

Ruby trunk - Bug #9223

Hash#reject!.size does not reflect changes to the hash

12/07/2013 12:25 AM - dmarcotte (Daniel Marcotte)

Status:	Closed	
Priority:	Normal	
Assignee:	matz (Yukihiro Matsumoto)	
Target version:	2.1.0	
ruby -v:	ruby 2.0.0p353	Backport: 1.9.3: UNKNOWN, 2.0.0: UNKNOWN
Description		
Here's an example demonstrating the issue, comparing to the regular reject behavior:		
<pre>h = {a: 'A', b: 'B'} reject_enum = h.reject reject_bang_enum = h.reject! h[:c] = 'C' p reject_enum.size # 3 p reject_bang_enum.size # 2</pre>		

Associated revisions

Revision 7416073c - 12/07/2013 02:16 AM - nobu (Nobuyoshi Nakada)

hash.c: rb_hash_reject without dup

- hash.c (rb_hash_reject): copy unrejected elements only to new hash, so that the change on the original receiver can affect. [ruby-core:58914] [Bug #9223]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@44047 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 44047 - 12/07/2013 02:16 AM - nobu (Nobuyoshi Nakada)

hash.c: rb_hash_reject without dup

- hash.c (rb_hash_reject): copy unrejected elements only to new hash, so that the change on the original receiver can affect. [ruby-core:58914] [Bug #9223]

Revision 44047 - 12/07/2013 02:16 AM - nobu (Nobuyoshi Nakada)

hash.c: rb_hash_reject without dup

- hash.c (rb_hash_reject): copy unrejected elements only to new hash, so that the change on the original receiver can affect. [ruby-core:58914] [Bug #9223]

Revision 44047 - 12/07/2013 02:16 AM - nobu (Nobuyoshi Nakada)

hash.c: rb_hash_reject without dup

- hash.c (rb_hash_reject): copy unrejected elements only to new hash, so that the change on the original receiver can affect. [ruby-core:58914] [Bug #9223]

Revision 44047 - 12/07/2013 02:16 AM - nobu (Nobuyoshi Nakada)

hash.c: rb_hash_reject without dup

- hash.c (rb_hash_reject): copy unrejected elements only to new hash, so that the change on the original receiver can affect. [ruby-core:58914] [Bug #9223]

Revision 44047 - 12/07/2013 02:16 AM - nobu (Nobuyoshi Nakada)

hash.c: rb_hash_reject without dup

- hash.c (rb_hash_reject): copy unrejected elements only to new hash, so that the change on the original receiver can affect. [ruby-core:58914] [Bug #9223]

Revision 44047 - 12/07/2013 02:16 AM - nobu (Nobuyoshi Nakada)

hash.c: rb_hash_reject without dup

- hash.c (rb_hash_reject): copy unrejected elements only to new hash, so that the change on the original receiver can affect. [ruby-core:58914] [Bug #9223]

Revision 740535f8 - 12/11/2013 07:01 AM - nobu (Nobuyoshi Nakada)

hash.c: reject should return a plain hash

- hash.c (rb_hash_reject): return a plain hash, without copying the class, default value, instance variables, and taintedness. they had been copied just by accident. [ruby-core:59045] [Bug #9223]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@44137 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 44137 - 12/11/2013 07:01 AM - nobu (Nobuyoshi Nakada)

hash.c: reject should return a plain hash

- hash.c (rb_hash_reject): return a plain hash, without copying the class, default value, instance variables, and taintedness. they had been copied just by accident. [ruby-core:59045] [Bug #9223]

Revision 44137 - 12/11/2013 07:01 AM - nobu (Nobuyoshi Nakada)

hash.c: reject should return a plain hash

- hash.c (rb_hash_reject): return a plain hash, without copying the class, default value, instance variables, and taintedness. they had been copied just by accident. [ruby-core:59045] [Bug #9223]

Revision 44137 - 12/11/2013 07:01 AM - nobu (Nobuyoshi Nakada)

hash.c: reject should return a plain hash

- hash.c (rb_hash_reject): return a plain hash, without copying the class, default value, instance variables, and taintedness. they had been copied just by accident. [ruby-core:59045] [Bug #9223]

Revision 44137 - 12/11/2013 07:01 AM - nobu (Nobuyoshi Nakada)

hash.c: reject should return a plain hash

- hash.c (rb_hash_reject): return a plain hash, without copying the class, default value, instance variables, and taintedness. they had been copied just by accident. [ruby-core:59045] [Bug #9223]

Revision 44137 - 12/11/2013 07:01 AM - nobu (Nobuyoshi Nakada)

hash.c: reject should return a plain hash

- hash.c (rb_hash_reject): return a plain hash, without copying the class, default value, instance variables, and taintedness. they had been copied just by accident. [ruby-core:59045] [Bug #9223]

Revision 44137 - 12/11/2013 07:01 AM - nobu (Nobuyoshi Nakada)

hash.c: reject should return a plain hash

- hash.c (rb_hash_reject): return a plain hash, without copying the class, default value, instance variables, and taintedness. they had been copied just by accident. [ruby-core:59045] [Bug #9223]

Revision 571955c1 - 12/17/2013 11:15 AM - nobu (Nobuyoshi Nakada)

hash.c: revert

- hash.c (rb_hash_reject): revert to deprecated behavior, with warnings, due to compatibility for HashWithDifferentAccess. [ruby-core:59154] [Bug #9223]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@44263 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 44263 - 12/17/2013 11:15 AM - nobu (Nobuyoshi Nakada)

hash.c: revert

- hash.c (rb_hash_reject): revert to deprecated behavior, with warnings, due to compatibility for HashWithDifferentAccess. [ruby-core:59154] [Bug #9223]

Revision 44263 - 12/17/2013 11:15 AM - nobu (Nobuyoshi Nakada)

hash.c: revert

- hash.c (rb_hash_reject): revert to deprecated behavior, with warnings, due to compatibility for HashWithDifferentAccess. [ruby-core:59154] [Bug #9223]

#9223]

Revision 44263 - 12/17/2013 11:15 AM - nobu (Nobuyoshi Nakada)

hash.c: revert

- hash.c (rb_hash_reject): revert to deprecated behavior, with warnings, due to compatibility for HashWithDifferentAccess. [ruby-core:59154] [Bug #9223]

Revision 44263 - 12/17/2013 11:15 AM - nobu (Nobuyoshi Nakada)

hash.c: revert

- hash.c (rb_hash_reject): revert to deprecated behavior, with warnings, due to compatibility for HashWithDifferentAccess. [ruby-core:59154] [Bug #9223]

Revision 44263 - 12/17/2013 11:15 AM - nobu (Nobuyoshi Nakada)

hash.c: revert

- hash.c (rb_hash_reject): revert to deprecated behavior, with warnings, due to compatibility for HashWithDifferentAccess. [ruby-core:59154] [Bug #9223]

Revision 44263 - 12/17/2013 11:15 AM - nobu (Nobuyoshi Nakada)

hash.c: revert

- hash.c (rb_hash_reject): revert to deprecated behavior, with warnings, due to compatibility for HashWithDifferentAccess. [ruby-core:59154] [Bug #9223]

Revision e1536a74 - 12/30/2013 09:11 PM - naruse (Yui NARUSE)

fix ruby/test_hash.rb [ruby-core:59154] [Bug #9223]

the behavior is under 2.2.0

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_1@44466 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 12/07/2013 09:41 AM - nobu (Nobuyoshi Nakada)

Seems inverse.

Hash#reject is equivalent to the following ruby code:

```
def reject(&block)
  dup.delete_if(&block)
end
```

That is, the change on the original receiver can't affect the result.

#2 - 12/07/2013 11:16 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r44047.

Daniel, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

hash.c: rb_hash_reject without dup

- hash.c (rb_hash_reject): copy unrejected elements only to new hash, so that the change on the original receiver can affect. [ruby-core:58914] [Bug #9223]

#3 - 12/11/2013 03:49 PM - tmm1 (Aman Gupta)

r44047 introduced a small change in behavior. Before, reject would copy ivars and default proc into the return hash:

```
h = {}
h.instance_variable_set(:@foo, 1)
p h.instance_variable_get(:@foo)    #=> 1
```

```
p h.reject{true}.instance_variable_get(:@foo) #=> 1
```

Now, only the class is copied. This can cause subtle issues when a subclass of Hash is used (if it employs ivars).

Note that both the old and new behavior here are completely different than Hash#select, which always returns a plain Hash with no ivars.

#4 - 12/12/2013 04:31 PM - nagachika (Tomoyuki Chikanaga)

- Status changed from Closed to Assigned
- Assignee set to matz (Yukihiro Matsumoto)

Hello,

r44137 introduce incompatible spec change of Hash#reject.

After r44137, Hash#reject return new hash and doesn't copied class, ivars, default value, taintedness from receiver.

ChangeLog says 'they had been copied just by accident.'

Even if it is the truth, hsh.reject is equivalent to hsh.dup.reject! for very long time and it is documented explicitly. There's a such description since ruby 1.8.1.

https://github.com/ruby/ruby/blob/v1_8_1/hash.c#L734

I think this change needs approval by matz, or at least by 2.1 release manager (naruse san).

#5 - 12/16/2013 05:31 PM - naruse (Yui NARUSE)

- Due date set to 12/16/2013
- Category set to core
- Priority changed from Normal to 7
- Target version set to 2.1.0
- % Done changed from 100 to 80

This blocks 2.1.0-rc

#6 - 12/16/2013 11:57 PM - matz (Yukihiro Matsumoto)

I accept this behavior change. #reject should not copy instance variables etc. just like #select.

Matz.

#7 - 12/17/2013 01:26 AM - naruse (Yui NARUSE)

NEWS says

+* Hash

- * incompatible changes:
- * Hash#reject now returns plain Hash object, that is the original object's
- subclass, instance variables, default value, and taintedness are no longer
- copied.

but test/ruby/test_hash.rb checks

```
assert_instance_of(Hash, h)
```

#8 - 12/17/2013 11:23 AM - nobu (Nobuyoshi Nakada)

(13/12/17 1:27), naruse (Yui NARUSE) wrote:

+* Hash

- * incompatible changes:
- * Hash#reject now returns plain Hash object, that is the original object's
- subclass, instance variables, default value, and taintedness are no longer
- copied.

but test/ruby/test_hash.rb checks

```
assert_instance_of(Hash, h)
```

Right.

The result of Hash#reject is always an instance of Hash, not SubHash, now.

Better descriptions are always welcome, of course.

#9 - 12/17/2013 03:44 PM - naruse (Yui NARUSE)

nobu (Nobuyoshi Nakada) wrote:

(13/12/17 1:27), naruse (Yui NARUSE) wrote:

+* Hash

- * incompatible changes:
- * Hash#reject now returns plain Hash object, that is the original object's
- subclass, instance variables, default value, and taintedness are no longer
- copied.

but test/ruby/test_hash.rb checks

```
assert_instance_of(Hash, h)
```

Right.

The result of Hash#reject is always an instance of Hash, not SubHash, now.

Better descriptions are always welcome, of course.

The change (SubClass) is not allowed for 2.1 because it breaks HashWithDifferentAccess.

#10 - 12/17/2013 08:15 PM - nobu (Nobuyoshi Nakada)

- Status changed from Assigned to Closed

- % Done changed from 80 to 100

This issue was solved with changeset r44263.

Daniel, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

hash.c: revert

- hash.c (rb_hash_reject): revert to deprecated behavior, with warnings, due to compatibility for HashWithDifferentAccess. [ruby-core:59154] [Bug [#9223](#)]