Ruby master - Feature #12739

deprecate_constant :Fixnum, :Bignum

09/08/2016 02:42 PM - znz (Kazuhiro NISHIYAMA)

Status: Closed
Priority: Normal
Assignee:
Target version:

Description

How about class Object; deprecate_constant :Fixnum, :Bignum; end?

- naruse supports: https://twitter.com/nalsh/status/773800491671101441
- akr objects: https://twitter.com/tanaka_akra/status/773817391218249729

I don't mind either way. Do you have any comment about it?

Related issues:
Related to Ruby master - Feature #12005: Unify Fixnum and Bignum into Integer - Closed

Revision 772fd010 - 09/19/2016 08:05 AM - nobu (Nobuyoshi Nakada)
deprecate Fixnum and Bignum

- numeric.c (Init_Numeric), bignum.c (Init_Bignum): deprecate Fixnum and Bignum. this may be reverted after previews. [Feature #12739]

Revision 56187 - 09/19/2016 08:05 AM - nobu (Nobuyoshi Nakada)
deprecate Fixnum and Bignum

- numeric.c (Init_Numeric), bignum.c (Init_Bignum): deprecate Fixnum and Bignum. this may be reverted after previews. [Feature #12739]

Revision 56187 - 09/19/2016 08:05 AM - nobu (Nobuyoshi Nakada)
deprecate Fixnum and Bignum

- numeric.c (Init_Numeric), bignum.c (Init_Bignum): deprecate Fixnum and Bignum. this may be reverted after previews. [Feature #12739]

Revision 56187 - 09/19/2016 08:05 AM - nobu (Nobuyoshi Nakada)
deprecate Fixnum and Bignum

- numeric.c (Init_Numeric), bignum.c (Init_Bignum): deprecate Fixnum and Bignum. this may be reverted after previews. [Feature #12739]

Revision 56187 - 09/19/2016 08:05 AM - nobu (Nobuyoshi Nakada)
deprecate Fixnum and Bignum

- numeric.c (Init_Numeric), bignum.c (Init_Bignum): deprecate Fixnum and Bignum. this may be reverted after previews. [Feature #12739]

History

#1 - 09/08/2016 02:44 PM - znz (Kazuhiro NISHIYAMA)
- Related to Feature #12005: Unify Fixnum and Bignum into Integer added

#2 - 09/10/2016 04:33 AM - nobu (Nobuyoshi Nakada)
I tried this, found rake defines Fixnum::MAX which seems useless, and submitted a PR to rake.
But it fails on ruby 2.2 or earlier, due to incompatibility of Bundler 1.13.0.

#3 - 09/19/2016 08:05 AM - nobu (Nobuyoshi Nakada)
- Status changed from Open to Closed
deprecate Fixnum and Bignum

- numeric.c (Init_Numeric), bignum.c (Init_Bignum): deprecate Fixnum and Bignum. this may be reverted after previews. [Feature #12739]

### #4 - 06/04/2017 09:44 PM - robergtleeson (Robert Gleeson)

znz (Kazuhiro NISHIYAMA) wrote:

> How about class Object; deprecate_constant :Fixnum,:Bignum; end?

- naruse supports: https://twitter.com/nalsh/status/773800491671101441
- akr objects: https://twitter.com/tanaka_akhir/status/773817391218249729

I don't mind either way.
Do you have any comment about it?

It would be nice to have constant_deprecated?,
To avoid all of these warnings while supporting <2.4 and >=2.4 i track what constants are deprecated in an array, and then avoid using them or not by checking the array at runtime. It's bit of a pain..

### #5 - 06/04/2017 09:54 PM - robergtleeson (Robert Gleeson)

robergtleeson (Robert Gleeson) wrote:

znz (Kazuhiro NISHIYAMA) wrote:

> How about class Object; deprecate_constant :Fixnum,:Bignum; end?

- naruse supports: https://twitter.com/nalsh/status/773800491671101441
- akr objects: https://twitter.com/tanaka_akhir/status/773817391218249729

I don't mind either way.
Do you have any comment about it?

It would be nice to have constant_deprecated?,
To avoid all of these warnings while supporting <2.4 and >=2.4 i track what constants are deprecated in an array, and then avoid using them or not by checking the array at runtime. It's bit of a pain..

I realise my comment isn't clear, here's very simple code to show what i mean:

```ruby
deprecated_constants = [:Fixnum,:TimeoutError,:Bignum,:FALSE,:TRUE,:NIL]
Object.constants.each do |constant|
  const_get(constant) unless deprecated_constants.include?(constant)
end
```

the real code i have is more complex, but in essence this is the problem, and deprecated_constants has to be maintained by me manually, which will become out of date over time.

### #6 - 06/16/2017 01:28 PM - znz (Kazuhiro NISHIYAMA)

Closed ticket should not be reused.
Please open new ticket.