Ruby master - Feature #2340

Removing YAML/Syck

11/06/2009 05:16 PM - naruse (Yui NARUSE)

Status:	Rejected	
Priority:	Normal	
Assignee:	tenderlovemaking (Aaron Patterson)	
Target version:	2.0.0	

Description

=begin
YAML and Syck is a _why's product and widely used bundled library of Ruby.
But they are not maintained for 2 years and no more by _why.
And they support only YAML 1.0, not 1.1 and 1.2.
So YAML/Syck considered harmful.
=end

Related issues:

Related to Ruby master - Feature #3112: require "yaml" doesn't use psych as d...	**Closed**	**04/08/2010**
Related to Ruby master - Feature #6163: Remove syck YAML extension	**Closed**	**03/17/2012**

History

#1 - 11/06/2009 05:37 PM - matz (Yukihiro Matsumoto)

=begin
Hi,

In message "Re: [ruby-core:26560] [Feature #2340] Removing YAML/Syck"
on Fri, 6 Nov 2009 17:16:12 +0900, Yui NARUSE redmine@ruby-lang.org writes:

|YAML and Syck is a _why's product and widely used bundled library of Ruby.
|But they are not maintained for 2 years and no more by _why.
|And they support only YAML 1.0, not 1.1 and 1.2.
|So YAML/Syck considered harmful.

Agreed with precondition. But do we have alternative?

```
                            matz.
```

=end

#2 - 11/06/2009 07:02 PM - naruse (Yui NARUSE)

=begin

> Agreed with precondition. But do we have alternative?

Currently no.
So this ticket intends to call for alternative or maintainer.

But if no proposal, YAML/Syck should be removed in Ruby 2.0 even if no alternative.
=end

#3 - 11/07/2009 12:09 AM - jonforums (Jon Forums)

=begin

> Issue #2340 has been updated by Yui NARUSE.
>
> > Agreed with precondition. But do we have alternative?
>
> Currently no.
> So this ticket intends to call for alternative or maintainer.
>
> But if no proposal, YAML/Syck should be removed in Ruby 2.0 even if

> > no alternative.
>
> YAML is a very popular library, unfortunately. Rails also uses it.
>
> People will definitely notice if we remove it.
>
> James Edward Gray II

If you're looking at alternatives, does http://pyyaml.org/wiki/LibYAML look potentially interesting?

Jon

=end

#4 - 11/07/2009 12:48 AM - now (Nikolai Weibull)

=begin
On Fri, Nov 6, 2009 at 16:00, James Edward Gray II
james@graysoftinc.com wrote:

> On Nov 6, 2009, at 4:02 AM, Yui NARUSE wrote:
>
> > Issue #2340 has been updated by Yui NARUSE.
> >
> > > Agreed with precondition. But do we have alternative?
> >
> > Currently no.
> > So this ticket intends to call for alternative or maintainer.
> >
> > But if no proposal, YAML/Syck should be removed in Ruby 2.0 even if no
> > alternative.
>
> YAML is a very popular library, unfortunately. Rails also uses it. People
> will definitely notice if we remove it.

How hard would it be to start using JSON instead?

=end

#5 - 11/07/2009 12:59 AM - naruse (Yui NARUSE)

=begin
Jon wrote:

> If you're looking at alternatives, does http://pyyaml.org/wiki/LibYAML
> look potentially interesting?

Yes, it looks good if someone port to ruby and maintain it.

--
NARUSE, Yui naruse@airemix.jp

=end

#6 - 11/07/2009 01:22 AM - jonforums (Jon Forums)

=begin

> > If you're looking at alternatives, does http://pyyaml.org/wiki/LibYAML
> > look potentially interesting?
>
> Yes, it looks good if someone port to ruby and maintain it.

Given recent discussions re: FFI and the previously mentioned 2.0 timeframe, which of the following alternatives is preferred?

- bottoms up C port of the library to Ruby
- C Ruby API wrapper around the existing library

> - FFI wrapper around the existing library
> - other?
>
> If the target is not the 2.0 timeframe, what alternative is preferred?
>
> Jon
>
> =end

#7 - 11/07/2009 01:31 AM - joshbuddy (Joshua Hull)

=begin
FYI, this exist:

http://github.com/cesare/ruby-libc-libyaml

j

On 2009-11-06, at 11:22 AM, Jon wrote:

> > > If you're looking at alternatives, does http://pyyaml.org/wiki/LibYAML
> > > look potentially interesting?
> >
> > Yes, it looks good if someone port to ruby and maintain it.
>
> Given recent discussions re: FFI and the previously mentioned 2.0
>
> timeframe, which of the following alternatives is preferred?
>
> - bottoms up C port of the library to Ruby
> - C Ruby API wrapper around the existing library
> - FFI wrapper around the existing library
> - other?
>
> If the target is not the 2.0 timeframe, what alternative is preferred?
>
> Jon

=end

#8 - 11/07/2009 02:45 AM - bleything (Ben Bleything)

=begin
On Fri, Nov 6, 2009 at 9:38 AM, Joel VanderWerf vjoel@path.berkeley.edu wrote:

> And references. And the symbol/string distinction. And instances of user
> defined classes. And custom emit/parse. Just off the top of my head....

and that it's far more human-readable. That part is key.

Ben

=end

#9 - 11/07/2009 09:15 AM - now (Nikolai Weibull)

=begin
On Fri, Nov 6, 2009 at 18:38, Joel VanderWerf vjoel@path.berkeley.edu wrote:

> James Edward Gray II wrote:
>
> > On Nov 6, 2009, at 9:48 AM, Nikolai Weibull wrote:
> >
> > > How hard would it be to start using JSON instead?
> >
> > I love JSON, but I don't think you will win everyone else over so easily.
> > YAML does do more than JSON, like handling dates for example.
>
> And references. And the symbol/string distinction. And instances of user

> defined classes. And custom emit/parse. Just off the top of my head....

Yes, and all those instances where there are things you can't
represent as JSON apply. That's not interesting. Can you solve the
same problems without using the things that YAML allow you to do using
JSON instead? That's the question.

=end

#10 - 11/07/2009 08:53 PM - naruse (Yui NARUSE)

=begin
Jon wrote:

> > > If you're looking at alternatives, does http://pyyaml.org/wiki/LibYAML
> > > look potentially interesting?
> > > Yes, it looks good if someone port to ruby and maintain it.
>
> Given recent discussions re: FFI and the previously mentioned 2.0 timeframe, which of the following alternatives is preferred?
>
> > - bottoms up C port of the library to Ruby
> > - C Ruby API wrapper around the existing library

If these two, we can bundled the library in next reelase.

> > - FFI wrapper around the existing library

If it depends on FFI, we can bundle after FFI comes.

> > - other?
>
> If the target is not the 2.0 timeframe, what alternative is preferred?

requirements are following:

- Current YAML/Syck API compatibility
- portability; it can run in Windows and VC++
- maintainer
- it can run standalone (if it depends on libyaml, it should bundle libyaml)

--
NARUSE, Yui naruse@airemix.jp

=end

#11 - 11/07/2009 09:00 PM - naruse (Yui NARUSE)

=begin
Joshua Hull wrote:

> FYI, this exist:
>
> http://github.com/cesare/ruby-libc-libyaml

It seems not bad.
But it doesn't have YAML/Syck API compatibility.
If it want to replace YAML/Syck, it is required.

--
NARUSE, Yui naruse@airemix.jp

=end

#12 - 11/07/2009 09:19 PM - naruse (Yui NARUSE)

=begin
Marc-Andre Lafortune wrote:

> YAML is important for some people and most will agree that it is a great format. I hope we can focus on how to offer a great YAML lib in ruby instead of on we could remove it.

I don't want to remove YAML as far as it has active maintainer.

--
NARUSE, Yui naruse@airemix.jp

=end

#13 - 11/07/2009 09:21 PM - naruse (Yui NARUSE)

=begin
Aaron Patterson wrote:

> On Sat, Nov 07, 2009 at 12:59:25AM +0900, NARUSE, Yui wrote:
>
>> Jon wrote:
>>
>>> If you're looking at alternatives, does http://pyyaml.org/wiki/LibYAML
>>> look potentially interesting?
>>> Yes, it looks good if someone port to ruby and maintain it.
>
> I will:
>
> https://github.com/tenderlove/psych

It seems good (thought it needs more docs and implementation).
Anyone has other alternatives or ideas to Aaron's?

Rough schedule is following:
1.9.2: Bundle the library as other than yaml/syck
1.9.x: Replace yaml/syck as alias to the library

--
NARUSE, Yui naruse@airemix.jp

=end

#14 - 11/07/2009 11:09 PM - now (Nikolai Weibull)

=begin
On Sat, Nov 7, 2009 at 01:31, Marc-Andre Lafortune
ruby-core-mailing-list@marc-andre.ca wrote:

>>> And references. And the symbol/string distinction. And instances of user defined classes. And custom emit/parse. Just off the top of my head....
>>
>> Yes, and all those instances where there are things you can’t represent as JSON apply. That’s not interesting. Can you solve the same problems without using the things that YAML allow you to do using JSON instead? That’s the question.
>
> Sorry, but that is not the question. We can solve all the problems in the world with assembly language and text-files. That doesn't mean it's the best way to go.

Sorry, but that is not an answer.

How many of the standard libraries need references, symbol/string
distinction, user defined classes, custom emit/parse, …?

> YAML is important for some people and most will agree that it is a great format. I hope we can focus on how to offer a great YAML lib in ruby instead of on we could remove it.

The JSON library is already part of the standard library. If
everything in the standard library can be made to use the JSON library
instead, then we can drop the dependency for now, perhaps adding a
YAML library once we have a new one that works to our satisfaction.

=end

#15 - 11/12/2009 05:47 AM - headius (Charles Nutter)

=begin
On Sat, Nov 7, 2009 at 2:52 PM, Aaron Patterson
aaron@tenderlovemaking.com wrote:

> Yes, it definitely needs more documentation. It passes *most* of the
> current Syck tests, although syck supports some syntax that isn't
> allowed by the YAML spec.

Do the Syck tests cover all the pluggable emitter stuff too? We were
never able to be totally compatible with Syck's API until Ola did a
straight-up port of Syck for JRuby 1.4. The emitter stuff was the
hardest, and it's used inside Rails in various places so we didn't
really have a choice to not support it.

We also ran into the YAML incompatibility issues on a regular basis,
usually punting the bugs as not being YAML compliant. Bottom line is
that Syck is too permissive and its API exposes a lot of how it's
implemented internally, and that's hard to escape.

Hopefully any breakage or API changes will be discussed first with
other implementations that have done a lot of work to match the
current YAML impl.

- Charlie

=end

#16 - 11/12/2009 10:50 AM - naruse (Yui NARUSE)

=begin
2009/11/12 5:47, Charles Oliver Nutter wrote:

> On Sat, Nov 7, 2009 at 2:52 PM, Aaron Patterson
> aaron@tenderlovemaking.com wrote:
>
>> Yes, it definitely needs more documentation. It passes *most* of the
>> current Syck tests, although syck supports some syntax that isn't
>> allowed by the YAML spec.
>
> Do the Syck tests cover all the pluggable emitter stuff too? We were
> never able to be totally compatible with Syck's API until Ola did a
> straight-up port of Syck for JRuby 1.4. The emitter stuff was the
> hardest, and it's used inside Rails in various places so we didn't
> really have a choice to not support it.

Yes, compatibility is very important.
But we gave up maintaining syck; so there are not so many options.
If someone contribute tests for syck, it will help Aaron's new impl
unless Aaron give up compatibility.

> We also ran into the YAML incompatibility issues on a regular basis,
> usually punting the bugs as not being YAML compliant. Bottom line is
> that Syck is too permissive and its API exposes a lot of how it's
> implemented internally, and that's hard to escape.

This is not a consensus Ruby core team but I think,
compatibility to YAML spec is more important than syck compatibility.
So new impl should follow standard YAML spec even if it breaks Rails.
# of course such changes should notice to public

> Hopefully any breakage or API changes will be discussed first with
> other implementations that have done a lot of work to match the

> current YAML impl.

I fully agree this, we should disscuss such changes
before they have done.

--
NARUSE, Yui naruse@airemix.jp

=end

#17 - 11/13/2009 02:57 AM - naruse (Yui NARUSE)

=begin
Ola Bini wrote:

> And as you might know, you will find that Syck is very incompatible
> towards YAML in many, many places - and there are lots of production
> apps that rely on those behaviors.

If you have tests for current syck or your impl,
please contribute us (MRI's test-all) or rubyspec.
And we all can know and discuss how treat such incompatibility.

Note that RubySpec is test/spec for all Ruby impl,
MRI's test-all is for MRI.

--
NARUSE, Yui naruse@airemix.jp

=end

#18 - 11/28/2009 11:55 AM - ujihisa (Tatsuhiro Ujihisa)

- Status changed from Open to Assigned

- Assignee set to matz (Yukihiro Matsumoto)

=begin

=end

#19 - 12/02/2009 11:17 AM - naruse (Yui NARUSE)

- Assignee changed from matz (Yukihiro Matsumoto) to tenderlovemaking (Aaron Patterson)

=begin

=end

#20 - 03/18/2010 10:50 PM - naruse (Yui NARUSE)

- Target version set to 2.0.0

=begin
How about this status, Aaron?

I'm ok that importing Psych is 1.9.3.
We, howerver, want to warn current Syck user about APIs which won't be available on Psych.

So please add warnings to current Syck implementation about future incompatibility.
=end

#21 - 03/19/2010 04:24 AM - headius (Charles Nutter)

=begin
We would also like to know when this is official, since we're going to
have to rewrite YAML support again.

On Thu, Mar 18, 2010 at 8:50 AM, Yui NARUSE redmine@ruby-lang.org wrote:

> Issue #2340 has been updated by Yui NARUSE.

> Target version set to 1.9.x
>
> How about this status, Aaron?
>
> I'm ok that importing Psych is 1.9.3.
> We, howerver, want to warn current Syck user about APIs which won't be available on Psych.
>
> ## So please add warnings to current Syck implementation about future incompatibility.
>
> http://redmine.ruby-lang.org/issues/show/2340
>
> ---
>
> http://redmine.ruby-lang.org

=end

#22 - 03/20/2010 08:48 PM - naruse (Yui NARUSE)

=begin
(2010/03/20 6:33), Aaron Patterson wrote:

> On Thu, Mar 18, 2010 at 10:50:18PM +0900, Yui NARUSE wrote:
>
> > Issue #2340 has been updated by Yui NARUSE.
> >
> > Target version set to 1.9.x
> >
> > How about this status, Aaron?
> >
> > I'm ok that importing Psych is 1.9.3.
>
> I'm happy to move it to ruby svn whenever. I thought you mentioned
> problems using it with MSVC though? I wanted to make sure it would work
> with MSVC before importing it.

I'm ok about importing Psych in 1.9.2 as ext/psych
until it doesn't affect current YAML impl.

When eeplace YAML is depend on its compatibility,
and Yugui decides it.

--
NARUSE, Yui naruse@airemix.jp

=end

#23 - 03/20/2010 11:12 PM - naruse (Yui NARUSE)

=begin
(2010/03/19 4:24), Charles Oliver Nutter wrote:

> We would also like to know when this is official, since we're going to
> have to rewrite YAML support again.

It's decided by Yugui.
1.9.2 feature will be freezed in this month.

--
NARUSE, Yui naruse@airemix.jp

=end

#24 - 06/08/2010 09:53 AM - naruse (Yui NARUSE)

- Status changed from Assigned to Rejected

=begin
Syck won't be removed in 1.9.x even if psych come to be default.
=end

#25 - 12/31/2011 06:31 PM - zdavatz (Zeno Davatz)

Please take good care when you introduce a new default YAML engine to Ruby 1.9.3 so people who have been using syck for 10 years will not be waked by an avalanche of changes. Currently using Psych as a drop-in replacement for syck will not_work. For highly dynamic objects spaces with a lot of recursive objects you will get following error:

/usr/local/lib/ruby/1.9.1/psych/tree_builder.rb:75: stack level too deep (SystemStackError)

We tested this here:

http://dev.ywesee.com/wiki.php/Choddb/Ruby193p0YamlExport

Please try to avoid a second Oniguruma disaster and start thinking hard about how to implement consistency across 100 years of Ruby development.

#26 - 12/31/2011 09:22 PM - alexeymuranov (Alexey Muranov)

Zeno Davatz wrote:

> Please take good care when you introduce a new default YAML engine to Ruby 1.9.3 so people who have been using syck for 10 years will not be waked by an avalanche of changes. Currently using Psych as a drop-in replacement for syck will not_work. For highly dynamic objects spaces with a lot of recursive objects you will get following error:
>
> /usr/local/lib/ruby/1.9.1/psych/tree_builder.rb:75: stack level too deep (SystemStackError)
>
> We tested this here:
>
> http://dev.ywesee.com/wiki.php/Choddb/Ruby193p0YamlExport
>
> Please try to avoid a second Oniguruma disaster and start thinking hard about how to implement consistency across 100 years of Ruby development.

I had some errors when upgrading to Rails 3.1, but it turned out there were some errors in my YAML file.

Alexey.

#27 - 01/01/2012 02:35 AM - zdavatz (Zeno Davatz)

The error is not in our YAML file. We are using the same code to dump our yaml file with Ruby 1.9.3 as we used to use with Ruby 1.8.6

The problem is that psych does not deal well with recursive data structures - specially when they are to deep. Syck solved that problem well.

Sample:

Company object has Registration objects, and Registration object has Indication object, and Indication object has Registration objects, etc.

If you do not have a recursive data structure Psych works fine.

So this is a fundamental issue and should be addressed, specially if Psych is becoming the new default for Yaml in Ruby 1.9.x

Best
Zeno