

Ruby - Bug #4005

YAML fails to roundtrip Time objects

10/30/2010 03:25 AM - pweldon (Peter Weldon)

Status:	Closed	Backport:
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	ruby 1.9.3dev (2010-10-30 trunk 29630) [i686-linux]	
Description		
<pre>=begin \$ rvm exec ruby -v -ryaml -e 't0 = Time.now; p YAML::load(YAML::dump(t0)) == t0' ruby 1.9.3dev (2010-10-30 trunk 29630) [i686-linux] false ruby 1.8.7 (2010-08-16 patchlevel 302) [i686-linux] true ruby 1.9.1p378 (2010-01-10 revision 26273) [i686-linux] false ruby 1.9.2p0 (2010-08-18 revision 29036) [i686-linux] false =end</pre>		

History

#1 - 10/30/2010 09:11 AM - zenspider (Ryan Davis)

=begin

On Oct 29, 2010, at 11:26 , Peter Weldon wrote:

Bug #4005: YAML fails to roundtrip Time objects
<http://redmine.ruby-lang.org/issues/show/4005>

Author: Peter Weldon
Status: Open, Priority: Normal
ruby -v: ruby 1.9.3dev (2010-10-30 trunk 29630) [i686-linux]

\$ rvm exec ruby -v -ryaml -e 't0 = Time.now; p YAML::load(YAML::dump(t0)) == t0'
ruby 1.9.3dev (2010-10-30 trunk 29630) [i686-linux]
false
ruby 1.8.7 (2010-08-16 patchlevel 302) [i686-linux]
true
ruby 1.9.1p378 (2010-01-10 revision 26273) [i686-linux]
false
ruby 1.9.2p0 (2010-08-18 revision 29036) [i686-linux]
false

I just got 4 trues... can you be more specific about what timezone you're in and roughly what that t0 was?

I'm in PST and ruby -e 'p Time.now' is currently "Fri Oct 29 17:08:27 -0700 2010". My env | grep TZ is empty.

=end

#2 - 10/30/2010 11:31 AM - pweldon (Peter Weldon)

=begin
:~\$ rvm exec ruby -v -ryaml -e 't0 = Time.now; p t0; p YAML::load(YAML::dump(t0)) == t0'
ruby 1.9.3dev (2010-10-30 trunk 29630) [i686-linux]
2010-10-30 02:30:09 +0000
false
ruby 1.8.7 (2010-08-16 patchlevel 302) [i686-linux]
Sat Oct 30 02:30:10 +0000 2010
true
ruby 1.9.1p378 (2010-01-10 revision 26273) [i686-linux]
2010-10-30 02:30:10 +0000

```
false
ruby 1.9.2p0 (2010-08-18 revision 29036) [i686-linux]
2010-10-30 02:30:11 +0000
false
```

```
~$ printenv | grep TZ
~$
```

```
=end
```

#3 - 10/30/2010 02:31 PM - naruse (Yui NARUSE)

```
=begin
It seems from less than 1 second, so try following:
ruby -v -ryaml -e 't0 = Time.now;t2=YAML::load(t1=YAML::dump(t0));p [t0.to_f,t2.to_f];p t1'
=end
```

#4 - 10/30/2010 11:08 PM - tenderlovmaking (Aaron Patterson)

```
=begin
On Sat, Oct 30, 2010 at 11:31:15AM +0900, Peter Weldon wrote:
```

Issue [#4005](#) has been updated by Peter Weldon.

```
:~$ rvm exec ruby -v -ryaml -e 't0 = Time.now; p t0; p YAML::load(YAML::dump(t0)) == t0'
ruby 1.9.3dev (2010-10-30 trunk 29630) [i686-linux]
2010-10-30 02:30:09 +0000
false
ruby 1.8.7 (2010-08-16 patchlevel 302) [i686-linux]
Sat Oct 30 02:30:10 +0000 2010
true
ruby 1.9.1p378 (2010-01-10 revision 26273) [i686-linux]
2010-10-30 02:30:10 +0000
false
ruby 1.9.2p0 (2010-08-18 revision 29036) [i686-linux]
2010-10-30 02:30:11 +0000
false
```

```
~$ printenv | grep TZ
~$
```

I'm not able to reproduce this with Syck. Can you try with Psych instead? Like this:

```
ruby -v -ryaml -e 'YAML::ENGINE.yamler = "psych"; t0 = Time.now; p t0; p YAML::load(YAML::dump(t0)) == t0'
```

I suspect your error is due to fractional second support. Observe the output of Syck compared to Psych:

Using Syck

```
$ ruby -v -ryaml -e 't0 = Time.now; p t0; p YAML::dump(t0)'
```

```
"--- 2010-10-30 07:05:46.535362 -07:00\n"
```

Using Psych

```
ruby -v -ryaml -e 'YAML::ENGINE.yamler = "psych"; t0 = Time.now; p t0; p YAML::dump(t0)'
```

```
"--- 2010-10-30 07:06:48.118999000 -07:00\n...\n"
```

```
--
```

Aaron Patterson
<http://tenderlovmaking.com/>

Attachment: (unnamed)
=end

#5 - 10/30/2010 11:15 PM - darix (Marcus Rückert)

```
=begin
On 2010-10-30 23:07:52 +0900, Aaron Patterson wrote:
```

I'm not able to reproduce this with Syck. Can you try with Psych

instead? Like this:

```
ruby -v -ryaml -e 'YAML::ENGINE.yamler = "psych"; t0 = Time.now; p t0; p YAML::load(YAML::dump(t0)) == t0'
```

I suspect your error is due to fractional second support. Observe the output of Syck compared to Psych:

Using Syck

```
$ ruby -v -ryaml -e 't0 = Time.now; p t0; p YAML::dump(t0)'
```

```
"--- 2010-10-30 07:05:46.535362 -07:00\n"
```

Using Psych

```
ruby -v -ryaml -e 'YAML::ENGINE.yamler = "psych"; t0 = Time.now; p t0; p YAML::dump(t0)'
```

```
"--- 2010-10-30 07:06:48.118999000 -07:00\n...\n"
```

```
$ ruby1.9 -v -ryaml -e 'YAML::ENGINE.yamler = "psych"; t0 = Time.now; p t0; p YAML::load(YAML::dump(t0)) == t0'
ruby 1.9.2p0 (2010-08-18 revision 29036) [x86_64-linux]
/usr/lib64/ruby/1.9.1/psych/deprecated.rb:79: warning: method redefined; discarding old to_yaml_properties
/usr/lib64/ruby/1.9.1/syck/rubytypes.rb:13: warning: previous definition of to_yaml_properties was here
2010-10-30 16:11:41 +0200
true
```

```
$ ruby1.9 -v -ryaml -e 't0 = Time.now; p t0; p YAML::load(YAML::dump(t0)) == t0'
ruby 1.9.2p0 (2010-08-18 revision 29036) [x86_64-linux]
2010-10-30 16:11:50 +0200
false
```

```
$ ruby1.9 -v -ryaml -e 'YAML::ENGINE.yamler = "psych"; t0 = Time.now; p t0; p YAML::dump(t0)'
ruby 1.9.2p0 (2010-08-18 revision 29036) [x86_64-linux]
/usr/lib64/ruby/1.9.1/psych/deprecated.rb:79: warning: method redefined; discarding old to_yaml_properties
/usr/lib64/ruby/1.9.1/syck/rubytypes.rb:13: warning: previous definition of to_yaml_properties was here
2010-10-30 16:12:39 +0200
"--- 2010-10-30 16:12:39.608620883 +02:00\n...\n"
```

```
$ ruby1.9 -v -ryaml -e 't0 = Time.now; p t0; p YAML::dump(t0)'
ruby 1.9.2p0 (2010-08-18 revision 29036) [x86_64-linux]
2010-10-30 16:12:52 +0200
"--- 2010-10-30 16:12:52.775869 +02:00\n"
```

why does psych add the "...n"?

hth

darix

--

openSUSE - SUSE Linux is my linux
openSUSE is good for you
www.opensuse.org

=end

#6 - 10/31/2010 05:04 AM - zenspider (Ryan Davis)

=begin

On Oct 30, 2010, at 07:14 , Marcus Rueckert wrote:

why does psych add the "...n"?

Not psych exactly, but libyaml. It signifies the end of the document and is yaml 1.1 specific.

This uses psych and goes through libyaml:

```
% multiruby -1 1.9.2 -rpsych -ryaml -e 'p YAML::dump(42)'
"--- 42\n...\n"
```

This uses syck only:

```
% multiruby -1 1.9.2 -ryaml -e 'p YAML::dump(42)'
"--- 42\n"
```

I look forward to the day when syck is no longer in the ruby distribution and we have faster and more compliant yaml by default.

=end

#7 - 10/31/2010 09:20 AM - darix (Marcus Rückert)

=begin

On 2010-10-31 05:03:56 +0900, Ryan Davis wrote:

On Oct 30, 2010, at 07:14 , Marcus Rueckert wrote:

why does psych add the "...\\n"?

Not psych exactly, but libyaml. It signifies the end of the document and is yaml 1.1 specific.

thanks for the explanation. can i help with more details with the date roundtrip problem?

darix

--

openSUSE - SUSE Linux is my linux

openSUSE is good for you

www.opensuse.org

=end

#8 - 11/01/2010 04:02 AM - pweldon (Peter Weldon)

=begin

As pointed out fractional time does not appear to round trip very well:

```
~$ rvm exec ruby -v -ryaml -e 't0 = Time.at(0.1); p YAML::load(YAML::dump(t0)) == t0'
ruby 1.9.3dev (2010-10-30 trunk 29648) [i686-linux]
false
ruby 1.8.7 (2010-08-16 patchlevel 302) [i686-linux]
true
ruby 1.9.1p378 (2010-01-10 revision 26273) [i686-linux]
true
ruby 1.9.2p0 (2010-08-18 revision 29036) [i686-linux]
false
```

It may be inherent in the Time object as syck / psych / json suffer from it:

```
~$ rvm exec ruby -v -rpsych -ryaml -e 't0 = Time.at(0.1); p YAML::load(YAML::dump(t0)) == t0'
ruby 1.9.3dev (2010-10-30 trunk 29648) [i686-linux]
false
ruby 1.9.2p0 (2010-08-18 revision 29036) [i686-linux]
false
```

```
~$ ruby -v -rjson/add/core -e 't0 = Time.at(0.1); p JSON.parse(JSON.generate(t0)) == t0'
ruby 1.9.2p0 (2010-08-18 revision 29036) [i686-linux]
false
```

If you avoid using Time#at(Float) and explicitly pass in microseconds thing look better:

```
~$ rvm exec ruby -v -ryaml -e 't0 = Time.at(0,100000); p YAML::load(YAML::dump(t0)) == t0'
ruby 1.9.3dev (2010-10-30 trunk 29648) [i686-linux]
true
ruby 1.8.7 (2010-08-16 patchlevel 302) [i686-linux]
true
ruby 1.9.1p378 (2010-01-10 revision 26273) [i686-linux]
true
ruby 1.9.2p0 (2010-08-18 revision 29036) [i686-linux]
true
```

Understanding why Time.at(0.1) and Time.at(0,100000) are not consistently equal/unequal might be a clue:

```
~$ rvm exec ruby -v -e 'p Time.at(0.1) == Time.at(0,100000)'
```

```
ruby 1.9.3dev (2010-10-30 trunk 29648) [i686-linux]
false
ruby 1.8.7 (2010-08-16 patchlevel 302) [i686-linux]
true
ruby 1.9.1p378 (2010-01-10 revision 26273) [i686-linux]
true
ruby 1.9.2p0 (2010-08-18 revision 29036) [i686-linux]
false

=end
```

#9 - 11/08/2010 09:14 PM - naruse (Yui NARUSE)

=begin

As pointed out fractional time does not appear to round trip very well:

Yes, this is originally intended.
From Ruby 1.9.2 Time stores its internal time as Rational, so it is more precise than before.
See Time#to_r or time.c (if you are interested).

I understand your trouble.
So if you suggest a reasonable feature, we can discuss about it.
=end

#10 - 11/10/2010 02:04 AM - pweldon (Peter Weldon)

- *File 0001-Make-syck-s-Time-timestamps-write-nsec-precision-fra.patch added*

=begin

Patch syck's handling of fractional seconds. Round-tripping of nsec Time values now works.
=end

#11 - 06/26/2011 03:58 PM - naruse (Yui NARUSE)

- *Status changed from Open to Closed*

1.9.3's YAML now dumps 9 digits of fractional (6 digits before).
So this seems resolved.

Files

0001-Make-syck-s-Time-timestamps-write-nsec-precision-fra.patch	4.35 KB	11/10/2010	pweldon (Peter Weldon)
---	---------	------------	------------------------