Ruby master - Feature #16742

RbConfig.windows? and RbConfig.host_os

03/29/2020 03:14 PM - Eregon (Benoit Daloze)

Status:	Open	
Priority:	Normal	
Assignee:		
Target version:		

Description

I think adding these two methods would greatly clarify platform checks:

```
RbConfig.windows? # obvious, much clearer than RUBY_PLATFORM =~ /(mswin|mingw)/
RbConfig.host_os # same as RbConfig::CONFIG['host_os'], but much shorter
```

RbConfig::CONFIG['host_os'] is too long and feels like digging in the internals even though it's such a simple thing to query.

For comparison, there is already a RbConfig.ruby method on RbConfig, which helps avoiding the boilerplate to access RbConfig::CONFIG manually.

Thoughts?

History

#1 - 03/29/2020 03:16 PM - Eregon (Benoit Daloze)

- Description updated

#2 - 03/29/2020 03:19 PM - Eregon (Benoit Daloze)

Also worth nothing RUBY_PLATFORM =~ /(mswin|mingw)/ is hacky and incomplete, and there are many subtle variants of that check such as RUBY_PLATFORM =~ /win32|mingw|bccwin|cygwin/.
RbConfig.windows? would be far more reliable and easier to read.

#3 - 03/29/2020 04:59 PM - shevegen (Robert A. Heiler)

I agree that RbConfig.windows? seems intuitive to me, in the sense that it would or could or should refer to the windows (Microsoft) platform.

I also concur with the explanation given.

As for RbConfig.host_os - I have no real preference either way, but I think it would be an ok addition too; it is shorter. People who prefer the longer variant can still use the longer variant too, so no change is necessary. So that suggestion seems ok as well (to me) - but I have no real preference here (so mostly neutral), whereas I do think RbConfig.windows? I am fairly positive.

> Also worth nothing RUBY_PLATFORM =~ /(mswin|mingw)/ is hacky and incomplete, and there are many subtle variants of that check such as RUBY_PLATFORM =~ /win32|mingw|bccwin|cygwin/.

Reminds me of ms-greg (I don't recall the proper name right now, sorry).

But anyway, my personal opinion is that the suggestions here seem useful. I also think there is no trade off when using ".windows?" since I do not think it will conflict with anything else. (I have not so strong a preference to comment on .host_os, I leave that to others if they feel strongly about it either way.)

#4 - 04/09/2020 02:43 AM - nobu (Nobuyoshi Nakada)

- Description updated

#5 - 04/10/2020 05:54 AM - matz (Yukihiro Matsumoto)

The #host_os seems OK.

Regarding #windows?, what do you want to check using the value? Drive letter? Path separator? Case insensitivity? Process behavior? Posix compatibility? Should it be true on Cygwin? It is not clear yet.

Matz.

#6 - 04/10/2020 06:00 AM - nobu (Nobuyoshi Nakada)

Is host_os what you want to see?
RbConfig::CONFIG['host_os'] is bare value by config.guess, like "linux-gnu", "darwin19.4.0", etc.
RUBY_PLATFORM consists of "target_cpu" and "target_os" instead.

#7 - 04/10/2020 11:53 AM - Eregon (Benoit Daloze)

For RbConfig.host_os I expect the same as RbConfig::CONFIG['host_os'], or better slightly simplified (no version, no variant information) like linux, darwin, mingw, mswin, etc (easier to check, and we could document the various possible values).

Similar to process.platform() on Node.js:
https://nodejs.org/api/process.html#process_process_platform (except win32 => windows)

For RbConfig.windows? that would include at least mswin and mingw.

Not sure about cygwin, what do you think MSP-Greg (Greg L)?
WSL would clearly be false for RbConfig.windows?, so I think it should be false for cygwin too since it also tries to hide the fact you're on Windows.

There are many checks for Windows for various reasons, but notably for whether some system features are supported.
Some of them can be checked more directly (e.g., fork, unix sockets) but not all (e.g., proper support for creating symlinks) which explain why we see RUBY_PLATFORM =~ /(mswin|mingw)/ in so many gems and tests and specs.

#8 - 04/11/2020 02:00 PM - MSP-Greg (Greg L)

> it should be false for cygwin too since it also tries to hide the fact you're on Windows.

Agreed. I think. I've never worked with the platform.

I've seen many gems/repos that have .windows?, .darwin?, .jruby?, .mingw?, mswin?, etc in their library or test code. Adding any similar predicate methods to Ruby would be helpful. Some use constants (RbConfig::WINDOWS, RbConfig::DARWIN, etc)...