# Ruby - Feature #7375

## embedding libyaml in psych for Ruby 2.0

11/17/2012 08:42 AM - tenderlovemaking (Aaron Patterson)

| | | |
|---|---|---|
| **Status:** | Closed | |
| **Priority:** | Normal | |
| **Assignee:** | tenderlovemaking (Aaron Patterson) | |
| **Target version:** | 2.0.0 | |

**Description**

People have a hard time using psych (and thus rubygems) because it depends on libyaml. We can ease upgrades by embedding libyaml in order to eliminate the library dependency.

libyaml is MIT license, so it shouldn't impact Ruby's license.

Matz, may I embed libyaml for preview2?

### History

**#1 - 11/17/2012 11:06 AM - mame (Yusuke Endoh)**

*- Status changed from Open to Assigned*

*- Assignee set to matz (Yukihiro Matsumoto)*

*- Target version set to 2.0.0*

**#2 - 11/17/2012 11:23 AM - mame (Yusuke Endoh)**

Looks good.  Matz, could you accept this?

□□□□□□□□2.0.0 □ libyaml □□□□□□□□□□ (□□□ rubygems □□□□□) □
□□□□□□□□□□ FAQ □□□□□□□□
aaron □□□□□□□□□□libyaml □□□□□□□□□□□□□□ psych □□□□□□□□
□□□□□□□□□□□□□□□

libyaml □ MIT □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□

--
Yusuke Endoh mame@tsg.ne.jp

2012/11/17 tenderlovemaking (Aaron Patterson) aaron@tenderlovemaking.com:

> Issue #7375 has been reported by tenderlovemaking (Aaron Patterson).
>
> ---
>
> Feature #7375: embedding libyaml in psych for Ruby 2.0
> https://bugs.ruby-lang.org/issues/7375
>
> Author: tenderlovemaking (Aaron Patterson)
> Status: Open
> Priority: Normal
> Assignee:
> Category:
> Target version:
>
> People have a hard time using psych (and thus rubygems) because it depends on libyaml. We can ease upgrades by embedding libyaml in order to eliminate the library dependency.
>
> libyaml is MIT license, so it shouldn't impact Ruby's license.
>
> Matz, may I embed libyaml for preview2?
>
> --
> http://bugs.ruby-lang.org/

--
Yusuke Endoh mame@tsg.ne.jp

#### #3 - 11/18/2012 03:05 AM - vo.x (Vit Ondruch)

I hope this proposal will be rejected. Here are some reasons:

https://fedoraproject.org/wiki/Packaging:No_Bundled_Libraries
http://wiki.gentoo.org/wiki/Why_not_bundle_dependencies

And here are policies regarding bundling for several Linux distributions. All of them forbids bundling

https://fedoraproject.org/wiki/Packaging:Guidelines#Bundling_of_multiple_projects
http://www.debian.org/doc/debian-policy/ch-source.html#s-embeddedfiles
http://en.opensuse.org/openSUSE:Packaging_guidelines#Bundling_of_multiple_projects

#### #4 - 11/18/2012 03:23 AM - jeremyevans0 (Jeremy Evans)

vo.x (Vit Ondruch) wrote:

> I hope this proposal will be rejected. Here are some reasons:
>
> https://fedoraproject.org/wiki/Packaging:No_Bundled_Libraries
> http://wiki.gentoo.org/wiki/Why_not_bundle_dependencies
>
> And here are policies regarding bundling for several Linux distributions. All of them forbids bundling
>
> https://fedoraproject.org/wiki/Packaging:Guidelines#Bundling_of_multiple_projects
> http://www.debian.org/doc/debian-policy/ch-source.html#s-embeddedfiles
> http://en.opensuse.org/openSUSE:Packaging_guidelines#Bundling_of_multiple_projects

There are two fairly easy ways to fix that, hopefully Aaron will choose one of them:

1. Default to using the system implementation if present, using the embedded one only if no system implementation is found.
2. Default to using the embedded implementation, but have a separate configure flag for using the system implementation.

I'd prefer 1), since it doesn't change things for existing packagers, while making it easier on other people who build ruby from source.

#### #5 - 11/18/2012 05:23 AM - Anonymous

On Sun, Nov 18, 2012 at 03:05:50AM +0900, vo.x (Vit Ondruch) wrote:

> Issue #7375 has been updated by vo.x (Vit Ondruch).
>
> I hope this proposal will be rejected. Here are some reasons:
>
> https://fedoraproject.org/wiki/Packaging:No_Bundled_Libraries
> http://wiki.gentoo.org/wiki/Why_not_bundle_dependencies

Thanks for the links.  Unfortunately these lists of rules don't help
solve the problems I enumerated in the ticket.

> And here are policies regarding bundling for several Linux distributions. All of them forbids bundling
>
> https://fedoraproject.org/wiki/Packaging:Guidelines#Bundling_of_multiple_projects
> http://www.debian.org/doc/debian-policy/ch-source.html#s-embeddedfiles
> http://en.opensuse.org/openSUSE:Packaging_guidelines#Bundling_of_multiple_projects

Again, doesn't help with the problems we're having.

--
Aaron Patterson
http://tenderlovemaking.com/

#### #6 - 11/18/2012 05:23 AM - Anonymous

On Sun, Nov 18, 2012 at 03:23:49AM +0900, jeremyevans0 (Jeremy Evans) wrote:

> Issue #7375 has been updated by jeremyevans0 (Jeremy Evans).
> vo.x (Vit Ondruch) wrote:
>
> > I hope this proposal will be rejected. Here are some reasons:
> >
> > https://fedoraproject.org/wiki/Packaging:No_Bundled_Libraries
> > http://wiki.gentoo.org/wiki/Why_not_bundle_dependencies

> > And here are policies regarding bundling for several Linux distributions. All of them forbids bundling
> >
> > https://fedoraproject.org/wiki/Packaging:Guidelines#Bundling_of_multiple_projects
> > http://www.debian.org/doc/debian-policy/ch-source.html#s-embeddedfiles
> > http://en.opensuse.org/openSUSE:Packaging_guidelines#Bundling_of_multiple_projects
>
> There are two fairly easy ways to fix that, hopefully Aaron will choose one of them:
>
> 1. Default to using the system implementation if present, using the embedded one only if no system implementation is found.
> 2. Default to using the embedded implementation, but have a separate configure flag for using the system implementation.
>
> I'd prefer 1), since it doesn't change things for existing packagers, while making it easier on other people who build ruby from source.

#1 seems totally reasonable. I could do that pretty easily. :-)

--
Aaron Patterson
http://tenderlovemaking.com/

**#7 - 11/18/2012 05:53 AM - Anonymous**

Dne 17.11.2012 21:19, Aaron Patterson napsal(a):

> Again, doesn't help with the problems we're having.

I don't understand what problems. Was this issue triggered by
[ruby-core:49416]? May be there could be better error message then

/test_gem_command_manager.rb:
cannot load such file -- psych

Also the if the point

However, I'm able to build and install ruby
without libyaml-dev

made there is valid, how it comes that the build does not fail if the
libyaml is not present on the system, while it seems to be hard
dependency of some officially supported Ruby feature?

And what about educational factor? You bundle once and everybody will
think it is good practice, while it is not. Nobody learns nothing :/

Vit

**#8 - 11/18/2012 05:53 AM - Eregon (Benoit Daloze)**

On 17 November 2012 21:34, Vít Ondruch v.ondruch@gmail.com wrote:

> Dne 17.11.2012 21:19, Aaron Patterson napsal(a):
>
> > Again, doesn't help with the problems we're having.
>
> I don't understand what problems. Was this issue triggered by
> [ruby-core:49416]? May be there could be better error message then
>
> /test_gem_command_manager.rb:
> cannot load such file -- psych
>
> Also the if the point
>
> However, I'm able to build and install ruby
> without libyaml-dev
>
> made there is valid, how it comes that the build does not fail if the
> libyaml is not present on the system, while it seems to be hard dependency
> of some officially supported Ruby feature?
>
> Vit

Agreed, it does not make sense to me to bundle to avoid this problem.
They are already many other dependencies which are of course not bundled.
rvm requirements lists amongst others readline, iconv, zlib,
libyaml, libffi, openssl.
I would be for a failing build if libyaml is not present (except if
explicitly specified, like --disable-libyaml), or a very visible
warning at least, since it is a dependency of a standard library
always installed by default (rubygems).
I always thought missing extensions due to missing libraries were too
hard to notice in the build process, maybe a good summary of what was
built and what failed at the end would solve that?

#### #9 - 11/18/2012 11:53 PM - mame (Yusuke Endoh)

Hello,

2012/11/18 Benoit Daloze eregontp@gmail.com:

> Agreed, it does not make sense to me to bundle to avoid this problem.
> They are already many other dependencies which are of course not bundled.
> rvm requirements lists amongst others readline, iconv, zlib,
> libyaml, libffi, openssl.

There are three practical reasons why libyaml should be bundled:

- rubygems strongly depends on libyaml.
- libyaml is not so popular. compared to readline, zlib, openssl, etc.
  I think there are many platforms that has no libyaml.
- libyaml is actually so small.

I can understand the policy of the Linux distribution. But, Ruby source
distribution is mainly used for those who intend to use Ruby with no
packaing system.

Note that there is a precident: nkf stdlib actually bundles the whole
source code of nkf project.

Vit, can you create a distro package of Ruby depending on libyaml package,
rathar than bundling it, even if Ruby source distribution bundles libyaml?
I'm sorry to trouble you, but I would greatly appreciate your cooperation.

> On 17 November 2012 21:34, Vít Ondruch v.ondruch@gmail.com wrote:
>
>> I don't understand what problems. Was this issue triggered by
>> [ruby-core:49416]? May be there could be better error message then

BTW: The trigger is a draft of 2.0.0 upgrade notes that I'm writing:

http://bugs.ruby-lang.org/projects/ruby/wiki/200UpgradeNotesDraft

It shows a cumbersome process to install libyaml to use rubygems.
I asked Aaron to check the process, and he counter-proposed bundling
libyaml.

--
Yusuke Endoh mame@tsg.ne.jp

#### #10 - 11/19/2012 12:53 AM - Anonymous

Dne 18.11.2012 15:41, Yusuke Endoh napsal(a):

> Hello,
>
> 2012/11/18 Benoit Daloze eregontp@gmail.com:
>
>> Agreed, it does not make sense to me to bundle to avoid this problem.
>> They are already many other dependencies which are of course not bundled.
>> rvm requirements lists amongst others readline, iconv, zlib,
>> libyaml, libffi, openssl.
>> There are three practical reasons why libyaml should be bundled:

> - rubygems strongly depends on libyaml.

No doubt about it.

> - libyaml is not so popular. compared to readline, zlib, openssl, etc.
>   I think there are many platforms that has no libyaml.

How that come that somebody could not have libyaml on some platform but
would be able to compile libyaml bundled with psych? This argument seems
moot. Or are you going to fork libyaml because of that?

> - libyaml is actually so small.

It is even smaller if you don't bundle it and use the system one :)

> I can understand the policy of the Linux distribution. But, Ruby source
> distribution is mainly used for those who intend to use Ruby with no
> packaing system.

Who are you referring to? If there is somebody like that, he/she should
have probably some level of understanding, because it will be needed
sooner or later.

> Note that there is a precident: nkf stdlib actually bundles the whole
> source code of nkf project.

I'll definitely look at nkf. Thanks for the tip. Also note that there
are other bundled code, such as rubygems, rake, rdoc, oniguruma,
minitest. But I am not sure why Ruby should use bad examples as a
justification for bundling.

> Vit, can you create a distro package of Ruby depending on libyaml package,
> rathar than bundling it, even if Ruby source distribution bundles libyaml?
> I'm sorry to trouble you, but I would greatly appreciate your cooperation.

If the implementation will follow proposal 1) from Jeremy, then probably
quite easily. But this is not technical question, if I can or cannot.

> > On 17 November 2012 21:34, Vít Ondruch v.ondruch@gmail.com wrote:
> >
> > > I don't understand what problems. Was this issue triggered by
> > > [ruby-core:49416]? May be there could be better error message then
> > > BTW: The trigger is a draft of 2.0.0 upgrade notes that I'm writing:

Thank you for explanation :)

> http://bugs.ruby-lang.org/projects/ruby/wiki/200UpgradeNotesDraft

I like this guide

> It shows a cumbersome process to install libyaml to use rubygems.

It is as cumbersome as compile Ruby from sources. If you don't like
cumbersome processes, then there are packaging systems, rubyinstaller,
RVM ...

> I asked Aaron to check the process, and he counter-proposed bundling
> libyaml.

### #11 - 11/19/2012 02:53 AM - mame (Yusuke Endoh)

Hello Vit,

2012/11/19 Vít Ondruch v.ondruch@gmail.com:

> > I can understand the policy of the Linux distribution.  But, Ruby source
> > distribution is mainly used for those who intend to use Ruby with no
> > packaing system.
>
> Who are you referring to? If there is somebody like that, he/she should have
> probably some level of understanding, because it will be needed sooner or
> later.

As you might guess, I'm concerned about those who build Ruby source
tarball themselves.
Probably, they are not a majority.  But we hold them in high regard.
They are the first users for us.

Of course, many of them will address the dependency lack.
But we can easily fix the lack ourselves.  I don't like to waste
their time to perform the following useless process:

- build and install Ruby tarball,
- (after a bit) notice that gem does not work,
- find the cause and workaround with impatience,
- find, download, build and install libyaml, and
- rebuild and reinstall Ruby tarball again!

--
Yusuke Endoh mame@tsg.ne.jp

### #12 - 11/19/2012 06:23 AM - Anonymous

Dne 18.11.2012 18:37, Yusuke Endoh napsal(a):

> Hello Vit,
>
> 2012/11/19 Vít Ondruch v.ondruch@gmail.com:
>
> > > I can understand the policy of the Linux distribution.  But, Ruby source
> > > distribution is mainly used for those who intend to use Ruby with no
> > > packaing system.
> >
> > Who are you referring to? If there is somebody like that, he/she should have
> > probably some level of understanding, because it will be needed sooner or
> > later.
>
> As you might guess, I'm concerned about those who build Ruby source
> tarball themselves.

I appreciate that, thank you. I build Ruby from sources as well. Since I
am preparing packages for Fedora and Red Hat Enterprise Linux, I don't
have other option ;)

> Probably, they are not a majority.  But we hold them in high regard.
> They are the first users for us.

I build the trunk quite often to keep up with development and to try to
prevent changes like this.

> Of course, many of them will address the dependency lack.
> But we can easily fix the lack ourselves.  I don't like to waste
> their time to perform the following useless process:
>
> - build and install Ruby tarball,
> - (after a bit) notice that gem does not work,
> - find the cause and workaround with impatience,
> - find, download, build and install libyaml, and

You do it once and then you can build Ruby as many times as you want
without that penalty.

- rebuild and reinstall Ruby tarball again!

If the configuration would fail earlier (#7385), they would definitely
save time of build and installation.

But anyway. I'll try to refrain from further comments to this thread,
since I made all my points.

Vit

#### #13 - 11/19/2012 08:23 AM - Anonymous

On Mon, Nov 19, 2012 at 12:48:36AM +0900, Vít Ondruch wrote:

> Dne 18.11.2012 15:41, Yusuke Endoh napsal(a):

[snip]

>> Vit, can you create a distro package of Ruby depending on libyaml package,
>> rathar than bundling it, even if Ruby source distribution bundles libyaml?
>> I'm sorry to trouble you, but I would greatly appreciate your cooperation.
>
> If the implementation will follow proposal 1) from Jeremy, then
> probably quite easily. But this is not technical question, if I can
> or cannot.

I think following the proposal from Jeremy is the best compromise, and
in fact is quite easy to do. I've implemented it here:

https://github.com/tenderlove/psych/tree/embed

Specifically this commit:

https://github.com/tenderlove/psych/commit/3b066cf94a4f37e0395b2b761cda1862a13fcbd9

--
Aaron Patterson
http://tenderlovemaking.com/

#### #14 - 11/19/2012 12:54 PM - mame (Yusuke Endoh)

Hello Vit,

2012/11/19 Vít Ondruch v.ondruch@gmail.com:

> But anyway. I'll try to refrain from further comments to this thread, since
> I made all my points.

You might have a coronary if you saw trunk/LEGAL.
Not only nkf but also the Ruby core itself bundles, in whole or in part,
many source files of other projects.

--
Yusuke Endoh mame@tsg.ne.jp

#### #15 - 11/20/2012 02:58 PM - matz (Yukihiro Matsumoto)

My only concern for bundling third party library is that leaving behind unmaintained after years.
So give me a promise that Aaron will keep it for foreseeable future. Then I will trust him.

Matz.

#### #16 - 11/21/2012 10:23 AM - Anonymous

On Tue, Nov 20, 2012 at 02:58:58PM +0900, matz (Yukihiro Matsumoto) wrote:

> Issue #7375 has been updated by matz (Yukihiro Matsumoto).
>
> My only concern for bundling third party library is that leaving behind unmaintained after years.
> So give me a promise that Aaron will keep it for foreseeable future. Then I will trust him.

I'm not sure exactly what you're asking, but I have no plans to quit
maintenance for the foreseeable future.  :-)

--
Aaron Patterson
http://tenderlovemaking.com/

**#17 - 11/24/2012 11:32 AM - mame (Yusuke Endoh)**

*- Priority changed from Normal to 5*

Thanks matz!  Aaron, go ahead.

--
Yusuke Endoh mame@tsg.ne.jp

**#18 - 11/24/2012 02:20 PM - mame (Yusuke Endoh)**

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

**#19 - 11/29/2012 03:26 AM - tenderlovemaking (Aaron Patterson)**

*- Status changed from Assigned to Closed*

*- % Done changed from 0 to 100*

I fixed this in r37919, but it didn't seem to close.  I'm closing now.

**#20 - 11/29/2012 03:50 AM - zzak (zzak _)**

Aaron, you have to add the ticket in the message, like:
[Feature #7375]