Ruby 1.8 - Bug #2647

Lack of testing for String#split

01/25/2010 09:09 PM - hgs (Hugh Sasse)

Status:	Closed
Priority:	Normal
Assignee:	
Target version:	
ruby -v:	ruby 1.8.7 (2008-08-11 patchlevel 72) [i386-cygwin]

Description

=begin
Having raised the issue of docs for String#split, I see there seem to be no tests. The attached patch seems to fix this [i.e is tested against the cygwin build] and is based on the documentation (as amended by my recent suggestion). Hope this is OK. It is a patch against the svn ruby_1_8.

This isn't exactly a feature for Ruby, it's not drawn from Ruby Spec, and it isn't a backport, so that only leaves "bug". I'm not sure it's that, it's more an "enhancement". Unless lack of test coverage is a bug.
=end

History

#1 - 01/25/2010 10:05 PM - vvs (Vladimir Sizikov)

=begin
Actually, it would be really nice to put such common/core tests into RubySpec, so that all various Ruby implementations would benefit from it. And since MRI 1.9.2 goal to pass 100% RubySpec as well, that would benefit MRI as well.

In fact, there are quite some tests for String#split in RubySpec already:
http://github.com/rubyspec/rubyspec/blob/master/core/string/split_spec.rb

If some of your tests provide better coverage in some cases, let me know and we could get them incorporated in RubySpec.

For more details about RubySpec, take a look here: http://rubyspec.org/
Main repository: http://github.com/rubyspec/rubyspec
=end

#2 - 01/25/2010 10:38 PM - headius (Charles Nutter)

=begin
Here here! The tests in RubySpec are vastly better than those in MRI,
and I strongly suggest the Ruby core team should be running RubySpec
to test MRI (perhaps in addition to existing tests, since not all
libraries are well covered by RubySpec). That's the bottom line.

As it stands now, MRI seems to be the only implementation *not* using
RubySpec on a daily basis. Doesn't that seem a little odd?

On Mon, Jan 25, 2010 at 2:05 PM, Vladimir Sizikov redmine@ruby-lang.org wrote:

> Issue #2647 has been updated by Vladimir Sizikov.
>
> Actually, it would be really nice to put such common/core tests into RubySpec, so that all various Ruby implementations would benefit from it. And since MRI 1.9.2 goal to pass 100% RubySpec as well, that would benefit MRI as well.
>
> In fact, there are quite some tests for String#split in RubySpec already:
> http://github.com/rubyspec/rubyspec/blob/master/core/string/split_spec.rb
>
> If some of your tests provide better coverage in some cases, let me know and we could get them incorporated in RubySpec.
>
> For more details about RubySpec, take a look here: http://rubyspec.org/
>
> ## Main repository: http://github.com/rubyspec/rubyspec
>
> http://redmine.ruby-lang.org/issues/show/2647

> http://redmine.ruby-lang.org

=end

#3 - 01/25/2010 11:22 PM - naruse (Yui NARUSE)

=begin
RubySpec is suitable for Implementation independent tests.

Ruby 1.9 already has a test for String#split but 1.8 doesn't.

> As it stands now, MRI seems to be the only implementation *not* using
> RubySpec on a daily basis. Doesn't that seem a little odd?

Current RubySpec for Ruby 1.9 has too many failure because of tests.
(yeah, we know we should fix tests; that's why we postponed 1.9.2 release)

Another reason is test-all is white box tests for MRI.
=end

#4 - 01/25/2010 11:27 PM - naruse (Yui NARUSE)

- Status changed from Open to Closed

- % Done changed from 0 to 100

=begin
This issue was solved with changeset r26410.
Hugh, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

=end

#5 - 01/25/2010 11:28 PM - hgs (Hugh Sasse)

=begin
OK, can someone please document the proper procedure for this, then? At present

http://www.ruby-lang.org/en/community/ruby-core/#patching-ruby

says we can send patches to the list, or put them in redmine, but I'm getting the impression some people say they get lost unless they are in redmine. Posting them to redmine means the attached patch doesn't show up as an attachement on the list. Redmine refers to RubySpec, and you're telling me that I should put tests into there. But the Documentation for Ruby is based on Rdoc, and my patch agrees with that. Yet, as my recent post about Zlib::GzipReader points out, we also have docs that differ from the Rdoc in RD format, still being shipped out with Ruby.

Can we have a process that is DRY, please?

Vladimir Sizikov wrote:
"If some of your tests provide better coverage in some cases, let me know"

The link is there in the body of the redmine posting. If you are asking me to read up on RubySpec, please could you look at the 93 line patch? It is mostly asserts to try and avoid dependency on Array#each, an entirely different fundamental type. So as code goes it is light reading, if rather dull :-).
=end

#6 - 01/25/2010 11:47 PM - hgs (Hugh Sasse)

=begin
On Mon, 25 Jan 2010, Yui NARUSE wrote:

> Issue #2647 has been updated by Yui NARUSE.
>
> Status changed from Open to Closed
> % Done changed from 0 to 100
>
> This issue was solved with changeset r26410.
> Hugh, thank you for reporting this issue.
> Your contribution to Ruby is greatly appreciated.
> May Ruby be with you.

Thank you, but when I did svn update my patch seems to have
been added three times, redefining the method test_split

twice as a result. I can't see how to re-open this on
redmine to flag that up. Do I submit a new bug or what?

```
    Hugh
```

=end

#7 - 01/26/2010 12:13 AM - naruse (Yui NARUSE)

=begin
Sorry, I missed the patch; thank you
=end

#8 - 01/26/2010 12:36 AM - hgs (Hugh Sasse)

=begin
On Tue, 26 Jan 2010, Yui NARUSE wrote:

> Issue #2647 has been updated by Yui NARUSE.
>
> Sorry, I missed the patch; thank you

No problem, that looks right to me now. Thank you for your trouble.

Had a look at Ruby Spec: Seems to require a number of things not
shipped with Ruby, rspec[?] and a Mocking Framework, requires a
github account to be able to contribute. The tests in RubySpec
don't seem to include Regexps with captures, which mine do. I
don't yet know how to rewrite mine to fit Ruby Spec syntax.

```
    Hugh
```

> ---
>
> http://redmine.ruby-lang.org/issues/show/2647
>
> ---
>
> http://redmine.ruby-lang.org

=end

#9 - 01/26/2010 01:11 AM - mame (Yusuke Endoh)

=begin
Hi,

2010/1/25 Charles Oliver Nutter headius@headius.com:

> Here here! The tests in RubySpec are vastly better than those in MRI,

Indeed, such common tests are also suitable for RubySpec and tests in
MRI 1.8 are really very short, but I think that current tests in 1.9
is not so bad. It achieves decent coverage.

http://dame.dyndns.org:7001/20100125/ruby/
http://dame.dyndns.org/misc/misc/test-based-code-reading.ppt (at RubyKaigi 2008)

> As it stands now, MRI seems to be the only implementation *not* using
> RubySpec on a daily basis. Doesn't that seem a little odd?

Yugui said that 1.9.2 would pass rubyspec (though I guess it is the
showstopper of 1.9.2 release.)

--
Yusuke ENDOH mame@tsg.ne.jp

=end

#10 - 01/26/2010 02:34 AM - rue (Eero Saynatkari)

=begin
Excerpts from Hugh Sasse's message of Mon Jan 25 17:36:06 +0200 2010:

> Had a look at Ruby Spec: Seems to require a number of things not shipped with Ruby, rspec[?] and a Mocking Framework, requires a github account to be able to contribute.

RubySpec uses MSpec rather than RSpec. It was written as a part of the project, and is essentially a vastly simpler codebase. Some additions like platform- and version guards, tags and interpreter selection are included. I think you could currently run the specs on RSpec as the syntax is compatible. Mocking is included in MSpec itself.

In order to have commit access you will need to have an account on Github, yes. You can, however, submit patches through the tracker or even just on the mailing list.

Until people get tired of applying them and tell you to just sign up so you can do it yourself, that is :)

> `The tests in RubySpec`

> don't seem to include Regexps with captures, which mine do. I don't yet know how to rewrite mine to fit Ruby Spec syntax.

These would be an excellent addition, so hopefully you will be able to. You can ask for help on the ML or on #rubyspec on Freenode.

`http://rubyspec.org`

Eero
--
Magic is insufficiently advanced technology.

=end

#11 - 03/04/2010 09:17 PM - mg (Dawid Grzesiak)

=begin
Hi,

I suppose that:
irb(main):050:0> ".".split "."
=> []

will give me an array with two empty strings instead of empty array.
How can I do that?

Moreover:
irb(main):051:0> ".l...".split "."
=> ["", "l"]

Shouldn't it yield ["", "l", "", "", ""] array?

Thanks,
Dawid
=end

#12 - 03/04/2010 09:19 PM - mg (Dawid Grzesiak)

=begin
Or in this example:

irb(main):053:0> "..l..".split "."
=> ["", "", "l"]

there is no symmetry :/
=end

#13 - 03/04/2010 09:36 PM - hgs (Hugh Sasse)

=begin
On Thu, 4 Mar 2010, Dawid Grzesiak wrote:

Issue #2647 has been updated by Dawid Grzesiak.

Or in this example:

irb(main):053:0> "..l..".split "."
=> ["", "", "l"]

there is no symmetry :/

hgs@Q2P14HGS ~
12:27:13$ ruby -e 'puts "..l..".split(/./).inspect'
["", "", "l"]
Confirmed for regexps
hgs@Q2P14HGS ~
12:27:36$ ruby -e 'puts "..l..".split(/./,-1).inspect'
["", "", "l", "", ""]
Limit parameter solves this
hgs@Q2P14HGS ~
12:27:46$

ri String#split gives:

-- String#split
str.split(pattern=$;, [limit]) => anArray

```
  From Ruby 1.8
```

```
     [...]
  If the _limit_ parameter is omitted, trailing null fields are
  suppressed. If _limit_ is a positive number, at most that number of
  fields will be returned (if _limit_ is +1+, the entire string is
  returned as the only entry in an array). If negative, there is no
  limit to the number of fields returned, and trailing null fields
  are not suppressed.


     HTH
     Hugh
```

=end

Files

test_string_patch	2.74 KB	01/25/2010	hgs (Hugh Sasse)