

Ruby master - Misc #8835

Introducing a semantic versioning scheme and branching policy

08/30/2013 09:49 PM - knu (Akinori MUSHA)

| | |
|--|---------------------|
| Status: | Closed |
| Priority: | Normal |
| Assignee: | knu (Akinori MUSHA) |
| Description | |
| <pre>=begin [This is a presentation for DevelopersMeeting20130831Japan.] Ruby's versioning scheme is currently not well defined or well utilized. First, RUBY_API_VERSION was introduced in 1.9.1, but has not been properly maintained since then. There must have been at least a few ABI incompatible changes (although we tried hard to keep source level compatibility) and many API additions between 1.9.1 and 1.9.3, but RUBY_API_VERSION was never bumped. Secondly, it looks like TEENY version is fixed to zero in the 2.x series without being used effectively. We often find overlooked incompatibility or feature bugs after an official release x.y.0, and would like to fix them in a backward compatible way. I'd suggest we accept such a situation will happen, and bump TEENY when such a fix that affects forward compatibility is made. We have up to nine times of chance for fixing such a situation, and shouldn't be too worried about running out of digits. So, here I propose introducing a new versioning scheme as follows, much with Semantic Versioning (http://semver.org/) in mind: (1) From 2.1.0 and on, RUBY_API_VERSION shall match RUBY_VERSION. Giving them different numbers has been a source of confusion, and introducing this scheme is a way to provide RUBY_VERSION with how to read API compatibility from the numbers. (2) MINOR version is incremented when an incompatible API change is made. In practice, it is incremented before a change to break compatibility is actually made. It is when a relatively significant amount of change is going to be made, which would break compatibility, or hurts stability for a certain period of time. Before making such a change, a new stable branch ruby_{MAJOR}_{MINOR} is cut off from trunk, and MINOR version is bumped on trunk. MAJOR version may be incremented instead, resetting MINOR to zero. On the ruby_{MAJOR}_{MINOR} branch, when the time comes for prereleasing {MAJOR}.{MINOR}.0, a new branch ruby_{MAJOR}_{MINOR}_0 is cut off from the ruby_{MAJOR}_{MINOR} branch. (3) TEENY version is incremented when functionality is added in a backward-compatible manner. It happens on ruby_{MAJOR}_{MINOR} branches, where a new branch ruby_{MAJOR}_{MINOR}_{TEENY} is cut off when it is released, resetting PATCHLEVEL to zero.</pre> | |

ABI Semantic Versioning
Ruby-CI ABI Semantic Versioning

TEENY MAJOR.MINOR.0
API 9

TEENY PATCHLEVEL
TEENY PATCHLEVEL

1.8 1.9 TEENY ABI/API
TEENY TEENY 1 TEENY
TEENY core

FreeBSD
NetBSD

--
Akinori MUSA / <https://akinori.org/>

#4 - 08/31/2013 08:53 AM - kosaki (Motohiro KOSAKI)

2013/8/30 Akinori MUSA knu@idaemons.org:

At Fri, 30 Aug 2013 21:49:34 +0900,
I wrote:

misc #8835: Introducing a semantic versioning scheme and branching policy
<https://bugs.ruby-lang.org/issues/8835>

DevelopersMeeting20130831Japan [ruby-core:56878]
[misc #8835]

[ruby-list:49555] RubyGems 1.8, 1.9, 2.0
PATCHLEVEL RUBY_VERSION
API RUBY_API_VERSION

[MAJOR & MINOR, TEENY, PATCHLEVEL] Semantic
Versioning (<http://semver.org/>)
Ruby API 1.9
RUBY_API_VERSION

ABI
Ruby-CI ABI Semantic Versioning

TEENY MAJOR.MINOR.0
API 9

Ruby

ABI
ABI checker

9

ABI checker

API version 1.9

1.9.3p0 symbol rb_ ruby_prefix Ruby

extconf.rb respond_to? RubyGems

Ruby

TEENY PATCHLEVEL

p0 p0

1.9.0 1.9.1 2.0 leap

TEENY Ruby 99.9% symbol

TEENY=0 TEENY=1

Akinori MUSHA / https://akinori.org/

#8 - 08/31/2013 12:53 PM - kosaki (Motohiro KOSAKI)

2.0 2.1 2.0.2 2.0.4 2.1.0 5

TEENY
PATCHLEVEL
PATCHLEVEL
symbol

p0 p0

1.9.0 1.9.1 2.0
1.9 leap

TEENY Ruby 99.9% symbol

TEENY=0 TEENY=1

#9 - 09/03/2013 01:24 AM - naruse (Yui NARUSE)

ABI Ruby 1.9.1

- ABI →
-
-
- 2
- 1.9.1-1.9.3 ABI (
- 2.0.0 1.9.x ABI

- [ruby-core:19113]
- [ruby-dev:43152]
- [ruby-dev:36655]
- [ruby-dev:36889]
- [ruby-dev:44604]

akr shim ?

kosaki Microsoft patch level

P.S.

#10 - 09/05/2013 04:23 AM - steveklabnik (Steve Klabnik)

I only know a very small amount of Japanese, but the Google Translate of your post was very understandable.

#11 - 12/10/2013 05:18 AM - zzak (Zachary Scott)

We have accepted hsbt's proposal: <https://gist.github.com/sorah/7803201>

#12 - 12/18/2013 07:44 AM - zzak (Zachary Scott)

I have submitted announcement for semantic versioning scheme to ruby-lang.org: <https://github.com/ruby/www.ruby-lang.org/pull/468>

#13 - 01/03/2014 01:57 AM - vo.x (Vit Ondruch)

Do I understand correctly that patch level won't be anymore part of the released tarball name? E.g. retrofitting semver on Ruby 2.0, the first released bugfix release had been ruby-2.0.0-p195.tar.gz, so with the semver, it would have be ruby-2.0.1.tar.gz

I am asking, since in Fedora, we include the patch level in version, i.e. we had ruby-2.0.0.195 package for Ruby 2.0.0-p195 release. Now we can drop this practice, if I am not mistaken.

Thanks for clarification.

#14 - 01/03/2014 04:15 AM - luislavena (Luis Lavena)

vo.x (Vit Ondruch) wrote:

Do I understand correctly that patch level won't be anymore part of the released tarball name? E.g. retrofitting semver on Ruby 2.0, the first released bugfix release had been ruby-2.0.0-p195.tar.gz, so with the semver, it would have be ruby-2.0.1.tar.gz

Hello Vit,

Is my understanding that semantic versioning will apply starting from 2.1.0 and not backward applied to 2.0.0

2.0.0 will continue the same version schema it has but Ruby 2.1 will bump TEENY version number on every release.

Patchlevel will still be increased, but that will refer to the number of "patches" it receive between releases.

#15 - 01/30/2014 06:17 AM - hsbt (Hiroshi SHIBATA)

- Target version changed from 2.1.0 to 2.2.0

#16 - 08/12/2014 03:34 AM - hsbt (Hiroshi SHIBATA)

- Status changed from Open to Closed

We already applied to use SemVer 'like' versioning after Ruby 2.1.

see also. <https://www.ruby-lang.org/en/news/2013/12/21/ruby-version-policy-changes-with-2-1-0/>

Files

| | | | |
|--------|-----------|------------|-----------|
| noname | 205 Bytes | 08/31/2013 | Anonymous |
| noname | 205 Bytes | 08/31/2013 | Anonymous |