

Ruby master - Bug #15791

Clarify reason for RbConfig's ruby_version not reflecting "teeny" value

04/25/2019 03:44 PM - headius (Charles Nutter)

Status: Closed	
Priority: Normal	
Assignee:	
Target version:	
ruby -v: all	Backport: 2.4: UNKNOWN, 2.5: UNKNOWN, 2.6: UNKNOWN
Description <p>I did not realize that MRI always reports RbConfig::CONFIG['ruby_version'] without the "teeny" value. Instead, it makes it always 0:</p> <pre>\$ rvm ruby-2.6.2 do ruby -v -e 'p RbConfig::CONFIG["ruby_version"]' ruby 2.6.2p47 (2019-03-13 revision 67232) [x86_64-darwin18] "2.6.0"</pre> <p>This seems like a bug to me. It is a visible behavior because several package-management systems (RubyGems, Bundler, stuff in Ruby switchers like RVM) use this value, rather than RUBY_VERSION, to set up directory paths.</p> <p>I believe it should reflect the full, accurate version, but I have not been able to find any discussion about why it does not do so.</p>	

History

#1 - 04/25/2019 06:44 PM - MSP-Greg (Greg L)

I always consider RbConfig::CONFIG['ruby_version'] as the ABI version, which is used in naming --user-install gem folders, etc.

If the version with 'teeny' was used, one would be recompiling extension gems everytime a 'teeny' release was done?

#2 - 04/27/2019 07:49 PM - Eregon (Benoit Daloze)

Indeed, AFAIK it's the ABI version.

MSP-Greg (Greg L) wrote:

If the version with 'teeny' was used, one would be recompiling extension gems everytime a 'teeny' release was done?

Which IMHO is a safer thing to do, but I suppose some people disagree and it depends on the point of view :)

FWIW, I created <https://github.com/postmodern/chruby/pull/410> so chruby would have a set of gems per installed Ruby version, even if RUBY_VERSION or RbConfig::CONFIG['ruby_version'] doesn't change for alternative Ruby implementations between two releases.

#3 - 04/29/2019 10:17 PM - headius (Charles Nutter)

MSP-Greg (Greg L) wrote:

I always consider RbConfig::CONFIG['ruby_version'] as the ABI version, which is used in naming --user-install gem folders, etc.

If the version with 'teeny' was used, one would be recompiling extension gems everytime a 'teeny' release was done?

Then why does it include .0 at all?

#4 - 05/01/2019 04:22 PM - MSP-Greg (Greg L)

I assume it was decided a long time ago. 1.9.3 has a value of 1.9.1.

Maybe a decision was made later that ABI versions would only change on minor releases?

#5 - 05/19/2019 08:21 AM - vo.x (Vit Ondruch)

ruby_version is user configurable variable:

<https://github.com/ruby/ruby/blob/trunk/configure.ac#L3663>

It was introduced to make possible parallel install e.g. two Ruby 2.6.0 side by side. Later somebody start to interpret it and add it different meanings, but it was never correct thing to do, therefore I would suggest against using `ruby_version` for anything meaningful. I tried to have this resolved by [#11002](#), but the result was never satisfactory.

#6 - 06/27/2019 08:50 PM - headius (Charles Nutter)

- Status changed from Open to Closed

Ok, if this is supposed to map the the ABI version, then I guess we'll leave it as is. It doesn't really have the same meaning on other impls, and I agree with Vit that it shouldn't be used for other purposes (like it seems to be used in RubyGems).

I'll consider this closed.