

Backport187 - Bug #2378

Regression in ParseDate.parsedate('nn-nn')

11/17/2009 05:18 AM - vvs (Vladimir Sizikov)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	
ruby -v:	ruby 1.8.7 (2009-06-12 patchlevel 174) [i386-mswin32]

Description

=begin
On MRI 1.8.6, ParseDate.parsedate('nn-nn') returns sensible data:

```
ruby -rparsedate -e "p ParseDate.parsedate('08-09')"  
ruby 1.8.6 (2007-09-24 patchlevel 111) [i386-mswin32]  
[8, 9, nil, nil, nil, nil, nil, nil]
```

Essentially, it treats nn-nn as yy-mm.

In MRI 1.8.7, there is a regression, and the returned values now are less sensible:

```
ruby 1.8.7 (2009-06-12 patchlevel 174) [i386-mswin32]  
[nil, nil, 8, nil, nil, nil, "-09", nil]
```

Day and weird zone, huh?

This also leads to RubySpec failures on MRI 1.8.7.

=end

History

#1 - 11/17/2009 05:40 AM - vvs (Vladimir Sizikov)

=begin
Sorry, this should be filed in 1.8.7 category, didn't know that there is that special category for 1.8.7 now.
=end

#2 - 11/24/2009 07:02 AM - tadf (tadayoshi funaba)

=begin
1.8.7 or later, parsedate does not support what yours pattern.
since, parsedate supports yymdd[+]-jzzzz more.

e.g.) 20091011-0900, 091011-0900, 1011-0900, 11-0900, 11-09.

i judged latter is more important.
=end

#3 - 11/24/2009 11:59 AM - hasari (Hiro Asari)

=begin
I beg to differ. If you tell 100 programmers that "11-09" is some combination of date and time, I wager that the vast majority will answer: "November 9th", "11th of September" or "November, 2009", maybe "2011 September".

Maybe we should only support ISO 8601, in which case "nn-nn" seems invalid (please correct me if I'm wrong).
=end

#4 - 11/24/2009 04:36 PM - vvs (Vladimir Sizikov)

=begin
Thank you for the comments. I agree with Hiro, this change of behavior is incompatible with 1.8.6 and surprising.

But I don't really have strong feelings about this. If you state state that this is an intentional change and it will stay that way, that's OK for me as well. But I do need that statement, since depending on it we either change RubySpecs to enforce this new behavior or we continue to require the old behavior marking current MRI 1.8.7 behavior as bug.

=end

#5 - 11/24/2009 08:55 PM - tadf (tadayoshi funaba)

=begin

I beg to differ. If you tell 100 programmers that "11-09" is some combination of date and time, I wager that the vast majority will answer: "November 9th", "11th of September" or "November, 2009", maybe "2011 September".

maybe.
that is a point.
this form is so ambiguous.

in parsedate, "09/11" means sept 11, not mean '09 nov.
yy-mm confuses users.
and parsedate accept --mm-dd (an iso 8601 form)
anyway i think yy-mm is not so important.

i think that it is interesting rubyspec has original interpretation.

=end

#6 - 11/25/2009 02:42 PM - hasari (Hiro Asari)

=begin

I agree that the intended meaning of "11-09" is ambiguous, but the current behavior is surprising all the same.

I am OK with Ruby behaving the way it is, as long as it is specified.
=end

#7 - 11/25/2009 11:29 PM - RickDeNatale (Rick DeNatale)

=begin

On Mon, Nov 23, 2009 at 9:59 PM, Hiro Asari redmine@ruby-lang.org wrote:

Issue [#2378](#) has been updated by Hiro Asari.

I beg to differ. If you tell 100 programmers that "11-09" is some combination of date and time, I wager that the vast majority will answer: "November 9th", "11th of September" or "November, 2009", maybe "2011 September".

Maybe we should only support ISO 8601, in which case "nn-nn" seems invalid (please correct me if I'm wrong).

ISO 8601 is really for computer interchange of time date information
as I understand it.

That's important, but most use cases of date parsing would I think
deal with parsing user input, and that has to deal with the vagaries
of world culture:

http://en.wikipedia.org/wiki/Date_and_time_notation_by_country

To most non-Americans 9/11 means September 11, but to the rest of the
world it means November 9. Personally when I write dates I tend to
write either 11 September or 9 November for this reason.

Ruby 1.8 tried to deal with this with heuristics, but I realize that
the problem is pretty hard,

--
Rick DeNatale

Blog: <http://talklikeaduck.denhaven2.com/>
Twitter: <http://twitter.com/RickDeNatale>
WWR: <http://www.workingwithrails.com/person/9021-rick-denatale>
LinkedIn: <http://www.linkedin.com/in/rickdenatale>

=end

#8 - 11/27/2009 07:31 PM - duerst (Martin Dürst)

=begin

On 2009/11/25 23:29, Rick DeNatale wrote:

On Mon, Nov 23, 2009 at 9:59 PM, Hiro Asari redmine@ruby-lang.org wrote:

Issue [#2378](#) has been updated by Hiro Asari.

I beg to differ. If you tell 100 programmers that "11-09" is some combination of date and time, I wager that the vast majority will answer: "November 9th", "11th of September" or "November, 2009", maybe "2011 September".

Maybe we should only support ISO 8601, in which case "nn-nn" seems invalid (please correct me if I'm wrong).

ISO 8601 is really for computer interchange of time date information as I understand it.

That's important, but most use cases of date parsing would I think deal with parsing user input, and that has to deal with the vagaries of world culture:

http://en.wikipedia.org/wiki/Date_and_time_notation_by_country

To most non-Americans 9/11 means September 11, but to the rest of the

I think that should read "To most Americans 9/11 means September 11".

world it means November 9. Personally when I write dates I tend to write either 11 September or 9 November for this reason.

Ruby 1.8 tried to deal with this with heuristics, but I realize that the problem is pretty hard,

Yes. "pretty hard" is a strong understatement, actually.

Regards, Martin.

--

Martin J. Dürst, Professor, Aoyama Gakuin University
<http://www.sw.it.aoyama.ac.jp> mailto:duerst@it.aoyama.ac.jp

=end

#9 - 11/25/2010 01:55 PM - shyouhei (Shyouhei Urabe)

=begin

Hello -- Can someone tell me the conclusion of this issue? Was this a bug to be fixed?

=end