CommonRuby - Feature #7845

Strip doesn't handle unicode space characters in ruby 1.9.2 & 1.9.3 (does in 1.9.1)

02/14/2013 12:27 AM - timothyg56 (Timothy Garnett)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	

Description

Strip and associated methods in ruby 1.9.2 and 1.9.3 do not remove leading/trailing unicode space characters (such as non-breaking space \u00A0 and ideographic space \u3000) unlike ruby 1.9.1. I'd expect the 1.9.1 behavior. Looking at the underlying native lstrip! and rstrip! methods it looks like this is because 1.9.1 uses rb_enc_isspace() whereas 1.9.2+ uses rb_isspace().

1.9.1p378 :001 > "\u3000\u00a0".strip
=> ""

1.9.2p320 :001 > "\u3000\u00a0".strip
=> "□ "

1.9.3p286 :001 > "\u3000\u00a0".strip
=> "□ "

Related issues:

Related to Ruby master - Feature #2093: String#strip□□□□\s□[:space:]□	**Closed**	**09/13/2009**

History

#1 - 02/17/2013 02:13 PM - ko1 (Koichi Sasada)

- Subject changed from Strip doesn't handle unicode space characters in ruby 1.9.2 & 1.9.3 (does in 1.9.1) to Strip doesn't handle unicode space characters in ruby 1.9.2 & 1.9.3 (does in 1.9.1)

- Category set to M17N

- Assignee set to naruse (Yui NARUSE)

#2 - 02/18/2013 12:28 AM - naruse (Yui NARUSE)

- Subject changed from Strip doesn't handle unicode space characters in ruby 1.9.2 & 1.9.3 (does in 1.9.1) to Strip doesn't handle unicode space characters in ruby 1.9.2 & 1.9.3 (does in 1.9.1)

- Status changed from Open to Rejected

The behavior is intended.
see also http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-core/19379

#3 - 02/26/2013 05:12 AM - timothyg56 (Timothy Garnett)

I'm not sure how convincing the linked conversation is. It seems to be about case sensitivity issues in varying locales particularly around identifiers, but whether a unicode space is a whitespace or not is not locale dependent as far as I know. It seems like strip, which is just whitespace, could easily be encoding aware while upcase/downcase and the like were ascii only for the cited complexity reasons.

It would be nice if strip removed the equivalent of :space: as it used to, but I guess that's what open source is for. If anyone stumbling upon this wants to patch ruby itself to restore the old behavior see https://gist.github.com/tgarnett/5032660 for ruby source or you can monkey patch in a fix to string

```
class String
def lstrip
sub(/:space:+/, '')
end
def rstrip
sub(/:space:+$/, '')
end
def strip
lstrip.rstrip
end
# etc. for ! versions
end
```

#4 - 03/01/2013 02:41 AM - abscond (James Darling)

I'm not sure I understand the rationale behind rejecting this issue based on locale issues. I'm in support of this ticket, and will try and grab someone who can respond to naruse's concerns.

#5 - 03/13/2013 11:56 PM - threedaymonk (Paul Battley)

It's true that whitespace *can* be locale-dependent, at least insofar as Unix locales can specify which codepoints are to be considered as whitespace (in addition to space, tab, etc.).

The linked conversation about capitalisation doesn't seem relevant, though. The problem with capitalisation is that it really is language-specific: uppercase i is not I in Turkish, for example. Space, on the other hand isn't; it's just that the inventory of spaces used and their encoding depends on the locale: you won't find a double-width space in English, and the representation in UTF-8 and EUC-JP is different.

However, given that String#upcase/downcase are basically useless for non-ASCII content, I'm not sure I'd expect strip to handle Unicode spaces either.

#6 - 03/14/2013 12:11 AM - marcandre (Marc-Andre Lafortune)

- Status changed from Rejected to Open

- Target version set to 2.1.0

Let's reopen this issue.

Yui: could you explain why strip wouldn't remove leading and trailing /\p{space}/ ? I can only see upside to this, but maybe you can point out downside?

#7 - 05/05/2013 01:50 AM - timothyg56 (Timothy Garnett)

A patch for this is pretty straightforward, see https://gist.github.com/tgarnett/5032660 which is only a couple of lines.

As someone dealing with a lot of web crawling and chinese source data, having strip remove non-breaking / ideographic spaces is a real boon (particularly given the large amount of code we have originally written to 1.9.1).

#8 - 05/05/2013 02:03 AM - naruse (Yui NARUSE)

- Status changed from Open to Rejected

marcandre (Marc-Andre Lafortune) wrote:

> Let's reopen this issue.
>
> Yui: could you explain why strip wouldn't remove leading and trailing /\p{space}/ ? I can only see upside to this, but maybe you can point out downside?

Did you read [ruby-core:19379]?

#9 - 05/06/2013 02:28 PM - marcandre (Marc-Andre Lafortune)

- Status changed from Rejected to Open

- Assignee deleted (naruse (Yui NARUSE))

naruse (Yui NARUSE) wrote:

> Did you read [ruby-core:19379]?

I did.

Out of respect, I will assume that you read [ruby-core:53374] which explains nicely that [ruby-core:19379] has absolutely nothing to do with what constitutes a space and how strip should behave.

It would have been appreciated if instead of repeating your reference (which some of us believe not to be relevant) you would explain why you still feel it is of any relevance. This would be the helpful and polite thing to do.

I would also appreciate if you answered my questions too. For your convenience:

> Yui: could you explain why strip wouldn't remove leading and trailing /\p{space}/ ? I can only see upside to this, but maybe you can point out downside?

Finally, could you please tell me why you rejected again this issue? Maybe if some people disagree with you, including a fellow committer, maybe the

right thing to do is to explain yourself and ask Matz for his decision instead, would you not agree?

#10 - 05/06/2013 05:58 PM - matz (Yukihiro Matsumoto)

Five yeas have passed since the decision in [ruby-core:19379], and Unicode had almost taken over the world.
Maybe it's time to relax the limitation at least when we are using Unicode.

Matz.

#11 - 05/06/2013 07:23 PM - naruse (Yui NARUSE)

- Tracker changed from Bug to Feature

- Project changed from Ruby master to CommonRuby

- Category deleted (M17N)

- Target version deleted (2.1.0)

#12 - 05/06/2013 09:25 PM - naruse (Yui NARUSE)

Current string-related policy is ASCII-based.
If it is changed, how wide it is applied is the issue; for example

- strip
- split
- upcase/downcase

#13 - 05/06/2013 09:52 PM - matz (Yukihiro Matsumoto)

Everything that can be resolved without locale/language information (for most of the cases).
Case conversion may have problems with some characters, e.g. german eszett or turkish i, but can be converted mostly.
Of course it would take time to implement everything, but the basic principle (and goal) will be.

Matz.

#14 - 05/06/2013 10:53 PM - akr (Akira Tanaka)

2013/5/6 matz (Yukihiro Matsumoto) matz@ruby-lang.org:

> Everything that can be resolved without locale/language information (for most of the cases).
> Case conversion may have problems with some characters, e.g. german eszett or turkish i, but can be converted mostly.
> Of course it would take time to implement everything, but the basic principle (and goal) will be.

Be careful about network libraries.

Text based network protocol itself is not Unicode based in general.
If a primitive used in a such library is extended to Unicode, it may
cause mismatch to the protocol.

For example, the result of "grep -r strip lib/net" should be examined to
strip is used properly or not.
--
Tanaka Akira

#15 - 05/07/2013 01:12 AM - matz (Yukihiro Matsumoto)

Akira, Thank you for pointing out.

But it's hard for me to imagine concrete problematic cases.
When text from network connection is marked as Unicode, that's OK to process them as Unicode text,
otherwise they should be marked as 'ASCII-8BIT' so that #strip and other methods should behave as
they are now.

Matz.

#16 - 05/07/2013 01:59 AM - naruse (Yui NARUSE)

matz (Yukihiro Matsumoto) wrote:

> Akira, Thank you for pointing out.
>
> But it's hard for me to imagine concrete problematic cases.
> When text from network connection is marked as Unicode, that's OK to process them as Unicode text,
> otherwise they should be marked as 'ASCII-8BIT' so that #strip and other methods should behave as
> they are now.

Matz.

Modern protocol like SMTPUTF8 http://tools.ietf.org/html/rfc6532 and URL Standard http://url.spec.whatwg.org/
use UTF-8 as its character encoding, but they use ASCII whitespace.

#17 - 05/07/2013 10:03 AM - matz (Yukihiro Matsumoto)

=begin
Thank you for valuable input.

That indicates the need for something like (({str.strip(:ascii)})), or opposite (({str.strip(:utf8)}))

Matz.
=end