Ruby - Bug #3411

Time.local 1916,5,1 #=> 1916-04-30 23:00:00 +0100

06/09/2010 03:58 AM - Eregon (Benoit Daloze)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:	1.9.2		
ruby -v:	ruby 1.9.3dev (2010-06-08 trunk 28202) [x86_64-darwin10.3.0]	**Backport:**	

Description

=begin
Everything is in the title.

My apologies if this is normal behavior, but I think is not.

It is the only first day of a month of the 20th century to behave like this
(and also the only day in -4000..4000 to not respect Time.local(y,m,d).day == d):
(1901..2000).each { |year|
(1..12).each { |month|
p Time.local(year, month, 1) if Time.local(year, month, 1).day != 1
}
} #=> 1916-04-30 23:00:00 +0100

1.8.7 gives:
Time.local 1916,5,1 #=> Mon May 01 01:00:00 +0200 1916

I believe this also happen in 1.9.2RC, though I can not test it myself.
=end

History

#1 - 06/09/2010 05:45 AM - Eregon (Benoit Daloze)

=begin
On 8 June 2010 21:53, brabuhr@gmail.com wrote:

> On Tue, Jun 8, 2010 at 2:58 PM, Benoit Daloze redmine@ruby-lang.org
> wrote:
>
> > Bug #3411: Time.local 1916,5,1 #=> 1916-04-30 23:00:00 +0100
> > http://redmine.ruby-lang.org/issues/show/3411
> >
> > Author: Benoit Daloze
> > Status: Open, Priority: Normal
> > Category: core, Target version: 1.9.2
> > ruby -v: ruby 1.9.3dev (2010-06-08 trunk 28202) [x86_64-darwin10.3.0]
> >
> > 1.8.7 gives:
> > Time.local 1916,5,1 #=> Mon May 01 01:00:00 +0200 1916
> >
> > I believe this also happen in 1.9.2RC, though I can not test it myself.
>
> Here's all the rubies I have available at the moment:
>
> ruby 1.9.2dev (2009-07-18 trunk 24186) [i686-linux]
> 1916-05-01 00:00:00 -0500
>
> ruby 1.9.2dev (2010-05-31 revision 28117) [i686-linux]
> 1916-05-01 00:00:00 -0500
>
> rubinius 1.0.0 (1.8.7 release 2010-05-14 JI) [i686-pc-linux-gnu]
> Mon May 01 00:00:00 -0500 1916
>
> jruby 1.5.0 (ruby 1.8.7 patchlevel 249) (2010-05-12 6769999) (OpenJDK

> Client VM 1.6.0_18) [i386-java]
> Mon May 01 00:00:00 -0500 1916
>
> jruby 1.5.0 (ruby 1.9.2dev trunk 24787) (2010-05-12 6769999) (OpenJDK
> Client VM 1.6.0_18) [i386-java]
> Mon May 01 00:00:00 -0500 1916
>
> ruby 1.8.7 (2010-01-10 patchlevel 249) [i486-linux]
> Mon May 01 00:00:00 -0500 1916
>
> Then maybe it is a platform problem
> or a zone problem (you are in -5, so it is likely not happening).
> Could someone checks on OSX with trunk and 1.9.2 ?

For my part I still get:
$ ruby -e 'p Time.local 1916,5,1'
1916-04-30 23:00:00 +0100

On 8 June 2010 21:53, <brabuhr@gmail.com> wrote:

> On Tue, Jun 8, 2010 at 2:58 PM, Benoit Daloze <redmine@ruby-lang.org> wrote:
> > Bug #3411: Time.local 1916,5,1 #=> 1916-04-30 23:00:00 +0100
> > http://redmine.ruby-lang.org/issues/show/3411
> >
> > Author: Benoit Daloze
> > Status: Open, Priority: Normal
> > Category: core, Target version: 1.9.2
> > ruby -v: ruby 1.9.3dev (2010-06-08 trunk 28202) [x86_64-darwin10.3.0]
> >
>
> > 1.8.7 gives:
> > Time.local 1916,5,1 #=> Mon May 01 01:00:00 +0200 1916
> >
> > I believe this also happen in 1.9.2RC, though I can not test it myself.
>
> Here's all the rubies I have available at the moment:
>
> ruby 1.9.2dev (2009-07-18 trunk 24186) [i686-linux]
> 1916-05-01 00:00:00 -0500
>
> ruby 1.9.2dev (2010-05-31 revision 28117) [i686-linux]
> 1916-05-01 00:00:00 -0500
>
> rubinius 1.0.0 (1.8.7 release 2010-05-14 JI) [i686-pc-linux-gnu]
> Mon May 01 00:00:00 -0500 1916
>
> jruby 1.5.0 (ruby 1.8.7 patchlevel 249) (2010-05-12 6769999) (OpenJDK
> Client VM 1.6.0_18) [i386-java]
> Mon May 01 00:00:00 -0500 1916
>
> jruby 1.5.0 (ruby 1.9.2dev trunk 24787) (2010-05-12 6769999) (OpenJDK
> Client VM 1.6.0_18) [i386-java]
> Mon May 01 00:00:00 -0500 1916
>
> ruby 1.8.7 (2010-01-10 patchlevel 249) [i486-linux]
> Mon May 01 00:00:00 -0500 1916

Then maybe it is a platform problem
or a zone problem (you are in -5, so it is likely not happening).
Could someone checks on OSX with trunk and 1.9.2 ?

For my part I still get:
$ ruby -e 'p Time.local 1916,5,1'
1916-04-30 23:00:00 +0100

=end

#2 - 06/09/2010 06:06 AM - Eregon (Benoit Daloze)

=begin

> Then maybe it is a platform problem
> or a zone problem (you are in -5, so it is likely not happening).
> Could someone checks on OSX with trunk and 1.9.2 ?
>
> For my part I still get:
> $ ruby -e 'p Time.local 1916,5,1'
> 1916-04-30 23:00:00 +0100

To clarify the problem: (first build from adding a day(86400), second
from Time.local)

> 5.times {|i| t=Time.local(1916,4,29)+i*86400; p t; d=i>1 ? Time.local(1916,5,i-1) : Time.local(1916,4,29+i); p d; puts}
> 1916-04-29 00:00:00 +0100
> 1916-04-29 00:00:00 +0100

1916-04-30 00:00:00 +0100
1916-04-30 00:00:00 +0100

So the time zone changes, and this probably cause the bug.

1916-05-01 01:00:00 +0200
1916-04-30 23:00:00 +0100

1916-05-02 01:00:00 +0200
1916-05-02 00:00:00 +0200

1916-05-03 01:00:00 +0200
1916-05-03 00:00:00 +0200

=end

#3 - 06/09/2010 07:27 AM - matz (Yukihiro Matsumoto)

=begin
Hi,

Cannot reproduce on my timezone. Could you tell us your timezone
setting?

```
matz.
```

In message "Re: [ruby-core:30672] [Bug #3411] Time.local 1916,5,1 #=> 1916-04-30 23:00:00 +0100"
on Wed, 9 Jun 2010 03:58:16 +0900, Benoit Daloze redmine@ruby-lang.org writes:

|Everything is in the title.
|
|My apologies if this is normal behavior, but I think is not.
|
|It is the only first day of a month of the 20th century to behave like this
|(and also the only day in -4000..4000 to not respect Time.local(y,m,d).day == d):
|(1901..2000).each { |year|
| (1..12).each { |month|
| p Time.local(year, month, 1) if Time.local(year, month, 1).day != 1
| }
|} #=> 1916-04-30 23:00:00 +0100
|
|1.8.7 gives:
|Time.local 1916,5,1 #=> Mon May 01 01:00:00 +0200 1916
|
|I believe this also happen in 1.9.2RC, though I can not test it myself.

=end

#4 - 06/09/2010 07:34 AM - hasari (Hiro Asari)

=begin
On Jun 8, 2010, at 4:06 PM, Benoit Daloze wrote:

> > Then maybe it is a platform problem

> or a zone problem (you are in -5, so it is likely not happening).
> Could someone checks on OSX with trunk and 1.9.2 ?
>
> For my part I still get:
> $ ruby -e 'p Time.local 1916,5,1'
> 1916-04-30 23:00:00 +0100

To clarify the problem: (first build from adding a day(86400), second from Time.local)

> 5.times {|i| t=Time.local(1916,4,29)+i*86400; p t; d=i>1 ? Time.local(1916,5,i-1) : Time.local(1916,4,29+i); p d; puts}
> 1916-04-29 00:00:00 +0100
> 1916-04-29 00:00:00 +0100

1916-04-30 00:00:00 +0100
1916-04-30 00:00:00 +0100

So the time zone changes, and this probably cause the bug.

1916-05-01 01:00:00 +0200
1916-04-30 23:00:00 +0100

1916-05-02 01:00:00 +0200
1916-05-02 00:00:00 +0200

1916-05-03 01:00:00 +0200
1916-05-03 00:00:00 +0200

I think this is most definitely a time zone thing:

$ uname -a; TZ=Europe/Amsterdam ruby1.9 -e 'p Time.local 1916,5,1'
Darwin aotearoa.local 10.3.0 Darwin Kernel Version 10.3.0: Fri Feb 26 11:58:09 PST 2010; root:xnu-1504.3.12~1/RELEASE_I386 i386
1916-04-30 23:00:00 +0020

Besides Europe/Amsterdam, Asia/Istanbul and Europe/Brussels return this result on my machine.
=end

#5 - 06/09/2010 08:42 AM - Eregon (Benoit Daloze)

=begin
Hi !
On 9 June 2010 00:27, Yukihiro Matsumoto matz@ruby-lang.org wrote:

> Hi,
>
> Cannot reproduce on my timezone. Could you tell us your timezone
> setting?
>
> matz.

Yes, I suppose The Land of the Rising Sun is far enough from UTC :)

I am at Brussels, BRU - Belgium, CEST (UTC+2) (and so CET (UTC+1) in winter).

On 9 June 2010 00:34, Hirotsugu Asari asari.ruby@gmail.com wrote:

> I think this is most definitely a time zone thing:
> $ uname -a; TZ=Europe/Amsterdam ruby1.9 -e 'p Time.local 1916,5,1'
> Darwin aotearoa.local 10.3.0 Darwin Kernel Version 10.3.0: Fri Feb 26
> 11:58:09 PST 2010; root:xnu-1504.3.12~1/RELEASE_I386 i386
> 1916-04-30 23:00:00 +0020
> Besides Europe/Amsterdam, Asia/Istanbul and Europe/Brussels return this
> result on my machine.

Indeed, this seems particular to UTC+1
Is this expected? :
$ TZ=Europe/Bucharest ruby -e 'p Time.local 1916,5,1' # 1916-05-01
00:00:00 +0144
$ TZ=Europe/Budapest ruby -e 'p Time.local 1916,5,1' # 1916-05-01 00:00:00 +0200

$ TZ=Europe/Riga ruby -e 'p Time.local 1916,5,1' # 1916-05-01 00:00:00 +0136
$ TZ=Europe/Helsinki ruby -e 'p Time.local 1916,5,1' # 1916-05-01 00:00:00 +0140

Should I try to find some history about timezones? (if I use Time.now,
I get expected results)

Anyway, knowing timezones have changed, I think I should use only
Time.utc for old dates in which only the day matters.
But this is still surprising me.

Regards,
B.D.

=end

#6 - 06/09/2010 08:52 AM - hasari (Hiro Asari)

=begin
On Jun 8, 2010, at 6:42 PM, Benoit Daloze eregontp@gmail.com wrote:

> Hi !
> On 9 June 2010 00:27, Yukihiro Matsumoto matz@ruby-lang.org wrote:
>
> > Hi,
> >
> > Cannot reproduce on my timezone. Could you tell us your timezone
> > setting?
> >
> > matz.
>
> Yes, I suppose The Land of the Rising Sun is far enough from UTC :)
>
> I am at Brussels, BRU - Belgium, CEST (UTC+2) (and so CET (UTC+1) in
> winter).
>
> On 9 June 2010 00:34, Hirotsugu Asari asari.ruby@gmail.com wrote:
>
> > I think this is most definitely a time zone thing:
> > $ uname -a; TZ=Europe/Amsterdam ruby1.9 -e 'p Time.local 1916,5,1'
> > Darwin aotearoa.local 10.3.0 Darwin Kernel Version 10.3.0: Fri Feb 26
> > 11:58:09 PST 2010; root:xnu-1504.3.12~1/RELEASE_I386 i386
> > 1916-04-30 23:00:00 +0020
> > Besides Europe/Amsterdam, Asia/Istanbul and Europe/Brussels return
> > this
> > result on my machine.
>
> Indeed, this seems particular to UTC+1
> Is this expected? :
> $ TZ=Europe/Bucharest ruby -e 'p Time.local 1916,5,1' # 1916-05-01
> 00:00:00 +0144
> $ TZ=Europe/Budapest ruby -e 'p Time.local 1916,5,1' # 1916-05-01
> 00:00:00 +0200
> $ TZ=Europe/Riga ruby -e 'p Time.local 1916,5,1' # 1916-05-01
> 00:00:00 +0136
> $ TZ=Europe/Helsinki ruby -e 'p Time.local 1916,5,1' # 1916-05-01
> 00:00:00 +0140
>
> Should I try to find some history about timezones? (if I use Time.now,
> I get expected results)
>
> Anyway, knowing timezones have changed, I think I should use only
> Time.utc for old dates in which only the day matters.

You can read zoneinfo database and see which dates are "funny" in
which time zones.

> But this is still surprising me.
>
> Regards,
> B.D.

=end

#7 - 06/09/2010 10:21 AM - akr (Akira Tanaka)

=begin
2010/6/9 Benoit Daloze eregontp@gmail.com:

> I am at Brussels, BRU - Belgium, CEST (UTC+2) (and so CET (UTC+1) in winter).

1916-05-01 00:00:00 is not exist in Brussels.

```
% zdump -v Europe/Brussels|grep 1916
Europe/Brussels  Sun Apr 30 22:59:59 1916 UTC = Sun Apr 30 23:59:59
1916 CET isdst=0 gmtoff=3600
Europe/Brussels  Sun Apr 30 23:00:00 1916 UTC = Mon May  1 01:00:00
1916 CEST isdst=1 gmtoff=7200
Europe/Brussels  Sat Sep 30 22:59:59 1916 UTC = Sun Oct  1 00:59:59
1916 CEST isdst=1 gmtoff=7200
Europe/Brussels  Sat Sep 30 23:00:00 1916 UTC = Sun Oct  1 00:00:00
1916 CET isdst=0 gmtoff=3600
```

This shows that a second after 1916-04-30 23:59:59 is 1916-05-01 01:00:00.

So, it is difficult to define the result of Time.local(1916,5,1).

It seems Ruby 1.9.1 interpretes it as 1 second after 1916-04-30 23:59:59.
It seems Ruby 1.9.2 interpretes it as 1 hour before 1916-05-01 01:00:00.

Tanaka Akira

=end

#8 - 06/09/2010 06:01 PM - matz (Yukihiro Matsumoto)

=begin
Hi,

In message "Re: [ruby-core:30683] Re: [Bug #3411] Time.local 1916,5,1 #=> 1916-04-30 23:00:00 +0100"
on Wed, 9 Jun 2010 10:21:37 +0900, Tanaka Akira akr@fsij.org writes:

|This shows that a second after 1916-04-30 23:59:59 is 1916-05-01 01:00:00.
|
|So, it is difficult to define the result of Time.local(1916,5,1).
|
|It seems Ruby 1.9.1 interpretes it as 1 second after 1916-04-30 23:59:59.
|It seems Ruby 1.9.2 interpretes it as 1 hour before 1916-05-01 01:00:00.

I prefer the 1.9.1 behavior (or something like it), because the result
differs from the specified date in the latter case.

```
matz.
```

=end

#9 - 06/09/2010 08:00 PM - Eregon (Benoit Daloze)

=begin
Hi,
On 9 June 2010 11:01, Yukihiro Matsumoto matz@ruby-lang.org wrote:

> Hi,
>
> In message "Re: [ruby-core:30683] Re: [Bug #3411] Time.local 1916,5,1 #=> 1916-04-30 23:00:00 +0100"
> on Wed, 9 Jun 2010 10:21:37 +0900, Tanaka Akira akr@fsij.org writes:
>
> |This shows that a second after 1916-04-30 23:59:59 is 1916-05-01 01:00:00.
> |
> |So, it is difficult to define the result of Time.local(1916,5,1).
> |
> |It seems Ruby 1.9.1 interpretes it as 1 second after 1916-04-30 23:59:59.
> |It seems Ruby 1.9.2 interpretes it as 1 hour before 1916-05-01 01:00:00.
>
> I prefer the 1.9.1 behavior (or something like it), because the result

differs from the specified date in the latter case.

matz.

I do prefer the 1.9.1 behavior too, because it is more intuitive and consistent to keep the good day.
Are they any downside ? Why did it changed ?

B.D.

=end

#10 - 06/09/2010 10:57 PM - akr (Akira Tanaka)

- Status changed from Open to Closed

- % Done changed from 0 to 100

=begin
This issue was solved with changeset r28238.
Benoit, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

=end