

Ruby master - Feature #6936

Forbid singleton class and instance variables for float

08/27/2012 08:12 AM - naruse (Yui NARUSE)

Status:	Closed
Priority:	Normal
Assignee:	ko1 (Koichi Sasada)
Target version:	2.0.0
Description	
[Feature #6763] flonum r36798	
<ol style="list-style-type: none">Float IDflonum floatflonum floatflonum float	
1. 2. 3. 4. frozen	
Ruby	
<ul style="list-style-type: none">flonum floatfloat	
true, false, nil, Fixnum, Symbol	
Bignum Time immutable	
Related issues:	
Related to Backport187 - Backport #4578: Fixnum.freeze not frozen?	Closed
Related to Ruby master - Feature #6763: Introduce Flonum technique to speedup...	Closed 07/21/2012
Related to Ruby master - Feature #3222: Can bignums have singleton class & me...	Closed 04/30/2010

Associated revisions

Revision 180c744b - 10/27/2012 02:10 AM - ko1 (Koichi Sasada)

- numeric.c (rb_float_new_in_heap), include/ruby/ruby.h: make all Float objects frozen. [ruby-dev:46081] [ruby-trunk - Feature #6936] Most part of patch by NARUSE, Yui naruse@ruby-lang.org.
- class.c (singleton_class_of): raise TypeError when trying to define a singleton method on Float objects.
- vm.c (vm_define_method): ditto.
- test/ruby/marshaltestlib.rb: catch up above changes.
- test/ruby/test_class.rb: ditto.
- test/test_pp.rb: ditto.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@37341 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 37341 - 10/27/2012 02:10 AM - ko1 (Koichi Sasada)

- numeric.c (rb_float_new_in_heap), include/ruby/ruby.h: make all Float objects frozen. [ruby-dev:46081] [ruby-trunk - Feature #6936] Most part of patch by NARUSE, Yui naruse@ruby-lang.org.
- class.c (singleton_class_of): raise TypeError when trying to define a singleton method on Float objects.
- vm.c (vm_define_method): ditto.
- test/ruby/marshaltestlib.rb: catch up above changes.
- test/ruby/test_class.rb: ditto.
- test/test_pp.rb: ditto.

Revision 37341 - 10/27/2012 02:10 AM - ko1 (Koichi Sasada)

- numeric.c (rb_float_new_in_heap), include/ruby/ruby.h: make all Float objects frozen. [ruby-dev:46081] [ruby-trunk - Feature #6936] Most part of patch by NARUSE, Yui naruse@ruby-lang.org.
- class.c (singleton_class_of): raise TypeError when trying to define a singleton method on Float objects.
- vm.c (vm_define_method): ditto.
- test/ruby/marshaltestlib.rb: catch up above changes.
- test/ruby/test_class.rb: ditto.
- test/test_pp.rb: ditto.

Revision 37341 - 10/27/2012 02:10 AM - ko1 (Koichi Sasada)

- numeric.c (rb_float_new_in_heap), include/ruby/ruby.h: make all Float objects frozen. [ruby-dev:46081] [ruby-trunk - Feature #6936] Most part of patch by NARUSE, Yui naruse@ruby-lang.org.
- class.c (singleton_class_of): raise TypeError when trying to define a singleton method on Float objects.
- vm.c (vm_define_method): ditto.
- test/ruby/marshaltestlib.rb: catch up above changes.
- test/ruby/test_class.rb: ditto.
- test/test_pp.rb: ditto.

Revision 37341 - 10/27/2012 02:10 AM - ko1 (Koichi Sasada)

- numeric.c (rb_float_new_in_heap), include/ruby/ruby.h: make all Float objects frozen. [ruby-dev:46081] [ruby-trunk - Feature #6936] Most part of patch by NARUSE, Yui naruse@ruby-lang.org.
- class.c (singleton_class_of): raise TypeError when trying to define a singleton method on Float objects.
- vm.c (vm_define_method): ditto.
- test/ruby/marshaltestlib.rb: catch up above changes.
- test/ruby/test_class.rb: ditto.
- test/test_pp.rb: ditto.

Revision 37341 - 10/27/2012 02:10 AM - ko1 (Koichi Sasada)

- numeric.c (rb_float_new_in_heap), include/ruby/ruby.h: make all Float objects frozen. [ruby-dev:46081] [ruby-trunk - Feature #6936] Most part of patch by NARUSE, Yui naruse@ruby-lang.org.
- class.c (singleton_class_of): raise TypeError when trying to define a singleton method on Float objects.
- vm.c (vm_define_method): ditto.
- test/ruby/marshaltestlib.rb: catch up above changes.
- test/ruby/test_class.rb: ditto.
- test/test_pp.rb: ditto.

Revision 37341 - 10/27/2012 02:10 AM - ko1 (Koichi Sasada)

- numeric.c (rb_float_new_in_heap), include/ruby/ruby.h: make all Float objects frozen. [ruby-dev:46081] [ruby-trunk - Feature #6936] Most part of patch by NARUSE, Yui naruse@ruby-lang.org.
- class.c (singleton_class_of): raise TypeError when trying to define a singleton method on Float objects.
- vm.c (vm_define_method): ditto.
- test/ruby/marshaltestlib.rb: catch up above changes.
- test/ruby/test_class.rb: ditto.
- test/test_pp.rb: ditto.

History

#1 - 08/27/2012 03:23 PM - ko1 (Koichi Sasada)

(2012/08/27 8:12), naruse (Yui NARUSE) wrote:

```
float frozen
```

```
Numeric freeze
```

```
--
// SASADA Koichi at atdot dot net
```

#2 - 08/27/2012 05:53 PM - kosaki (Motohiro KOSAKI)

```
float frozen
```

Numeric freeze

#3 - 08/27/2012 10:53 PM - naruse (Yui NARUSE)

(2012/08/27 17:46), KOSAKI Motohiro wrote:

frozen

Numeric freeze

NARUSE, Yui naruse@airemix.jp

#4 - 08/28/2012 09:59 AM - usa (Usaku NAKAMURA)

In message "[ruby-dev:46082] Re: [ruby-trunk - Feature #6936][Assigned] Forbid singleton class and instance variabls for float" on Aug.27,2012 15:22:07, ko1@atdot.net wrote:

Numeric freeze

+1

U.Nakamura usa@garbagecollect.jp

#5 - 08/28/2012 10:23 PM - mrkn (Kenta Murata)

2012/8/28 U.Nakamura usa@garbagecollect.jp:

In message "[ruby-dev:46082] Re: [ruby-trunk - Feature #6936][Assigned] Forbid singleton class and instance variabls for float" on Aug.27,2012 15:22:07, ko1@atdot.net wrote:

Numeric freeze

+1

+1

Kenta Murata
OpenPGP FP = 1D69 ADDE 081C 9CC2 2E54 98C1 CEFE 8AFB 6081 B062

Ruby <http://www.amazon.co.jp/dp/4798119881/mrkn-22>

E-mail: mrkn@mrkn.jp
twitter: <http://twitter.com/mrkn/>
blog: <http://d.hatena.ne.jp/mrkn/>

#6 - 08/30/2012 01:59 AM - Anonymous

(Perl)

(2012/08/28 22:04), Kenta Murata wrote:

Numeric freeze

Float frozen

```
diff --git a/class.c b/class.c
index 1d871fb..1df38e4 100644
--- a/class.c
+++ b/class.c
@@ -1324,6 +1324,10 @@ singleton_class_of(VALUE obj)
rb_bug("unknown immediate %p", (void *)obj);
return klass;
}
```

- else {
- if (BUILTIN_TYPE(obj) == T_FLOAT)
- rb_raise(rb_eTypeError, "can't define singleton");

• }

```
if (FL_TEST(RBASIC(obj)->klass, FL_SINGLETON) &&
rb_ivar_get(RBASIC(obj)->klass, id_attached) == obj) {
diff --git a/include/ruby/ruby.h b/include/ruby/ruby.h
index a674de8..53de6a8 100644
--- a/include/ruby/ruby.h
+++ b/include/ruby/ruby.h
@@ -1129,7 +1129,7 @@ struct RBasic {
(FL_TAINT | FL_UNTRUSTED); \
} while (0)
```

```

#define OBJ_FROZEN(x) (!FL_TEST((x), FL_FREEZE))
+#define OBJ_FROZEN(x) (!FL_ABLE(x)?(RBASIC(x)->flags&(FL_FREEZE)):FLONUM_P(x))
#define OBJ_FREEZE(x) FL_SET((x), FL_FREEZE)
```

```

#if SIZEOF_INT < SIZEOF_LONG
diff --git a/numeric.c b/numeric.c
index 58ac7ad..6a72fba 100644
--- a/numeric.c
+++ b/numeric.c
@@ -621,6 +621,7 @@ rb_float_new_in_heap(double d)
OBJSETUP(flt, rb_cFloat, T_FLOAT);
```

```
flt->float_value = d;
```

- OBJ_FREEZE(flt); return (VALUE)flt; }

#10 - 10/27/2012 08:38 AM - matz (Yukihiro Matsumoto)

- Assignee changed from matz (Yukihiro Matsumoto) to ko1 (Koichi Sasada)

OK, I'd like to see if everything goes well. Merge it.

Matz.

#11 - 10/27/2012 09:23 AM - ko1 (Koichi Sasada)

(2012/10/27 8:38), matz (Yukihiro Matsumoto) wrote:

Assignee changed from matz (Yukihiro Matsumoto) to ko1 (Koichi Sasada)

OK, I'd like to see if everything goes well. Merge it.

Roger, boss.

--
// SASADA Koichi at atdot dot net

#12 - 10/27/2012 10:23 AM - ko1 (Koichi Sasada)

(2012/10/27 9:05), SASADA Koichi wrote:

(2012/10/27 8:38), matz (Yukihiko Matsumoto) wrote:

Assignee changed from matz (Yukihiko Matsumoto) to ko1 (Koichi Sasada)

OK, I'd like to see if everything goes well. Merge it.

□□□□□□□□□□
test-all □□□□□□□□

Running tests:

[846/11367] PPTestModule::PPInspectTest#test_to_s_with_iv = 0.00 s

1) Error:

test_to_s_with_iv(PPTestModule::PPInspectTest):
RuntimeError: can't modify frozen Float
/mnt/sdb1/ruby/trunk/test/test_pp.rb:111:in block in test_to_s_with_iv'
/mnt/sdb1/ruby/trunk/test/test_pp.rb:111:ininstance_eval'
/mnt/sdb1/ruby/trunk/test/test_pp.rb:111:in `test_to_s_with_iv'

[2709/11367] TestClass#test_singleton_class = 0.00 s

2) Failure:

test_singleton_class(TestClass) [/mnt/sdb1/ruby/trunk/test/ruby/test_class.rb:195]:
Exception raised:
<#>.

[4022/11367] TestFloat#test_singleton_method = 0.00 s

3) Failure:

test_singleton_method(TestFloat) [/mnt/sdb1/ruby/trunk/test/ruby/test_float.rb:608]:
[TypeError] exception expected, not
Class:
Message: <"can't modify frozen Float">
---Backtrace---

/mnt/sdb1/ruby/trunk/test/ruby/test_float.rb:608:in `block in test_singleton_method'

[6113/11367] TestMarshal#test_float_extend = 0.00 s

4) Error:

test_float_extend(TestMarshal):
TypeError: can't define singleton
/mnt/sdb1/ruby/trunk/test/ruby/marshaltestlib.rb:224:in extend_object'
/mnt/sdb1/ruby/trunk/test/ruby/marshaltestlib.rb:224:inextend'
/mnt/sdb1/ruby/trunk/test/ruby/marshaltestlib.rb:224:in `test_float_extend'

[6115/11367] TestMarshal#test_float_ivar = 0.00 s

5) Error:

test_float_ivar(TestMarshal):
RuntimeError: can't modify frozen Float
/mnt/sdb1/ruby/trunk/test/ruby/marshaltestlib.rb:212:in block in test_float_ivar'
/mnt/sdb1/ruby/trunk/test/ruby/marshaltestlib.rb:212:ininstance_eval'
/mnt/sdb1/ruby/trunk/test/ruby/marshaltestlib.rb:212:in `test_float_ivar'

[6116/11367] TestMarshal#test_float_ivar_self = 0.00 s

6) Error:

test_float_ivar_self(TestMarshal):
RuntimeError: can't modify frozen Float
/mnt/sdb1/ruby/trunk/test/ruby/marshaltestlib.rb:218:in block in test_float_ivar_self'
/mnt/sdb1/ruby/trunk/test/ruby/marshaltestlib.rb:218:ininstance_eval'
/mnt/sdb1/ruby/trunk/test/ruby/marshaltestlib.rb:218:in `test_float_ivar_self'

Finished tests in 563.933874s, 20.1566 tests/s, 6086.8360 assertions/s.
11367 tests, 3432573 assertions, 2 failures, 4 errors, 29 skips

test-ruby spec

1)
String#% taints result for %s when argument is tainted ERROR
RuntimeError: can't modify frozen Float
/mnt/sdb1/ruby/trunk/spec/ruby spec/core/string/modulo_spec.rb:654:in
taint'
/mnt/sdb1/ruby/trunk/spec/ruby spec/core/string/modulo_spec.rb:654:in
block (2 levels) in '
/mnt/sdb1/ruby/trunk/spec/ruby spec/core/string/modulo_spec.rb:4:in
`

// SASADA Koichi at atdot dot net

#13 - 10/27/2012 10:54 AM - ko1 (Koichi Sasada)

(2012/10/27 10:02), SASADA Koichi wrote:

- 1 :sym "can't define singleton method (TypeError)"
• frozen object "can't modify frozen Float (RuntimeError)" RuntimeError

Float TypeError
frozen TypeError

frozen type RuntimeError
Fixnum frozen object TypeError

Float Bignum frozen

Index: include/ruby/ruby.h

```
--- include/ruby/ruby.h (revision 37336)
+++ include/ruby/ruby.h (working copy)
@@ -1133,7 +1133,7 @@ struct RBignum {
 (FL_TAINT | FL_UNTRUSTED); \
 } while (0)

-#define OBJ_FROZEN(x) (!FL_TEST((x), FL_FREEZE))
+#define OBJ_FROZEN(x) (!(FL_ABLE(x)?(RBASIC(x)->flags&(FL_FREEZE)):FLONUM_P(x)))
#define OBJ_FREEZE(x) FL_SET((x), FL_FREEZE)

#if SIZEOF_INT < SIZEOF_LONG
```

Index: class.c

```
--- class.c (revision 37336)
+++ class.c (working copy)
@@ -1322,6 +1322,10 @@ singleton_class_of(VALUE obj)
 rb_bug("unknown immediate %p", (void *)obj);
 return klass;
}

• else {
• if (BUILTIN_TYPE(obj) == T_FLOAT)
```

- rb_raise(rb_eTypeError, "can't define singleton");

- }

```
if (FL_TEST(RBASIC(obj)->klass, FL_SINGLETON) &&  
rb_ivar_get(RBASIC(obj)->klass, id_attached) == obj) {
```

Index: numeric.c

```
--- numeric.c (revision 37336)  
+++ numeric.c (working copy)  
@@ -620,6 +620,7 @@ rb_float_new_in_heap(double d)  
NEWOBJ_OF(flt, struct RFloat, rb_cFloat, T_FLOAT);
```

```
flt->float_value = d;
```

- OBJ_FREEZE(flt);
return (VALUE)flt;
}

Index: vm.c

```
--- vm.c (revision 37336)  
+++ vm.c (working copy)  
@@ -1872,7 +1872,7 @@ vm_define_method(rb_thread_t *th, VALUE  
}
```

```
if (is_singleton) {
```

- if (FIXNUM_P(obj) || FLONUM_P(obj) || SYMBOL_P(obj)) {
- if (FIXNUM_P(obj) || SYMBOL_P(obj) || CLASS_OF(obj) == rb_cFloat) {
rb_raise(rb_eTypeError,
"can't define singleton method \"%s\" for %s",
rb_id2name(id), rb_obj_classname(obj));

Index: test/ruby/marshaltestlib.rb

```
--- test/ruby/marshaltestlib.rb (revision 37336)  
+++ test/ruby/marshaltestlib.rb (working copy)  
@@ -207,30 +207,6 @@ module MarshalTestLib  
marshal_equal(NegativeZero) {|o| 1.0/o}  
end
```

- def test_float_ivar

- o1 = 1.23

- o1.instance_eval { @iv = 1 }

- marshal_equal(o1) {|o| o.instance_eval { @iv }}

- end

- def test_float_ivar_self
- o1 = 5.5
- o1.instance_eval { @iv = o1 }
- marshal_equal(o1) {|o| o.instance_eval { @iv }}

- end

- def test_float_extend
- o1 = 0.0/0.0
- o1.extend(Mod1)
- marshal_equal(o1) { |o|
(class << self; self; end).ancestors
- }
- o1.extend(Mod2)

- vm.c (vm_define_method): ditto.
- test/ruby/marshaltestlib.rb: catch up above changes.
- test/ruby/test_class.rb: ditto.
- test/test_pp.rb: ditto.

#15 - 10/27/2012 11:23 AM - ko1 (Koichi Sasada)

(2012/09/05 11:09), naruse (Yui NARUSE) wrote:

```
□□□□□□□□□□□□
```

```
□□□□□□□□□□□□□□□□□□□□□□□□□□
```

```
p 1.frozen? #=> false  
p 1.0.frozen? #=> true
```

```
□□□□□ Float □ frozen □□□□□□□□ Fixnum □ frozen □□□□□□□□□□
```

```
--  
// SASADA Koichi at atdot dot net
```

#16 - 10/27/2012 11:23 AM - ko1 (Koichi Sasada)

(2012/10/27 10:45), SASADA Koichi wrote:

```
□rubyspec □□□□□□□□□□□□□□□□□□□□□□□□□□
```

```
("%s" % -0.0.taint).tainted?.should == true # -0.0 is not flonum
```

```
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
```

```
# -0.0 □ flonum □□□□□□□□□□ ad-hoc □□□□□□□□□□□□□□
```

```
--  
// SASADA Koichi at atdot dot net
```

#17 - 10/29/2012 11:25 AM - naruse (Yui NARUSE)

ko1 (Koichi Sasada) wrote:

(2012/10/27 10:45), SASADA Koichi wrote:

```
□rubyspec □□□□□□□□□□□□□□□□□□□□□□□□□□
```

```
("%s" % -0.0.taint).tainted?.should == true # -0.0 is not flonum
```

```
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
```

```
# -0.0 □ flonum □□□□□□□□□□ ad-hoc □□□□□□□□□□□□□□
```

```
□□□□□□□□□□□□□□□□□□□□□□□□□□
```