Ruby trunk - Feature #2032

Change the license to "GPLv2+ or Ruby's original".

09/02/2009 05:44 PM - naruse (Yui NARUSE)

Status:	Closed	
Priority:	Normal	
Assignee:	matz (Yukihiro Matsumoto)	
Target version:	1.9.3	

Description

=begin
This is moved ticked from ruby-dev.
Original post and ticket is #2000 in English.

----- Original Post -----

Hello.

Recently readline 6.0 was released and its license was changed from
GPLv2+ (GPL version 2 and any later) to GPLv3+ [1][2]
Unfortunately Ruby's license is still under GPLv2 and Ruby's original license [3],
which is incompatible with GPLv3 [4]. So unless Ruby's license is changed
to "GPLv2+ or Ruby's original license" or so , Ruby's readline module cannot be shipped
any more. Note that "Ruby's original license" is regarded as incompatible with
GPL [5].

So please change the Ruby's license to GPLv3 (and GPLv2) compat.

[1] http://tiswww.case.edu/php/chet/readline/rltop.html
[2] https://www.redhat.com/archives/fedora-devel-list/2009-July/msg00192.html
[3] http://svn.ruby-lang.org/cgi-bin/viewvc.cgi/trunk/COPYING?view=co
[4] https://fedoraproject.org/wiki/Licensing#GPL_Compatibility_Matrix
[5] https://fedoraproject.org/wiki/Licensing
=end

Related issues:

Has duplicate Ruby trunk - Feature #2000: Change the license to "GPLv2+ or Ru...	**Closed**	**08/26/2009**
Has duplicate Ruby trunk - Feature #2983: Ruby (GPLv2 only) tries to link to ...	**Third Party's Issue**	**03/18/2010**
Is duplicate of Ruby trunk - Feature #2982: Ruby tries to link with both open...	**Third Party's Issue**	**03/18/2010**

History

#1 - 09/02/2009 09:02 PM - hramrach (Michal Suchanek)

=begin
2009/9/2 Yui NARUSE redmine@ruby-lang.org:

> Feature #2032: Change the license to "GPLv2+ or Ruby's original".
> http://redmine.ruby-lang.org/issues/show/2032
>
> Author: Yui NARUSE
> Status: Open, Priority: Normal
>
> This is moved ticked from ruby-dev.
> Original post and ticket is #2000 in English.
>
> ----- Original Post -----
>
> Hello.
>
> Recently readline 6.0 was released and its license was changed from
> GPLv2+ (GPL version 2 and any later) to GPLv3+ [1][2]
> Unfortunately Ruby's license is still under GPLv2 and Ruby's original license [3],
> which is incompatible with GPLv3 [4]. So unless Ruby's license is changed
> to "GPLv2+ or Ruby's original license" or so , Ruby's readline module cannot be shipped
> any more. Note that "Ruby's original license" is regarded as incompatible with
> GPL [5].

The source can still be shipped, and the binary module can be shipped
if compiled with readline before the license change. It is readline
changing license in an incompatible way, not Ruby. Such things happen,
and the solution is rarely relicensing the whole project.

There is also pure ruby readline which is perhaps becoming more relevant now.

> So please change the Ruby's license to GPLv3 (and GPLv2) compat.

I am not sure this is so easy as just changing a line somewhere.

You would likely need agreement from people who wrote different parts
of Ruby code.

Thanks

Michal

=end

#2 - 09/02/2009 11:06 PM - sandal (Gregory Brown)

=begin
On Wed, Sep 2, 2009 at 8:01 AM, Michal Suchanek hramrach@centrum.cz wrote:

> > So please change the Ruby's license to GPLv3 (and GPLv2) compat.

> I am not sure this is so easy as just changing a line somewhere.

> You would likely need agreement from people who wrote different parts
> of Ruby code.

A while ago, Ruby was changed from GPLv2+ to GPLv2 explicitly. This is
mainly to avoid giving the FSF a blank check for license changes.

Changing it to allow GPL v2 or v3 explicitly (*not* GPL 2+) might make
people happy and last for a long time, but as you said, contributors
of significant works would need to be contacted because AFAIK people
don't assign their copyrights to Matz when they contribute Ruby code.

But depending on how much readline is changing, it might be also okay
to go with the solution you suggested.

-greg

=end

#3 - 09/03/2009 10:18 AM - drbrain (Eric Hodel)

=begin
On my platforms (OS X and FreeBSD) there is no GNU readline, instead libedit is used which is not GPL licensed (no compatibility problem).

Why not change to a less restrictive license like MIT?
=end

#4 - 09/03/2009 05:36 PM - feyeleanor (Eleanor McHugh)

=begin
On 3 Sep 2009, at 02:18, Eric Hodel wrote:

> Issue #2032 has been updated by Eric Hodel.

> On my platforms (OS X and FreeBSD) there is no GNU readline, instead

> libedit is used which is not GPL licensed (no compatibility problem).

> Why not change to a less restrictive license like MIT?

Hasn't the Ruby License traditionally been an MIT-style license? Seems

a lot more appropriate for the reference implementation of a general

purpose programming language than the GPL.

Ellie

Eleanor McHugh
Games With Brains
http://slides.games-with-brains.net
———

raise ArgumentError unless @reality.responds_to? :reason

=end

#5 - 09/03/2009 10:05 PM - sandal (Gregory Brown)

=begin
On Thu, Sep 3, 2009 at 4:36 AM, Eleanor
McHugheleanor@games-with-brains.com wrote:

> On 3 Sep 2009, at 02:18, Eric Hodel wrote:
>
>> Issue #2032 has been updated by Eric Hodel.
>>
>> On my platforms (OS X and FreeBSD) there is no GNU readline, instead
>> libedit is used which is not GPL licensed (no compatibility problem).
>>
>> Why not change to a less restrictive license like MIT?
>
>
> Hasn't the Ruby License traditionally been an MIT-style license? Seems a lot
> more appropriate for the reference implementation of a general purpose
> programming language than the GPL.

The trouble is that the Ruby License (on its own) is not considered a
free software license by the FSF, which makes it incompatible with the
GPL.
The only thing that makes Ruby free software in the eyes of the FSF is
the fact that it is dual licensed.

Dual licensing is in some sense appealing because while consumers of
Ruby are free to do with it what they please, the maintainers are in
essence promising that you will always be able use the GPL for your
projects that extend core Ruby (as they're required to keep in
compliance with the GPL despite the dual license).

Of course, this is something that can be upheld even under an MIT
license, but it wouldn't be a guarantee in writing.   In some
communities, that's a sore spot for folks. In Ruby, I'm not sure that
it is, so a more simple arrangement might not be a bad thing.

-greg

=end

#6 - 09/04/2009 03:23 AM - spatulasnout (B Kelly)

=begin

From: "Gregory Brown" gregory.t.brown@gmail.com

> The trouble is that the Ruby License (on its own) is not considered a
> free software license by the FSF, which makes it incompatible with the
> GPL.
> The only thing that makes Ruby free software in the eyes of the FSF is
> the fact that it is dual licensed.

I tried to locate some information regarding the FSF
not considering the Ruby License a free software
license, but all I've found so far was this:

http://www.gnu.org/licenses/license-list.html#Ruby

Which says:

```
License of Ruby

This is a free software license ...
```

?

Regards,

Bill

=end

#7 - 09/04/2009 04:17 AM - sandal (Gregory Brown)

=begin
On Thu, Sep 3, 2009 at 2:23 PM, Bill Kellybillk@cts.com wrote:

> From: "Gregory Brown" gregory.t.brown@gmail.com
>
>> The trouble is that the Ruby License (on its own) is not considered a
>> free software license by the FSF, which makes it incompatible with the
>> GPL.
>> The only thing that makes Ruby free software in the eyes of the FSF is
>> the fact that it is dual licensed.
>
>
> I tried to locate some information regarding the FSF
> not considering the Ruby License a free software license, but all I've found
> so far was this:
>
> http://www.gnu.org/licenses/license-list.html#Ruby
>
> Which says:
> License of Ruby
>
> This is a free software license ...

You didn't finish the sentence:

"This is a free software license, compatible with the GPL **via an
explicit dual-licensing clause.**"

This is technically true of anything with an explicit dual license.
(Such as GPL or pay for license) models.
If you dual license with the GPL, and your codebase is GPL compliant,
the other license could be "You must ride a pony while using this
software OR abide by the GPL" and it'd be valid.

To be fair, there isn't anything except for wording that might get in
the way of Matz's terms being considered a free software license.
And for a bit more disclosure, I had asked this question a while back
to Richard Stallman directly in an email, so... I'm not sure it has
been discussed extensively beyond that.

-greg

=end

#8 - 09/04/2009 04:30 AM - spatulasnout (B Kelly)

=begin

From: "Gregory Brown" gregory.t.brown@gmail.com

> On Thu, Sep 3, 2009 at 2:23 PM, Bill Kellybillk@cts.com wrote:
>
>> I tried to locate some information regarding the FSF
>> not considering the Ruby License a free software license,
>> but all I've found so far was this:
>>
>> http://www.gnu.org/licenses/license-list.html#Ruby

Which says:
License of Ruby

This is a free software license ...

You didn't finish the sentence:

"This is a free software license, compatible with the GPL
via an explicit dual-licensing clause."

I may be misunderstanding what is being discussed. I didn't
finish the sentence deliberately, because GPL compatibility
is a different concept than something being a free software
license.

Numerous free software licenses are not GPL-compatible.

You stated, the Ruby License "is not considered a free
software license by the FSF".

That piqued my interest, as I don't understand why the
Ruby license would not be considered a free software
license.

Regards,

Bill

=end

#9 - 09/04/2009 07:25 AM - sandal (Gregory Brown)

=begin
On Thu, Sep 3, 2009 at 3:29 PM, Bill Kelly billk@cts.com wrote:

From: "Gregory Brown" gregory.t.brown@gmail.com

On Thu, Sep 3, 2009 at 2:23 PM, Bill Kelly billk@cts.com wrote:

I tried to locate some information regarding the FSF
not considering the Ruby License a free software license,
but all I've found so far was this:

http://www.gnu.org/licenses/license-list.html#Ruby

Which says:
License of Ruby

This is a free software license ...

You didn't finish the sentence:

"This is a free software license, compatible with the GPL
via an explicit dual-licensing clause."

I may be misunderstanding what is being discussed. I didn't
finish the sentence deliberately, because GPL compatibility
is a different concept than something being a free software
license.

Numerous free software licenses are not GPL-compatible.

You stated, the Ruby License "is not considered a free
software license by the FSF".

That piqued my interest, as I don't understand why the
Ruby license would not be considered a free software
license.

I can try to get another email by the Stallman-to-real-world compiler,

or maybe we could ask the FSF Licensing and Compliance Lab?

http://www.fsf.org/licensing/

I think it is because Matz's terms do not explicitly grant the freedom
to sell the software, even though his terms implicitly say they may be
used for 'possibly commercial' means. The FSF can be pretty
aggressive about their interpretations of things...

But I could have misunderstood their stance here, so in either case,
if you want me to try to get a more definite answer, just let me know.
FWIW, I was suggesting we should either a) Add explicit GPL 2 or 3
terms to the current license, or b) drop it entirely in favor of
MIT/BSD/etc.

I don't think that c) is ever going to be a good option (Just use Matz's terms).

-greg

=end

#10 - 09/04/2009 06:38 PM - shyouhei (Shyouhei Urabe)

=begin
My understanding is that if ruby links to readline6, not only ruby itself is "tainted" by GPLv3, but also other libraries that ruby links, such as openssl,
zlib, mysql adaper, are also tainted. I don't think it being realistic for library authors such as openssl people to accept that situation.
=end

#11 - 09/05/2009 01:24 AM - sandal (Gregory Brown)

=begin
On Fri, Sep 4, 2009 at 5:38 AM, Shyouhei Uraberedmine@ruby-lang.org wrote:

> Issue #2032 has been updated by Shyouhei Urabe.
>
> My understanding is that if ruby links to readline6, not only ruby itself is "tainted" by GPLv3, but also other libraries that ruby links, such as
> openssl, zlib, mysql adaper, are also tainted. I don't think it being realistic for library authors such as openssl people to accept that situation.

I think that is correct. But this is true of using any GPL2 library
as well. Unless you make the feature optional and isolated from the
rest of the code, it will cause Ruby to in effect, be GPLed. (And not
simply dual licensed).

AFAIK, the general rule of thumb is that dual licensing gives a choice
to consumers, but project maintainers must abide by all of their
licenses simultaneously.
Linking to GPLed code prevents Matz's terms from actually being an
option in that case.

But using a GPL3 library is problematic for a different reason. It is
not compatible with GPL2, nor is it compatible with Matz's license.
:-/
So essentially, adding a "Matz's Terms, GPL2, or GPL3" clause would
only allow people extending Ruby to choose to GPL3 their code, not
allow Ruby itself to link to GPL3 code.

(Hope my understanding is correct here, open to hearing from those who
know more about this)

-greg

=end

#12 - 09/05/2009 02:10 AM - kazuhiko (Kazuhiko Shiozaki)

=begin
greg:

> But using a GPL3 library is problematic for a different reason. It is
> not compatible with GPL2, nor is it compatible with Matz's license.

Exactly.
If we want to link GPL3 (or later :p) library with Ruby, we have following

possibilities.

- change Ruby's license to "(GPL2 or later) or Matz L."
- change Ruby's license to "(GPL3 or later) or Matz L."
- change Ruby's license to any GPL compatible license, eg. MIT L., BSD L. etc.

The followings are what Matz said, FYI (translated by me):

- http://www.rubyist.net/~matz/20030607.html#p06

    I'm planning to use the following license for RITE:
    /*

    - Copyright (C) 2003 Yukihiro "matz" Matsumoto *
    - Permission is granted for use, copying, modification, distribution,
    - and distribution of modified versions of this work as long as the
    - above copyright notice is included. */

- http://www.rubyist.net/~matz/20030608.html#p02

    When I defined Ruby's license, I wanted to permit reuse of a part of codes
    explicitly. But in reality, BSD license or GPL also fulfills my desire enough
    (though not so explicitly).

According to his words, I guess that MIT (or BSD) license can be a good choice,
if we change the license.

And as Shyouhei said, I am afraid that the following can have a license problem.
$ ruby -r readline -r openssl
Because ...

- OpenSSL is not GPL-compatible, so we need to choose Matz L. for '-r openssl'.
- Matz L. is not GPL-compatible, so we need to choose GPL for '-r readline'. This problem exists even with GPL2 readline, even if we change the license, unless we make an GPL-compatibile alternative for ruby-openssl (using GNU TLS for example).

BTW, 'loading ruby script' itself is not 'link', in FSF's statement.
See http://www.gnu.org/copyleft/gpl-faq.html#IfInterpreterIsGPL for the
detail.

Kazuhiko
=end

#13 - 10/28/2009 02:45 AM - naruse (Yui NARUSE)

- Status changed from Open to Assigned

- Assignee set to matz (Yukihiro Matsumoto)

- Priority changed from Normal to 5

=begin

=end

#14 - 03/18/2010 06:48 PM - naruse (Yui NARUSE)

- Target version set to 1.9.2

=begin

=end

#15 - 03/19/2010 01:41 PM - nobu (Nobuyoshi Nakada)

- Status changed from Assigned to Closed

=begin

=end

#16 - 03/20/2010 05:06 AM - naruse (Yui NARUSE)

- Status changed from Closed to Assigned

=begin

=end

#17 - 03/20/2010 08:35 AM - nobu (Nobuyoshi Nakada)

- Status changed from Assigned to Closed

=begin
I close this ticket because it is duplication.
Also, note that libedit has been supported already,
libreadline will be disabled by --enable-libedit configure option.
=end

#18 - 03/20/2010 07:11 PM - naruse (Yui NARUSE)

- Status changed from Closed to Assigned

=begin

=end

#19 - 03/21/2010 09:04 AM - nobu (Nobuyoshi Nakada)

- Status changed from Assigned to Third Party's Issue

=begin

=end

#20 - 04/16/2010 09:43 AM - naruse (Yui NARUSE)

- Status changed from Third Party's Issue to Assigned

=begin
Matz, you decided that MRI can't link to readline 6, right?
If so, I'll merge nobu's patch showed in [ruby-dev:39172]
=end

#21 - 04/17/2010 02:25 AM - matz (Yukihiro Matsumoto)

=begin
Hi,

In message "Re: [ruby-core:29545] Feature #2032 Change the license to "GPLv2+ or Ruby's original"."
on Fri, 16 Apr 2010 09:44:00 +0900, Yui NARUSE redmine@ruby-lang.org writes:

|Matz, you decided that MRI can't link to readline 6, right?
|If so, I'll merge nobu's patch showed in [ruby-dev:39172]

I agree with merging a patch in [ruby-dev:39172]. So please check in.
1.9.2 might be a good time to restore "or later" phrase. Any opinion?

                                matz.

=end

#22 - 04/17/2010 07:46 AM - shyouhei (Shyouhei Urabe)

=begin
Yukihiro Matsumoto wrote:

    1.9.2 might be a good time to restore "or later" phrase. Any opinion?

It's not a matter of timing I'm afraid. So -1 without any other good reasons
that we can agree.

Attachment: signature.asc
=end

#23 - 06/01/2010 03:57 AM - naruse (Yui NARUSE)

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

=begin
This issue was solved with changeset r28118.
Yui, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

=end

#24 - 06/01/2010 10:33 AM - naruse (Yui NARUSE)

=begin
I merged a patch in [ruby-dev:39172]: reject readline6.
If someone want to change the license of Ruby and work about it, please reopen this.
=end

#25 - 06/06/2010 01:37 PM - mame (Yusuke Endoh)

- Status changed from Closed to Assigned

- Priority changed from 5 to 6

=begin
Hi,

     I merged a patch in [ruby-dev:39172]: reject readline6.

I really hate the check that rejects linking with readline6,
because it is actually *new feature*. The 1.9.2 feature was
already frozen. Rather, preview3 was even released.
I'm not against the feature itself, but now is a bad time.
I'll revert in a few days.

Of course, the license issue still remains. But it is enough
to just add a note to document. I think we have no obligation
to add such a mechanism to check readline6 and to reject its
link.

In addition, I suggest to change Ruby's license to triple
(Matz's, GPLv2 and GPLv3) right now. I think no one objects
GPLv3.
The current license will makes us unhappy because we cannot use
not only readline6 but also other GPLv3 libraries that will be
increased.
We can discuss adding "or later" clause or changing to BSDL
after 1.9.2 is released.

Matz, how about the triple license?

--
Yusuke Endoh mame@tsg.ne.jp
=end

#26 - 06/06/2010 04:58 PM - shyouhei (Shyouhei Urabe)

=begin
Yusuke, I know your feeling so I don't want to attack you, but your
suggestion is just an yet another new feature -- as long as you see
licensing issues as features (rather than bugs of contractual kind).
=end

#27 - 06/06/2010 05:40 PM - mame (Yusuke Endoh)

=begin
Hi,

2010/6/6 Shyouhei Urabe redmine@ruby-lang.org:

     Yusuke, I know your feeling so I don't want to attack you, but your

suggestion is just an yet another new feature -- as long as you see
licensing issues as features (rather than bugs of contractual kind).

I see this licensing issue as a bug.
But I consider "the mechanism to reject readline6" as a new feature
because there is another solution.

If the mechanism is the only solution, we can't help but include the
mechanism in 1.9.2.
But we can solve this kind of issue with modifying no code; adding a
note, changing the license, etc.

To avoid enbugging a new bug, we must choose the another solutions.
In this case, it is enough to add a note, "DO NOT USE etc/readline
WITH readline6!", to README.
The triple license would be the best solution as far as I can see,
unless it is rejected by matz.

--
Yusuke Endoh mame@tsg.ne.jp

=end

#28 - 06/06/2010 06:42 PM - shyouhei (Shyouhei Urabe)

=begin

> To avoid enbugging a new bug, we must choose the another solutions.
> In this case, it is enough to add a note, "DO NOT USE etc/readline
> WITH readline6!", to README.

? Sorry, can't interpret it. what's a new bug? Why is README is enough?

=end

#29 - 06/06/2010 08:27 PM - mame (Yusuke Endoh)

=begin
2010/6/6 Urabe Shyouhei shyouhei@ruby-lang.org:

>> To avoid enbugging a new bug, we must choose the another solutions.
>> In this case, it is enough to add a note, "DO NOT USE etc/readline
>> WITH readline6!", to README.

> ? Sorry, can't interpret it. what's a new bug? ?Why is README is enough?

Any code modification may often make a new bug even if it looks like
trivial one. We should select a solution with no code modification
if any, especially during release process.

By adding a note to README, we can explicitly state that ext/readline
cannot be linked with readline6. Why do you think it is not enough?

--
Yusuke Endoh mame@tsg.ne.jp

=end

#30 - 06/08/2010 01:55 PM - shyouhei (Shyouhei Urabe)

=begin
(2010/06/06 20:27), Yusuke ENDOH wrote:

> Any code modification may often make a new bug even if it looks like
> trivial one. We should select a solution with no code modification
> if any, especially during release process.

You are speaking too generally. Where's the bug you are talking about?

> By adding a note to README, we can explicitly state that ext/readline

cannot be linked with readline6. Why do you think it is not enough?

Because people can ignore such statements. Machine checking is stronger than a
mare readme.

But OK, people's ignoring license agreements is definitely not our fault. If
you think we should not care about that situation, I can accept that. Though I
don't think the status quo is a problem.

=end

#31 - 06/08/2010 11:12 PM - mame (Yusuke Endoh)

=begin
Hi,

2010/6/8 Urabe Shyouhei shyouhei@ruby-lang.org:

> (2010/06/06 20:27), Yusuke ENDOH wrote:
>
>> Any code modification may often make a new bug even if it looks like
>> trivial one. We should select a solution with no code modification
>> if any, especially during release process.
>
>
> You are speaking too generally. Where's the bug you are talking about?

We don't know *yet* where the bug is. It is the problem.
Concretely, I'm concerned that the check causes false positive or
build error in some peculiar environment or configuration.

>> By adding a note to README, we can explicitly state that ext/readline
>> cannot be linked with readline6. Why do you think it is not enough?
>
>
> Because people can ignore such statements. Machine checking is stronger than a
> mare readme.

Sure. I agree with the check itself. It is nice feature.
But new feature should not be included in 1.9.2.

Though I think "or GPLv3" is much better if matz allows,
I'll commit the following patch unless there is objection:

diff --git a/README b/README
index 3d2e636..4a032c4 100644
--- a/README
+++ b/README
@@ -120,6 +120,10 @@ This is what you need to do to compile and install Ruby:
If you fail to compile ruby, please send the detailed error report with
the error log and machine/OS type, to help others.

+DO NOT LINK Ruby WITH A LIBRARY WHOSE LICENSE IS NOT COMPATIBLE WITH Ruby
+License (see COPYING).
+For example, currently you cannot link with readline6, which is under GPLv3.
+

  * Copying

--
Yusuke Endoh mame@tsg.ne.jp

=end

#32 - 06/15/2010 01:25 AM - meta (mathew murphy)

=begin
On Tue, Jun 8, 2010 at 09:12, Yusuke ENDOH mame@tsg.ne.jp wrote:

> Though I think "or GPLv3" is much better if matz allows,
> I'll commit the following patch unless there is objection:

[...]

> +DO NOT LINK Ruby WITH A LIBRARY WHOSE LICENSE IS NOT COMPATIBLE WITH Ruby
> +License (see COPYING).
> +For example, currently you cannot link with readline6, which is under GPLv3.
> +

The wording here is incorrect, and I think reflects a misunderstanding
of the GPL (and GPL v3).

The GNU Public License is *not* an EULA. You do not have to agree to
it in order to download and use GPL-licensed software. As David Jones
points out at http://drj11.wordpress.com/2008/02/25/the-gnu-gpl-is-not-an-eula/,
the license itself states that "Activities other than copying,
distribution and modification are not covered by this License; they
are outside its scope. The act of running the Program is not
restricted".

So it is perfectly legal for me to link Ruby with any number of GPL3
libraries, including readline6, and to use that copy of Ruby as much
as I wish. The only thing I cannot do is *distribute* a version of
Ruby which has been linked with readline6.

In addition, I cannot distribute Ruby linked with a library unless I
comply with all of the license terms. So libraries I link and
distribute must be compatible with *both* the GPL and the alternative
Matz license terms in http://www.ruby-lang.org/en/LICENSE.txt

This means that you can't link and distribute GPL2 libraries under the
Ruby license either, so you can't use versions of readline prior to
readline6 either.

So the wording should be more like this:

DO NOT DISTRIBUTE Ruby WITH A LIBRARY WHOSE LICENSE IS NOT
COMPATIBLE WITH EITHER SET OF Ruby LICENSE TERMS (see COPYING).

For example, currently you cannot distribute Ruby linked with any
version of GNU readline, under the original Ruby license terms; you
must distribute Ruby linked with readline under the GPL. You cannot
distribute Ruby linked with readline6 at all, because it uses GPL v3,
and Ruby is not yet approved for GPLv3 licensing.

You can link Ruby with any GPL or GPLv3 licensed software and use it;
you just may not be able to distribute it if you do so.

---

mathew
--
URL:http://www.pobox.com/~meta/

=end

#33 - 06/15/2010 12:48 PM - mame (Yusuke Endoh)

=begin
Hi,

2010/6/15 mathew meta@pobox.com:

> The GNU Public License is *not* an EULA. You do not have to agree to
> it in order to download and use GPL-licensed software. As David Jones
> points out at http://drj11.wordpress.com/2008/02/25/the-gnu-gpl-is-not-an-eula/,
> the license itself states that "Activities other than copying,
> distribution and modification are not covered by this License; they
> are outside its scope. The act of running the Program is not
> restricted".

I think you are right in your perspective.
But the problem is, "can the core team distribute Ruby?"

We actually distribute Ruby with readline binding.
A user may link the binding with readline6, which we do not intend.
If it is allowed, we can distribute a product using GPLv3 library
with license that is imcompatible with GPLv3, I guess.
Even so, can we distribute Ruby without the note prohibiting such
a link?

IMHO, I agree with your opinion, but I'm not a lawyer. I cannot
determine. License issue sucks.

--
Yusuke Endoh mame@tsg.ne.jp

=end

#34 - 06/16/2010 11:48 AM - meta (mathew murphy)

=begin
On Jun 14, 2010, at 22:48, Yusuke ENDOH wrote:

> I think you are right in your perspective.
> But the problem is, "can the core team distribute Ruby?"
>
> We actually distribute Ruby with readline binding.
> A user may link the binding with readline6, which we do not intend.
> If it is allowed, we can distribute a product using GPLv3 library
> with license that is imcompatible with GPLv3, I guess.
> Even so, can we distribute Ruby without the note prohibiting such
> a link?

I don't believe there's any legal precedent to determine this.

The FSF believes that if your code requires that a GPL library be linked with it, then your code is derivative of the GPL library, and must also fall under the GPL.

However, I haven't seen a FSF statement as to whether distributing GPL2 code that links with GPL3 code is a violation.

My take is that so long as the code is written to be linked with readline5 and only tested and supported with readline5, that's what it's a derivative of. Just state that clearly where the list of required libraries is in the documentation,.

If the developers of readline6 have made that library compatible enough with earlier versions that it can be used as a substitute, that's up to them, but you're only required to fall under the same license as the library you're intentionally using. You can't be held to a new license just because someone decides to substitute readline5 with something else that's API-compatible but has a new license.

Having said all of that... I'd welcome Ruby being released under GPL v3. If that cannot be done, or is not deemed desirable by Matz, then the best option is probably to start migrating from readline to libeditline:
http://s11n.net/editline/

mathew

=end

#35 - 06/16/2010 02:44 PM - naruse (Yui NARUSE)

=begin
2010/6/16 Monty Williams monty@vmware.com:

> On 6/15/10 7:48 PM, mathew wrote:
>
>> Having said all of that... I'd welcome Ruby being released under GPL v3.
>> If that cannot be done, or is not deemed desirable by Matz, then the best
>> option is probably to start migrating from readline to libeditline:
>> http://s11n.net/editline/
>
>
> Or even linenoise. A minimal, zero-config, BSD licensed, readline
> replacement.
> http://github.com/antirez/linenoise

Both of them are only support ASCII.

--
NARUSE, Yui

=end

#36 - 06/17/2010 04:25 AM - mame (Yusuke Endoh)

=begin
Hi,

2010/6/16 mathew meta@pobox.com:

> However, I haven't seen a FSF statement as to whether distributing GPL2 code that links with GPL3 code is a violation.

http://www.gnu.org/licenses/gpl-faq.html#v2v3Compatibility

> Is GPLv3 compatible with GPLv2?
>
> No. Some of the requirements in GPLv3, such as the requirement
> to provide Installation Information, do not exist in GPLv2. As
> a result, the licenses are not compatible: if you tried to
> combine code released under both these licenses, you would
> violate section 6 of GPLv2.
>
> My take is that so long as the code is written to be linked with readline5 and only tested and supported with readline5, that's what it's a derivative
> of. Just state that clearly where the list of required libraries is in the documentation,.
>
> If the developers of readline6 have made that library compatible enough with earlier versions that it can be used as a substitute, that's up to
> them, but you're only required to fall under the same license as the library you're intentionally using. You can't be held to a new license just
> because someone decides to substitute readline5 with something else that's API-compatible but has a new license.

I entirely agree with you. We have no legal responsibility to care
readline6, I believe.

However, some people (including Matz [ruby-dev:39178] and Shyouhei
Urabe [ruby-core:30652]) think that it is "dishonest" to get along
with the status quo after we found out about readline6.
I don't deny that we have "moral" responsibility about readline6.

But now, I'm not sure if the note about readline6 really complies
with Ruby License, especially the section 6 of GPLv2:

> You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

The note restricts the link with readline6. Isn't it "any further
restrictions"? I don't know. Shyouhei, let me ask your opinion.

--
Yusuke Endoh mame@tsg.ne.jp

=end

#37 - 06/17/2010 03:19 PM - lucas (Lucas Nussbaum)

=begin
On 16/06/10 at 11:48 +0900, mathew wrote:

> However, I haven't seen a FSF statement as to whether distributing GPL2 code that links with GPL3 code is a violation.

I don't think it requires an FSF statement. It's obvious: if your code
is GPLv2-only (like Ruby), you can't link to GPLv3 code and distribute
the resulting binary. If your code is GPLv2+, you are permitted to do
so, but the resulting binary is distributed under the terms of GPLv3,
not GPLv2 or GPLv3.

What makes the situation even more "interesting" in the Ruby case is
that the ruby binary itself or the libruby shared lib, are not linked
to readline directly. Instead, you load a .so using 'require' if you
want to add readline support.
Debian's point of view on that, which is very pragmatic (possibly too
much), is that it's fine to distribute readline.so linked to readline5,
and that the resulting binary is only "contaminated" by GPL when you

load readline.so. That makes it possible for the Debian package to ship
support for both readline and openssl: what is forbidden is to ship a
binary (or, probably, a script), that requires both modules loaded at
the same time.
Now, this approach is very liberal, and I'm not sure the FSF would
agree.

Maybe the best thing to do would be to contact the FSF lawyers to ask
them for feedback?
--
| Lucas Nussbaum
| lucas@lucas-nussbaum.nethttp://www.lucas-nussbaum.net/ |
| jabber: lucas@nussbaum.fr   GPG: 1024D/023B3F4F |

=end

#38 - 06/18/2010 12:57 AM - shyouhei (Shyouhei Urabe)

=begin
(2010/06/17 4:25), Yusuke ENDOH wrote:

> Is GPLv3 compatible with GPLv2?
>
> No.

> The note restricts the link with readline6. Isn't it "any further
> restrictions"? I don't know. Shyouhei, let me ask your opinion.

As you quote GPLv2 vs GPLv3 are incompatible already, so the note do not impose
any *further* restrictions than what are there now. At least I believe so.
This is not a restriction; a fact.

=end

#39 - 09/16/2010 12:37 AM - naruse (Yui NARUSE)

- Status changed from Assigned to Closed

=begin
This issue was solved with changeset r29262.
Yui, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

=end

#40 - 04/27/2011 10:23 AM - naruse (Yui NARUSE)

- Target version changed from 1.9.2 to 1.9.3

=begin

=end

#41 - 04/27/2011 01:15 PM - headius (Charles Nutter)

=begin
Is shipping the "editline" library instead an option? It is a BSD-licensed, drop-in replacement for readline.
=end

#42 - 04/27/2011 01:16 PM - headius (Charles Nutter)

=begin
Nevermind, my browser showed an old version of this bug. I see editline was discussed already.
=end