

## Ruby trunk - Bug #15118

### Method [] & []= does not respect frozen\_string\_literal: true comment

09/14/2018 06:57 AM - chopraanmol1 (Anmol Chopra)

<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>		
<b>Target version:</b>		
<b>ruby -v:</b>	ruby 2.6.0dev (2018-09-13 trunk 64736) [x86_64-linux]	<b>Backport:</b> 2.3: REQUIRED, 2.4: DONE, 2.5: DONE

#### Description

Calling ["something"] on object or proc (non-hash aref implementation) does not respect frozen\_string\_literal: true comment

#### Script:

```
# frozen_string_literal: true

require 'benchmark'
require 'memory_profiler'
class NopId
  def self.[](str)
    str.__id__
  end

  def self.[]=(str, val)
    str.__id__
  end
end

SampleHash = {"sometext" => 0}

NopProc = proc{|a| a.__id__}

N = 1_000_000

def method1
  NopId["sometext"]
end

def method2
  SampleHash["sometext"]
end

def method3
  NopProc["sometext"]
end

def method4
  NopId["sometext"] = 'othertext'
end

def print_iseq method_name
  puts "-"*20
  puts RubyVM::InstructionSequence.disasm method(method_name)
  puts "-"*20
end

def print_memory_profiler title, &block
  puts "-"*20
  puts title
  MemoryProfiler.report{N.times(&block)}.pretty_print(detailed_report: false, allocated_strings: 0, retained_strings: 0)
```

```

puts "-+"*20
end

print_iseq :method1
print_iseq :method2
print_iseq :method3
print_iseq :method4

Benchmark.bm(10) do |bm|
  bm.report("method[]"){ N.times{ method1 } }
  bm.report("hash[]"){ N.times{ method2 } }
  bm.report("proc[]"){ N.times{ method3 } }
  bm.report("method[]="){ N.times{ method4 } }
end

```

```

print_memory_profiler("method[]"){method1}
print_memory_profiler("hash[]"){method2}
print_memory_profiler("proc[]"){method3}
print_memory_profiler("method[]="){method4}

```

### Output:

```

-+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+
== disasm: #<ISeq:method1@../test_aref.rb:21 (21,0)-(23,3)> (catch: FALSE)
0000 getinlinecache      7, <is:0>                ( 22) [LiCa]
0003 getconstant        :NopId
0005 setinlinecache     <is:0>
0007 opt_aref_with      "sometext", <callinfo!mid:[], argc:1, ARGS_SIMPLE>, <callcache>
0011 leave                ( 23) [Re]
-+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+
-+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+
== disasm: #<ISeq:method2@../test_aref.rb:25 (25,0)-(27,3)> (catch: FALSE)
0000 getinlinecache      7, <is:0>                ( 26) [LiCa]
0003 getconstant        :SampleHash
0005 setinlinecache     <is:0>
0007 opt_aref_with      "sometext", <callinfo!mid:[], argc:1, ARGS_SIMPLE>, <callcache>
0011 leave                ( 27) [Re]
-+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+
-+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+
== disasm: #<ISeq:method3@../test_aref.rb:29 (29,0)-(31,3)> (catch: FALSE)
0000 getinlinecache      7, <is:0>                ( 30) [LiCa]
0003 getconstant        :NopProc
0005 setinlinecache     <is:0>
0007 opt_aref_with      "sometext", <callinfo!mid:[], argc:1, ARGS_SIMPLE>, <callcache>
0011 leave                ( 31) [Re]
-+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+
-+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+
== disasm: #<ISeq:method4@../test_aref.rb:33 (33,0)-(35,3)> (catch: FALSE)
0000 getinlinecache      7, <is:0>                ( 34) [LiCa]
0003 getconstant        :NopId
0005 setinlinecache     <is:0>
0007 putobject           "othertext"
0009 swap
0010 topn                1
0012 opt_aset_with      "sometext", <callinfo!mid:[]=, argc:2, ARGS_SIMPLE>, <callcache>
0016 pop
0017 leave                ( 35) [Re]
-+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+
      user      system      total      real
method[]    0.120000    0.004000    0.124000 ( 0.121883)
hash[]      0.088000    0.000000    0.088000 ( 0.088723)
proc[]      0.132000    0.000000    0.132000 ( 0.133687)
method[]=   0.128000    0.000000    0.128000 ( 0.126702)
-+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+--+-+
method[]
Total allocated: 40000000 bytes (1000000 objects)

```

```
Total retained: 0 bytes (0 objects)
```

```
-----+
-----+
hash[]
Total allocated: 0 bytes (0 objects)
Total retained: 0 bytes (0 objects)
```

```
-----+
-----+
proc[]
Total allocated: 40000000 bytes (1000000 objects)
Total retained: 0 bytes (0 objects)
```

```
-----+
-----+
method[]=
Total allocated: 40000000 bytes (1000000 objects)
Total retained: 0 bytes (0 objects)
```

```
-----+
```

As you can observe calling NopClass["something"] & NopProc["something"] does not respect frozen\_string\_literal: true comment

### Patch:

<https://github.com/ruby/ruby/pull/1957>

### After Patch Result:

```
-----+
== disasm: #<ISeq:method1@../test_aref.rb:21 (21,0)-(23,3)> (catch: FALSE)
0000 getinlinecache      7, <is:0>                ( 22) [LiCa]
0003 getconstant        :NopId
0005 setinlinecache     <is:0>
0007 putobject          "sometext"
0009 opt_aref           <callinfo!mid:[], argc:1, ARGS_SIMPLE>, <callcache>
0012 leave               ( 23) [Re]
-----+
-----+
== disasm: #<ISeq:method2@../test_aref.rb:25 (25,0)-(27,3)> (catch: FALSE)
0000 getinlinecache      7, <is:0>                ( 26) [LiCa]
0003 getconstant        :SampleHash
0005 setinlinecache     <is:0>
0007 putobject          "sometext"
0009 opt_aref           <callinfo!mid:[], argc:1, ARGS_SIMPLE>, <callcache>
0012 leave               ( 27) [Re]
-----+
-----+
== disasm: #<ISeq:method3@../test_aref.rb:29 (29,0)-(31,3)> (catch: FALSE)
0000 getinlinecache      7, <is:0>                ( 30) [LiCa]
0003 getconstant        :NopProc
0005 setinlinecache     <is:0>
0007 putobject          "sometext"
0009 opt_aref           <callinfo!mid:[], argc:1, ARGS_SIMPLE>, <callcache>
0012 leave               ( 31) [Re]
-----+
-----+
== disasm: #<ISeq:method4@../test_aref.rb:33 (33,0)-(35,3)> (catch: FALSE)
0000 putnil              ( 34) [LiCa]
0001 getinlinecache      8, <is:0>
0004 getconstant        :NopId
```

```

0006 setinlinecache          <is:0>
0008 putobject               "sometext"
0010 putobject               "othertext"
0012 setn                    3
0014 opt_aset                <callinfo!mid:[]=, argc:2, ARGS_SIMPLE>, <callcache>
0017 pop
0018 leave                    ( 35) [Re]
-+-+-----+
      user      system      total      real
method[]  0.092000  0.004000  0.096000 ( 0.094751)
hash[]    0.088000  0.000000  0.088000 ( 0.087865)
proc[]    0.116000  0.000000  0.116000 ( 0.117047)
method[]= 0.096000  0.000000  0.096000 ( 0.096765)
-+-+-----+
method[]
Total allocated: 0 bytes (0 objects)
Total retained: 0 bytes (0 objects)

-+-+-----+
-+-+-----+
hash[]
Total allocated: 0 bytes (0 objects)
Total retained: 0 bytes (0 objects)

-+-+-----+
-+-+-----+
proc[]
Total allocated: 0 bytes (0 objects)
Total retained: 0 bytes (0 objects)

-+-+-----+
-+-+-----+
method[]=
Total allocated: 0 bytes (0 objects)
Total retained: 0 bytes (0 objects)

-+-+-----+

```

## Associated revisions

### Revision 64745 - 09/15/2018 02:20 AM - nobu (Nobuyoshi Nakada)

Use `opt_{aref,aset}` over `opt_{aref,aset}_with`

- `compile.c` (`iseq_compile_each0`): Use `opt_aref/opt_aset` over `opt_aref_with/opt_aset_with` when `frozen_string_literal: true`, not to resurrect the index string on non-Hash receiver.

[Fix GH-1957]

From: chopraanmol1 [chopraanmol1@gmail.com](mailto:chopraanmol1@gmail.com)

### Revision ff5475bf - 10/11/2018 03:12 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 64745: [Backport #15118]

Use `opt_{aref,aset}` over `opt_{aref,aset}_with`

- \* `compile.c` (`iseq_compile_each0`): Use ``opt_aref`/`opt_aset`` over ``opt_aref_with`/`opt_aset_with`` when `frozen_string_literal: true`, not to resurrect the index string on non-Hash receiver.

[Fix GH-1957]

From: chopraanmol1 <[chopraanmol1@gmail.com](mailto:chopraanmol1@gmail.com)>

git-svn-id: [svn+ssh://ci.ruby-lang.org/ruby/branches/ruby\\_2\\_5@65002](https://ci.ruby-lang.org/ruby/branches/ruby_2_5@65002) b2dd03c8-39d4-4d8f-98ff-823fe69b080e

## Revision 65002 - 10/11/2018 03:12 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 64745: [Backport #15118]

Use `opt_{aref,aset}` over `opt_{aref,aset}_with`

```
* compile.c (iseq_compile_each0): Use `opt_aref`/`opt_aset` over
`opt_aref_with`/`opt_aset_with` when frozen_string_literal: true,
not to resurrect the index string on non-Hash receiver.
```

[Fix GH-1957]

From: chopraanmoll <chopraanmoll@gmail.com>

## Revision a04a5fc1 - 10/17/2018 09:10 AM - usa (Usaku NAKAMURA)

merge revision(s) 64745: [Backport #15118]

Use `opt_{aref,aset}` over `opt_{aref,aset}_with`

```
* compile.c (iseq_compile_each0): Use `opt_aref`/`opt_aset` over
`opt_aref_with`/`opt_aset_with` when frozen_string_literal: true,
not to resurrect the index string on non-Hash receiver.
```

[Fix GH-1957]

From: chopraanmoll <chopraanmoll@gmail.com>

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby\_2\_4@65116 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

## Revision 65116 - 10/17/2018 09:10 AM - usa (Usaku NAKAMURA)

merge revision(s) 64745: [Backport #15118]

Use `opt_{aref,aset}` over `opt_{aref,aset}_with`

```
* compile.c (iseq_compile_each0): Use `opt_aref`/`opt_aset` over
`opt_aref_with`/`opt_aset_with` when frozen_string_literal: true,
not to resurrect the index string on non-Hash receiver.
```

[Fix GH-1957]

From: chopraanmoll <chopraanmoll@gmail.com>

## History

---

### #1 - 09/14/2018 07:05 AM - chopraanmol1 (Anmol Chopra)

- Description updated

### #2 - 09/14/2018 09:44 AM - shevegen (Robert A. Heiler)

Interesting.

I just tested with this code:

```
#!/System/Index/bin/ruby -w
# Encoding: ISO-8859-1
# frozen_string_literal: true
# ===== #
NopProc = proc {|a| a }

def foobar
  result = NopProc["sometext"]
  result
end

x = foobar

puts x.class
puts x.frozen?

y = 'abc'
puts y.frozen?

# Result:
```

```
# String
# false
# true
```

And I think you are right. I am testing `.frozen?` twice; the `'abc'` string is indeed frozen whereas the variant returned by `[]` does not seem to honour the instruction in the "magic" comment section.

Very good catch. May I ask, for curiosity, how you discovered it?  
I assume you may have tested somehow systematically or something?

### #3 - 09/14/2018 10:43 AM - chopraanmol1 (Anmol Chopra)

shevegen (Robert A. Heiler) wrote:

Very good catch. May I ask, for curiosity, how you discovered it?  
I assume you may have tested somehow systematically or something?

I was working on reducing memory allocation on `roo` gem. While profiling (with `memory_profiler` gem) I observed string literal being allocated multiple time at specific locations even after adding magic string literal comment. After debugging for while I discovered calling this method <https://github.com/sparklemotion/nokogiri/blob/7b8cd0f5b15a926e92c869b450dd6f71cdd17b61/lib/nokogiri/xml/node.rb#L120> in `cell_xml["r"]` fashion resulted into above behavior.

### #4 - 09/14/2018 11:37 AM - chopraanmol1 (Anmol Chopra)

- Description updated

### #5 - 09/14/2018 11:40 AM - chopraanmol1 (Anmol Chopra)

- Subject changed from `Method [] does not respect frozen_string_literal: true` comment to `Method [] & []= does not respect frozen_string_literal: true` comment

### #6 - 09/15/2018 03:32 AM - chopraanmol1 (Anmol Chopra)

Fixed in

<https://bugs.ruby-lang.org/projects/ruby-trunk/repository/revisions/64745>

### #7 - 09/17/2018 09:56 AM - Hanmac (Hans Mackowiak)

[nobu \(Nobuyoshi Nakada\)](#) : should this ticket be closed or not yet?

it seems fixed in 64745

### #8 - 09/17/2018 11:35 PM - nobu (Nobuyoshi Nakada)

- Backport changed from 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN to 2.3: REQUIRED, 2.4: REQUIRED, 2.5: REQUIRED

- Status changed from Open to Closed

- Description updated

Thank you, I missed the reference to this ticket in the commit log.

### #9 - 10/11/2018 03:12 PM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 2.3: REQUIRED, 2.4: REQUIRED, 2.5: REQUIRED to 2.3: REQUIRED, 2.4: REQUIRED, 2.5: DONE

`ruby_2_5` r65002 merged revision(s) 64745.

### #10 - 10/17/2018 09:10 AM - usa (Usaku NAKAMURA)

- Backport changed from 2.3: REQUIRED, 2.4: REQUIRED, 2.5: DONE to 2.3: REQUIRED, 2.4: DONE, 2.5: DONE

`ruby_2_4` r65116 merged revision(s) 64745.