "\"aa\rbb\n" it encounters the \r and assumes that it is your row separator. In order to point it to the correct row separator, you have to supply the option :row_sep => "\n":

$ ruby -rcsv -ve 'p ["aa\rbb"].to_csv.parse_csv(:row_sep => "\n")'
ruby 1.9.3dev (2010-11-19 trunk 29830) [x86_64-linux]
["aa\rbb"]
=end

#2 - 11/19/2010 06:37 AM - JEG2 (James Gray)
- Status changed from Open to Rejected
- Assignee set to JEG2 (James Gray)

=begin
Sorry, not sure how I missed this ticket. As Timothy says, this is intended documented behavior:

```ruby
# <b>&lt;t&gt;:row_sep&lt;/t&gt;</b>:: The String appended to the end of each row. This can be set to the special
# &lt;t&gt;:auto&lt;/t&gt; setting, which requests
# that CSV automatically discover this
# from the data. Auto-discovery reads
# ahead in the data looking for the next
# &lt;t&gt;"\n"&lt;/t&gt;, &lt;t&gt;"\r"&lt;/t&gt;, or
# &lt;t&gt;"\r\n"&lt;/t&gt; sequence. A sequence
# will be selected even if it occurs in
# a quoted field, assuming that you
# would have the same line endings
# there. If none of those sequences is
# found, &lt;data&gt; is &lt;t&gt;ARGV&lt;/t&gt;, 
# &lt;t&gt;STDIN&lt;/t&gt;, &lt;t&gt;STDOUT&lt;/t&gt;, or
# &lt;t&gt;STDERR&lt;/t&gt;, or the stream is only
# available for output, the default
# &lt;t&gt;$INPUT_RECORD_SEPARATOR&lt;/t&gt;
# (&lt;t&gt;$\&lt;/t&gt;) is used. Obviously,
```

09/12/2021
discovery takes a little time. Set manually if speed is important. Also note that IO objects should be opened in binary mode on Windows if this feature will be used as the line-ending translation can cause problems with resetting the document position to where it was before the read ahead. This String will be transcoded into the data's Encoding before parsing.

=end