Ruby - Feature #14217

Expose RUBY_PATCHLEVEL_STR or similar with patch level info for rc/preview as a constant

12/22/2017 12:46 AM - shayonj (Shayon Mukherjee)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		

Description

Problem

When ruby is in release candidate or preview, RUBY_PATCHLEVEL is -1. Without parsing RUBY_DESCRIPTION, its hard to tell using constant the right and absolute ruby version.

Proposal

Expose RUBY_PATCHLEVEL_STR as a constant, just like RUBY_VERSION or RUBY_PATCHLEVEL. So that, we can know the right ruby version, especially when its in preview or release candidate. This is also helpful when using gems that rely on RUBY_PATCHLEVEL and RUBY_VERSION to serve the right experience. Example: Bundler, which validates gem installation by making sure right ruby is installed. Currently, we cannot install gems using 2.5.0preview1.

```
Your Ruby version is 2.5.0, but your Gemfile specified 2.5.0preview1
```

This can be handled in bundler through some different wokraround, but I think by exposing RUBY_PATCHLEVEL_STR, it will be helpful in building the appropriate solutions.

History

#1 - 12/22/2017 12:51 AM - shayonj (Shayon Mukherjee)

I am happy to a patch/create a Github PR, if this sounds like a good option.

#2 - 12/22/2017 01:04 AM - shayonj (Shayon Mukherjee)

- Subject changed from Expose RUBY_PATCHLEVEL_STR as a constant to Expose RUBY_PATCHLEVEL_STR or similar with patch level info for rc/preview as a constant

#3 - 12/22/2017 12:07 PM - shevegen (Robert A. Heiler)

I can't say whether the name that you haved picked for the constant is good or bad - it seems a
bit long. But I agree on the general problem mentioned, so I am all for more fine-tuned control.

The ruby core team may have decide on the name perhaps.

> Currently, we cannot install gems using 2.5.0preview1.

I think there is another bug report that is vaguely related, e. g. a hardcoded ruby version
to 2.5.0 preview1, for ruby 2.4.3, so you may have a point there. Strictly speaking, the
release for 2.4.3 would be invalid if an internal gem depends on ANOTHER ruby version
that is HIGHER than the released ruby. :)

#4 - 12/22/2017 05:07 PM - shayonj (Shayon Mukherjee)

Agreed that it may appear as long. I chose this as a proposal, because it was already defined and was simpler to expose. Example:

```
From 8cef26bbdf314dccf1d36984a49c35f0a815bbea Mon Sep 17 00:00:00 2001
From: Shayon Mukherjee <dev@shayon.me>
Date: Thu, 21 Dec 2017 16:51:13 -0800
Subject: [PATCH] Introduce RUBY_PATCHLEVEL_STR constant

---
 version.c | 7 +++++++
 1 file changed, 7 insertions(+)

diff --git a/version.c b/version.c
```

```
index 4c7b1abb40..6c66c12eb0 100644
--- a/version.c
+++ b/version.c
@@ -30,6 +30,7 @@ const char ruby_version[] = RUBY_VERSION;
 const char ruby_release_date[] = RUBY_RELEASE_DATE;
 const char ruby_platform[] = RUBY_PLATFORM;
 const int ruby_patchlevel = RUBY_PATCHLEVEL;
+const char ruby_patchlevel_str[] = RUBY_PATCHLEVEL_STR;
 const char ruby_description[] = RUBY_DESCRIPTION;
 const char ruby_copyright[] = RUBY_COPYRIGHT;
 const char ruby_engine[] = "ruby";
@@ -59,6 +60,12 @@ Init_version(void)
      * the patchlevel will be -1
      */
     rb_define_global_const("RUBY_PATCHLEVEL", MKINT(patchlevel));
+    /*
+     * The patchlevel string for this ruby. If this is a development build
+     * of ruby the patchlevel string will be dev otherwise the respective
+     * rc or preview candidate.
+     */
+    rb_define_global_const("RUBY_PATCHLEVEL_STR", MKSTR(patchlevel_str));
     /*
      * The SVN revision for this ruby.
      */
--
2.15.1
```

IMO, RUBY_PATCHLEVEL_STR still makes sense since it differentiates it from RUBY_PATCHLEVEL, but at the same time I am not too big on the idea of having the the type of the return value in a constant name. I am more than happy to go with a different name. Few others I had in mind were RUBY_PATCHNAME and RUBY_PATCHLEVEL_NAME.

#5 - 05/31/2018 04:48 PM - shayonj (Shayon Mukherjee)

Is it possible to get any updates on this approach? :)

#6 - 06/01/2018 03:00 AM - nobu (Nobuyoshi Nakada)

- Description updated

Could you elaborate how you want to use it?

#7 - 04/02/2024 06:29 AM - hsbt (Hiroshi SHIBATA)

- Status changed from Open to Closed

This issue has been fixed at recent version of Bundler.

https://github.com/rubygems/rubygems/pull/7016