

Ruby trunk - Feature #12005

Unify Fixnum and Bignum into Integer

01/19/2016 02:38 AM - naruse (Yui NARUSE)

Status: Closed	
Priority: Normal	
Assignee: mrkn (Kenta Murata)	
Target version:	
Description CRuby has two Integer classes, Fixnum and Bignum. But it is implementation detail. They should be seen as a single class Integer like Flonum.	
Compatibility note <ul style="list-style-type: none">• Q: How do I check whether Fixnum and Bignum are unified or not?• A: check RUBY_INTEGER_UNIFICATION macro	
Related issues:	
Related to Ruby trunk - Bug #12427: Defining methods with the same name to bo...	Closed
Related to Ruby trunk - Feature #12739: deprecate_constant :Fixnum, :Bignum	Closed

Associated revisions

Revision f9727c12 - 05/17/2016 06:53 AM - akr (Akira Tanaka)

[Feature #12005] Unify Fixnum and Bignum into Integer

- [Feature #12005] Unify Fixnum and Bignum into Integer
- include/ruby/ruby.h (rb_class_of): Return rb_cInteger for fixnums.
- insns.def (INTEGER_REDEFINED_OP_FLAG): Unified from FIXNUM_REDEFINED_OP_FLAG and BIGNUM_REDEFINED_OP_FLAG.
- vm_core.h: Ditto.
- vm_inshelper.c (opt_eq_func): Use INTEGER_REDEFINED_OP_FLAG instead of FIXNUM_REDEFINED_OP_FLAG.
- vm.c (vm_redefinition_check_flag): Use rb_cInteger instead of rb_cFixnum and rb_cBignum.
(C): Use Integer instead of Fixnum and Bignum.
- numeric.c (fix_succ): Removed.
(Init_Numeric): Define Fixnum as Integer.
- bignum.c (bignew): Use rb_cInteger instead of Rb_cBignum.
(rb_int_coerce): replaced from rb_big_coerce and return fixnums as-is.
(Init_Bignum): Define Bignum as Integer.
Don't define ==.
- error.c (builtin_class_name): Return "Integer" for fixnums.
- sprintf.c (ruby__sfvextra): Use rb_cInteger instead of rb_cFixnum.
- ext/-test-/testutil: New directory to test.
Currently it provides utilities for fixnum and bignum.
- ext/json/generator/generator.c: Define mInteger_to_json.
- lib/mathn.rb (Fixnum#): Redefinition removed.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@55024 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 55024 - 05/17/2016 06:53 AM - akr (Akira Tanaka)

[Feature #12005] Unify Fixnum and Bignum into Integer

- [Feature #12005] Unify Fixnum and Bignum into Integer
- include/ruby/ruby.h (rb_class_of): Return rb_cInteger for fixnums.
- insns.def (INTEGER_REDEFINED_OP_FLAG): Unified from FIXNUM_REDEFINED_OP_FLAG and BIGNUM_REDEFINED_OP_FLAG.
- vm_core.h: Ditto.
- vm_inshelper.c (opt_eq_func): Use INTEGER_REDEFINED_OP_FLAG instead of FIXNUM_REDEFINED_OP_FLAG.
- vm.c (vm_redefinition_check_flag): Use rb_cInteger instead of rb_cFixnum and rb_cBignum.
(C): Use Integer instead of Fixnum and Bignum.
- numeric.c (fix_succ): Removed.
(Init_Numeric): Define Fixnum as Integer.
- bignum.c (bignew): Use rb_cInteger instead of Rb_cBignum.
(rb_int_coerce): replaced from rb_big_coerce and return fixnums as-is.
(Init_Bignum): Define Bignum as Integer.
Don't define ==.
- error.c (builtin_class_name): Return "Integer" for fixnums.
- sprintf.c (ruby__sfvextra): Use rb_cInteger instead of rb_cFixnum.
- ext/-test-/testutil: New directory to test.
Currently it provides utilities for fixnum and bignum.
- ext/json/generator/generator.c: Define mInteger_to_json.
- lib/mathn.rb (Fixnum#): Redefinition removed.

Revision 55024 - 05/17/2016 06:53 AM - akr (Akira Tanaka)

[Feature #12005] Unify Fixnum and Bignum into Integer

- [Feature #12005] Unify Fixnum and Bignum into Integer
- include/ruby/ruby.h (rb_class_of): Return rb_cInteger for fixnums.
- insns.def (INTEGER_REDEFINED_OP_FLAG): Unified from FIXNUM_REDEFINED_OP_FLAG and BIGNUM_REDEFINED_OP_FLAG.
- vm_core.h: Ditto.
- vm_inshelper.c (opt_eq_func): Use INTEGER_REDEFINED_OP_FLAG instead of FIXNUM_REDEFINED_OP_FLAG.
- vm.c (vm_redefinition_check_flag): Use rb_cInteger instead of rb_cFixnum and rb_cBignum.
(C): Use Integer instead of Fixnum and Bignum.
- numeric.c (fix_succ): Removed.
(Init_Numeric): Define Fixnum as Integer.
- bignum.c (bignew): Use rb_cInteger instead of Rb_cBignum.
(rb_int_coerce): replaced from rb_big_coerce and return fixnums as-is.
(Init_Bignum): Define Bignum as Integer.
Don't define ==.
- error.c (builtin_class_name): Return "Integer" for fixnums.
- sprintf.c (ruby__sfvextra): Use rb_cInteger instead of rb_cFixnum.

- ext/-test-/testutil: New directory to test.
Currently it provides utilities for fixnum and bignum.
- ext/json/generator/generator.c: Define mlInteger_to_json.
- lib/mathn.rb (Fixnum#): Redefinition removed.

Revision 55024 - 05/17/2016 06:53 AM - akr (Akira Tanaka)

[Feature #12005] Unify Fixnum and Bignum into Integer

- [Feature #12005] Unify Fixnum and Bignum into Integer
- include/ruby/ruby.h (rb_class_of): Return rb_cInteger for fixnums.
- insns.def (INTEGER_REDEFINED_OP_FLAG): Unified from
FIXNUM_REDEFINED_OP_FLAG and BIGNUM_REDEFINED_OP_FLAG.
- vm_core.h: Ditto.
- vm_inshelper.c (opt_eq_func): Use INTEGER_REDEFINED_OP_FLAG instead
of FIXNUM_REDEFINED_OP_FLAG.
- vm.c (vm_redefinition_check_flag): Use rb_cInteger instead of
rb_cFixnum and rb_cBignum.
(C): Use Integer instead of Fixnum and Bignum.
- numeric.c (fix_succ): Removed.
(Init_Numeric): Define Fixnum as Integer.
- bignum.c (bignew): Use rb_cInteger instead of Rb_cBignum.
(rb_int_coerce): replaced from rb_big_coerce and return fixnums
as-is.
(Init_Bignum): Define Bignum as Integer.
Don't define ==.
- error.c (builtin_class_name): Return "Integer" for fixnums.
- sprintf.c (ruby__sfvextra): Use rb_cInteger instead of rb_cFixnum.
- ext/-test-/testutil: New directory to test.
Currently it provides utilities for fixnum and bignum.
- ext/json/generator/generator.c: Define mlInteger_to_json.
- lib/mathn.rb (Fixnum#): Redefinition removed.

Revision 55024 - 05/17/2016 06:53 AM - akr (Akira Tanaka)

[Feature #12005] Unify Fixnum and Bignum into Integer

- [Feature #12005] Unify Fixnum and Bignum into Integer
- include/ruby/ruby.h (rb_class_of): Return rb_cInteger for fixnums.
- insns.def (INTEGER_REDEFINED_OP_FLAG): Unified from
FIXNUM_REDEFINED_OP_FLAG and BIGNUM_REDEFINED_OP_FLAG.
- vm_core.h: Ditto.
- vm_inshelper.c (opt_eq_func): Use INTEGER_REDEFINED_OP_FLAG instead
of FIXNUM_REDEFINED_OP_FLAG.
- vm.c (vm_redefinition_check_flag): Use rb_cInteger instead of
rb_cFixnum and rb_cBignum.
(C): Use Integer instead of Fixnum and Bignum.
- numeric.c (fix_succ): Removed.
(Init_Numeric): Define Fixnum as Integer.
- bignum.c (bignew): Use rb_cInteger instead of Rb_cBignum.

(rb_int_coerce): replaced from rb_big_coerce and return fixnums as-is.
(Init_Bignum): Define Bignum as Integer.
Don't define ==.

- error.c (builtin_class_name): Return "Integer" for fixnums.
- sprintf.c (ruby__sfvextra): Use rb_cInteger instead of rb_cFixnum.
- ext/-test-/testutil: New directory to test.
Currently it provides utilities for fixnum and bignum.
- ext/json/generator/generator.c: Define mInteger_to_json.
- lib/mathn.rb (Fixnum#?): Redefinition removed.

History

#1 - 01/19/2016 04:34 AM - nobu (Nobuyoshi Nakada)

- Description updated

Could you elaborate the road map?

1. move methods from Fixnum and Bignum to Integer, with code to dispatch by its range
2. deprecate direct use of Fixnum and Bignum
3. remove them

Please correct it if something is missing.

#2 - 01/23/2016 08:44 AM - shevegen (Robert A. Heiler)

I like the idea behind it. \o/

Would probably make more sense too if the defining difference for it used to be due to performance reasons, as Yui NARUSE wrote.

#3 - 02/21/2016 03:36 PM - jwmittag (Jörg W Mittag)

Yui NARUSE wrote:

CRuby has two Integer classes, Fixnum and Bignum.
But it is implementation detail.
They should be seen as a single class Integer like Flonum.

I like this very much! It always struck me as odd that Integer is special-cased like that. Some implementations have special-cased optimizations for small arrays or hashes (e.g. less than 6 elements), yet they don't have a separate SmallArray or SmallHash class. Some implementations have special-cased optimizations for small objects (e.g. less than 3 instance variables), yet they don't have a separate SmallObject inheritance hierarchy. YARV has flonums, yet no separate Flonum class. And so on ...

The ISO Ruby Language Specification also explicitly treats them as private internal implementation details. It only specifies the Integer class and then allows for "implementation-defined subclasses". (And in fact, I suspect that sentence was only put in there because Fixnum/Bignum already existed when the spec was written.)

Treating it as a private internal implementation detail would also give more freedom to implementors to choose different, more aggressive, or maybe just no optimizations at all. (The latter might be useful for very small implementations intended for embedded use, or very simple implementations intended for educational use.)

#4 - 03/16/2016 07:39 AM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Assigned

- Assignee set to mrkn (Kenta Murata)

#5 - 05/03/2016 10:18 AM - akr (Akira Tanaka)

- File unify-fixnum-and-bignum.patch added

I made a patch to unify Fixnum and Bignum:
unify-fixnum-and-bignum.patch

For rough compatibility, I defined Fixnum and Bignum as Integer.

So, `foo.kind_of?(Fixnum)` works as `foo.kind_of?(Integer)`.
This works mostly because most application doesn't have interest on the boundary between `Fixnum` and `Bignum` which vary on platforms.

#6 - 05/04/2016 02:54 PM - akr (Akira Tanaka)

I feel that unifying `Fixnum` and `Bignum` is simple and beautiful.

However I'm not sure that there are enough concrete benefits over the incompatibility.

For example, when I made the patch, I need to fix `complex.c` and `ext/json` because they used `Fixnum` and `Bignum` class to dispatch.

`mathn.rb` is another example. `mathn.rb` modifies methods in `Fixnum` and `Bignum`.

I changed `mathn.rb` to modify methods in `Integer`.

Concrete benefits I know is that less code for `mathn.rb` like library. It needs to modify only one class, `Integer`, instead of two, `Fixnum` and `Bignum`.

Other benefits?

Are they enough to justify the incompatibility?

I hope enough benefits to justify this issue.

#7 - 05/05/2016 03:11 AM - duerst (Martin Dürst)

On 2016/05/04 23:54, akr@fsij.org wrote:

I feel that unifying `Fixnum` and `Bignum` is simple and beautiful.

However I'm not sure that there are enough concrete benefits over the incompatibility.

Concrete benefits I know is that less code for `mathn.rb` like library. It needs to modify only one class, `Integer`, instead of two, `Fixnum` and `Bignum`.

Other benefits?

Are they enough to justify the incompatibility?

I hope enough benefits to justify this issue.

I think there are some serious benefits in documentation and for learning Ruby. But these are not so concrete, and more long term.

Regards, Martin.

#8 - 05/05/2016 06:00 PM - naruse (Yui NARUSE)

Akira Tanaka wrote:

Other benefits?

As you show people write `Fixnum` case and `Bignum` case. But sometimes they write only `Fixnum` even though they must write `Bignum` case. Unified `Integer` avoids such pitfall.

#9 - 05/17/2016 06:39 AM - matz (Yukihiro Matsumoto)

I should have unified those classes at the first hand. It's much cleaner & simpler. Try unifying them and see if it would go well.

Matz.

#10 - 05/17/2016 06:53 AM - akr (Akira Tanaka)

- Status changed from *Assigned* to *Closed*

Applied in changeset r55024.

[Feature [#12005](#)] Unify `Fixnum` and `Bignum` into `Integer`

- [Feature #12005] Unify Fixnum and Bignum into Integer
- include/ruby/ruby.h (rb_class_of): Return rb_cInteger for fixnums.
- insns.def (INTEGER_REDEFINED_OP_FLAG): Unified from FIXNUM_REDEFINED_OP_FLAG and BIGNUM_REDEFINED_OP_FLAG.
- vm_core.h: Ditto.
- vm_inshelper.c (opt_eq_func): Use INTEGER_REDEFINED_OP_FLAG instead of FIXNUM_REDEFINED_OP_FLAG.
- vm.c (vm_redefinition_check_flag): Use rb_cInteger instead of rb_cFixnum and rb_cBignum.
(C): Use Integer instead of Fixnum and Bignum.
- numeric.c (fix_succ): Removed.
(Init_Numeric): Define Fixnum as Integer.
- bignum.c (bignew): Use rb_cInteger instead of Rb_cBignum.
(rb_int_coerce): replaced from rb_big_coerce and return fixnums as-is.
(Init_Bignum): Define Bignum as Integer.
Don't define ==.
- error.c (builtin_class_name): Return "Integer" for fixnums.
- sprintf.c (ruby__sfvextra): Use rb_cInteger instead of rb_cFixnum.
- ext/-test-/testutil: New directory to test.
Currently it provides utilities for fixnum and bignum.
- ext/json/generator/generator.c: Define mInteger_to_json.
- lib/mathn.rb (Fixnum#): Redefinition removed.

#11 - 06/15/2016 04:44 AM - naruse (Yui NARUSE)

- Related to Bug #12427: Defining methods with the same name to both Fixnum and Bignum classes could cause SEGV in C extensions since Feature #12005 added

#12 - 06/28/2016 05:26 AM - naruse (Yui NARUSE)

- Description updated

#13 - 09/08/2016 02:44 PM - znz (Kazuhiro NISHIYAMA)

- Related to Feature #12739: deprecate_constant :Fixnum, :Bignum added

Files

unify-fixnum-and-bignum.patch	30.5 KB	05/03/2016	akr (Akira Tanaka)
-------------------------------	---------	------------	--------------------