

Ruby master - Misc #9215

Maintenance Policy for Future Releases (2.1.0 & beyond)

12/05/2013 12:07 PM - hone (Terence Lee)

Status:	Closed
Priority:	Normal
Assignee:	zzak (Zachary Scott)
Description	
In order to support long lived applications better, people need information to make decisions. When someone chooses to use a particular programming language it's important to know how long something is going to be supported.	
For future releases it'd be great if we could provide a formal end of life window upon release. This would allow companies to be able to make decisions on how long something will be supported. For instance, when Ruby 2.1.0 comes out this Christmas, giving an expectation that support will be stopped by 12/25/2015 gives an accurate picture of how much time must be allocated to upgrading, how urgent it will be, or if Ruby is even the right language for the project.	

History

#1 - 12/10/2013 05:16 AM - zzak (Zachary Scott)

- Category changed from Project to doc
- Status changed from Open to Assigned
- Assignee set to zzak (Zachary Scott)

We have decided the following:

- MINOR level versions of Ruby have a "best effort" goal of at least 6 months and at best 3 years.
- TEENY versions will be released approx. every 2-3 months
- Branch maintainers are not obligated to commit 3 years maintenance
- ruby-core as a team is responsible for a proper EOL for each MINOR version

Our main goal is to avoid a sudden death

Assigning this ticket to myself in order to document the approved maintenance plan.

#2 - 01/13/2014 04:14 AM - zzak (Zachary Scott)

I've written a draft of the [future maintenance policy](#).

Please check it.

#3 - 01/13/2014 04:20 AM - zzak (Zachary Scott)

- Status changed from Assigned to Feedback

I will give some time for feedback before closing this and opening a Pull Request on the upstream www.ruby-lang.org repository.

#4 - 01/18/2014 07:48 AM - naruse (Yui NARUSE)

- Who is "we"?
- It doesn't define what is "support"
- "at least 6 months and at best 3 years" is the conclusion. it is at least X years or after two newer release, in my understanding
- "TEENY versions will be released approx. every 2-3 months" should be 2-4 months. it also needs how ruby-core treat older teeny (if this is announce)
- How to finish a release maintenance is essential information. an announce must have it.

Anyway decisions on Ruby development is based on consensus building.

A decision must be implicitly/explicitly approved by persons concerned in the decision.

Persons concerned in the maintenance policy is naruse, usa, nagachika.

At least you get explicit approval by them. (idealy a comment on redmine or reply to ML)

A meeting is not enough. It means only sharing information)

#5 - 01/20/2014 04:44 AM - zzak (Zachary Scott)

[naruse \(Yui NARUSE\)](#) thank you for the review!

I've made the changes you listed above in the following pull request: <https://github.com/ruby/www.ruby-lang.org/pull/602>

#6 - 01/20/2014 06:43 AM - naruse (Yui NARUSE)

We'd like to take a minute and discuss our plans for ruby-core supported maintenance periods of Ruby.

"take a minute and discuss" is not useful for users. It should be simply "We release" or "decide" or something.

backports will be made to the ruby_MAJOR_MINOR branch

this information is not for users.

The ruby-core team is responsible for a proper End-of-Life for each MINOR version of Ruby.

Our current format is to assign a maintainer for each MINOR series of Ruby. It's important to note that this person is not obligated to commit for the entire 3 year suggested maintenance period.

Our main goal to avoid sudden death.

What do you want to say in these paragraph?

If this is announce, the audience is users.

They read this, "We want to avoid sudden death".

"ah, ok... and what ruby-core will do?"

The content is just a memorandum.

See also rails' maintenance policy: <http://weblog.rubyonrails.org/2013/2/24/maintenance-policy-for-ruby-on-rails/> or you should see other policies.

Any decision made on Ruby development is based on the consensus of ruby-core as a team. The decision must be implicitly or explicitly approved by members of the maintenance policy team:

This is not for users, just for you, zzak.

#7 - 01/20/2014 02:23 PM - duerst (Martin Dürst)

Hello Zachary,

From the comments from Yui, it looks like the policy should be split up into an internal (for Ruby committers, and in particular those in charge of releases,...) and an external (for users) part.

Also, please note that because Ruby isn't a company, we have to be careful with commitments. It's clear that for Ruby users, the more and the more explicit commitments there are, the better. But on the other hand, everybody, in particular also people in charge of releases, are just volunteers. They may be able to use some of their company's time, but that may change. They may use some of their free time, but that may also change.

So with respect to release and support policy, it's very important to get the okay of the actual person in charge. Matz may be okay with some release or support policy, but if the actual person in charge isn't okay with it, that won't help.

Of course it's always possible that companies explicitly commit support for some version of Ruby (that has e.g. happened for Ruby 1.8.7), and I think it would be good if any related document such as a release policy would mention this possibility.

Regards, Martin.

On 2014/01/20 15:43, naruse@airemix.jp wrote:

Issue [#9215](#) has been updated by Yui NARUSE.

We'd like to take a minute and discuss our plans for ruby-core supported maintenance periods of Ruby.

"take a minute and discuss" is not useful for users. It should be simply "We release" or "decide" or something.

backports will be made to the ruby_MAJOR_MINOR branch

this information is not for users.

The ruby-core team is responsible for a proper End-of-Life for each MINOR version of Ruby.

Our current format is to assign a maintainer for each MINOR series of Ruby. It's important to note that this person is not obligated to commit for the entire 3 year suggested maintenance period.

Our main goal to avoid sudden death.

What do you want to say in these paragraph?

If this is announce, the audience is users.

They read this, "We want to avoid sudden death".

"ah, ok... and what ruby-core will do?"

The content is just a memorandum.

See also rails' maintenance policy: <http://weblog.rubyonrails.org/2013/2/24/maintenance-policy-for-ruby-on-rails/> or you should see other policies.

Any decision made on Ruby development is based on the consensus of ruby-core as a team. The decision must be implicitly or explicitly approved by members of the maintenance policy team:

This is not for users, just for you, zzak.

misc [#9215](#): Maintenance Policy for Future Releases (2.1.0& beyond)
<https://bugs.ruby-lang.org/issues/9215#change-44446>

- Author: Terence Lee
- Status: Feedback
- Priority: Normal
- Assignee: Zachary Scott
- Category: doc

* Target version: 2.1.0

In order to support long lived applications better, people need information to make decisions. When someone chooses to use a particular programming language it's important to know how long something is going to be supported.

For future releases it'd be great if we could provide a formal end of life window upon release. This would allow companies to be able to make decisions on how long something will be supported. For instance, when Ruby 2.1.0 comes out this Christmas, giving an expectation that support will be stopped by 12/25/2015 gives an accurate picture of how much time must be allocated to upgrading, how urgent it will be, or if Ruby is even the right language for the project.

#8 - 01/21/2014 07:40 AM - naruse (Yui NARUSE)

Thank you for clearing out what I want to say.
And split up internal/external sounds good idea.
(both document will be public)

#9 - 01/21/2014 09:32 AM - zzak (Zachary Scott)

- Status changed from Feedback to Assigned

Thank you both for the feedback, I'm going to apply your suggestions and ping back when its ready.

#10 - 01/30/2014 06:17 AM - hsbt (Hiroshi SHIBATA)

- Target version changed from 2.1.0 to 2.2.0

#11 - 01/31/2014 08:53 PM - zzak (Zachary Scott)

I've updated the pull request given Martin and narsue's feedback. Please check it.

https://github.com/zzak/www.ruby-lang.org/blob/future_release_policy/en/news/_posts/2014-01-13-approved-maintenance-policy.md

My plan is to write a separate document in the redmine wiki for internal policy.

#12 - 02/14/2014 09:20 AM - naruse (Yui NARUSE)

Zachary Scott wrote:

I've updated the pull request given Martin and narsue's feedback. Please check it.

https://github.com/zzak/www.ruby-lang.org/blob/future_release_policy/en/news/_posts/2014-01-13-approved-maintenance-policy.md

My plan is to write a separate document in the redmine wiki for internal policy.

We've decided our plans for ruby-core supported maintenance periods of Ruby.

Who are "We"?

For context, you should read our recent policy changes for semantic versioning.

Ruby's versioning policy is not correct "semantic versioning" because Ruby breaks compatibility in minor. Don't use this word without detailed notes.

An approved commercial party may be able to accept responsibility of maintaining an End-of-Life version.

This sentence should be removed.

It's important to recognize the maintainer for each version may decide they can no longer support a version, in which case ruby-core would work to responsibly discontinue the version.

useless sentence

#13 - 02/15/2014 11:59 AM - zzak (Zachary Scott)

We've decided our plans for ruby-core supported maintenance periods of Ruby.

Who are "We"?

Who should "we" be? As an official post on ruby-lang.org, it must refer to ruby-core? Should it read:

The ruby-core team has decided plans for supported maintenance periods of Ruby.

How is that?

Ruby's versioning policy is not correct "semantic versioning".

So just rephrasing this to:

For context, you should read about our recent changes to the Ruby versioning policy."

Furthermore:

An approved commercial party may be able to accept responsibility of maintaining an End-of-Life version.

This sentence should be removed.

And:

It's important to recognize the maintainer for each version may decide they can no longer support a version, in which case ruby-core would work to responsibly discontinue the version.

useless sentence

Are both stated to enforce how ruby-core will handle end-of-life responsibly, which is what you suggested before:

What do you want to say in these paragraph?
If this is announce, the audience is users.
They read this, "We want to avoid sudden death".
"ah, ok... and what ruby-core will do?"

However, I think this paragraph:

It's important to recognize the maintainer for each version may decide they can no longer support a version, in which case ruby-core would work to responsibly discontinue the version.

Could be rephrased to simply:

When support is discontinued unexpectedly ruby-core will work to support for a reasonable time until it can be End-of-Life, or an approved commercial party is able to accept maintenance of said version.

I want to reinforce what actions we plan to take to handle discontinuation of a version.

#14 - 03/08/2014 09:50 PM - zzak (Zachary Scott)

@nurse: ping!

Could you review my comments? I'd like to move on from this :)

#15 - 06/14/2014 12:09 AM - zzak (Zachary Scott)

- *Status changed from Assigned to Closed*

I've migrated my (pending) NEWS document from www.r-l.o to the wiki: [GeneralMaintenancePolicy](#)

Please make any changes to the wiki directly, if you need access ping @hsbt.