

## Ruby master - Bug #14380

### Expected transform\_keys! to work just as transform\_keys, but it doesn't

01/20/2018 07:03 PM - law (Tomasz Wegrzanowski)

<b>Status:</b> Closed	
<b>Priority:</b> Normal	
<b>Assignee:</b> matz (Yukihiro Matsumoto)	
<b>Target version:</b>	
<b>ruby -v:</b> ruby 2.5.0p0 (2017-12-25 revision 61468) [x86_64-darwin17]	<b>Backport:</b> 2.3: DONTNEED, 2.4: DONTNEED, 2.5: DONE

#### Description

This seriously violates the Principle of Least Surprise to me:

```
{1 => :a, -1 => :b}.transform_keys{|k| -k} #=> {-1=>:a, 1=>:b}
{1 => :a, -1 => :b}.transform_keys!{|k| -k} # => {1=>:a}
```

# This fails:

```
ht=(1..10).map{|k| [k,k]}.to_h; ht.transform_keys(&:succ) # => {2=>1, 3=>2, 4=>3, 5=>4, 6=>5, 7=>6, 8=>7, 9=>8, 10=>9, 11=>10}
ht=(1..10).map{|k| [k,k]}.to_h; ht.transform_keys!(&:succ) # => {11=>1}
```

# This code with same issue works just because of key ordering:

```
ht=(1..10).map{|k| [k,k]}.to_h; ht.transform_keys(&:pred) #=> {0=>1, 1=>2, 2=>3, 3=>4, 4=>5, 5=>6, 6=>7, 7=>8, 8=>9, 9=>10}
ht=(1..10).map{|k| [k,k]}.to_h; ht.transform_keys!(&:pred) #=> {0=>1, 1=>2, 2=>3, 3=>4, 4=>5, 5=>6, 6=>7, 7=>8, 8=>9, 9=>10}
```

Of course in these examples it's very easy to see the problem, but in bigger programs it could be really difficult.

If the implementation instead did equivalent of:

```
class Hash
  def transform_values!(&block)
    replace transform_values(&block)
  end
end
```

it would be much less surprising.

Hash#transform\_keys / Hash#transform\_keys! inherently require that resulting values don't collide, but in these examples it works in surprising ways even though there's no collision between results.

#### Associated revisions

##### Revision 52bb93c2 - 01/26/2018 01:33 AM - mrkn (Kenta Murata)

hash.c: support key swapping in Hash#transform\_keys!

- hash.c (rb\_hash\_transform\_keys\_bang): support key swapping in Hash#transform\_keys!  
[Bug #14380] [ruby-core:84951]
- test/ruby/test\_hash.rb (test\_transform\_keys\_bang):  
add assertions for this change

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@62042 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 62042 - 01/26/2018 01:33 AM - mrkn (Kenta Murata)

hash.c: support key swapping in Hash#transform\_keys!

- hash.c (rb\_hash\_transform\_keys\_bang): support key swapping in Hash#transform\_keys!  
[Bug #14380] [ruby-core:84951]

- test/ruby/test\_hash.rb (test\_transform\_keys\_bang):  
add assertions for this change

#### Revision 62042 - 01/26/2018 01:33 AM - mrkn (Kenta Murata)

hash.c: support key swapping in Hash#transform\_keys!

- hash.c (rb\_hash\_transform\_keys\_bang): support key swapping in Hash#transform\_keys!  
[Bug #14380] [ruby-core:84951]
- test/ruby/test\_hash.rb (test\_transform\_keys\_bang):  
add assertions for this change

#### Revision 9c7caa3b - 01/26/2018 02:01 AM - mrkn (Kenta Murata)

Fix rubyspec against the change in Hash#transform\_keys!

[Bug #14380] [ruby-core:84951]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@62044 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

#### Revision 62044 - 01/26/2018 02:01 AM - mrkn (Kenta Murata)

Fix rubyspec against the change in Hash#transform\_keys!

[Bug #14380] [ruby-core:84951]

#### Revision 62044 - 01/26/2018 02:01 AM - mrkn (Kenta Murata)

Fix rubyspec against the change in Hash#transform\_keys!

[Bug #14380] [ruby-core:84951]

#### Revision 54717626 - 03/22/2018 07:00 AM - naruse (Yui NARUSE)

merge revision(s) 62042,62044: [Backport #14380]

```
hash.c: support key swapping in Hash#transform_keys!
```

```
* hash.c (rb_hash_transform_keys_bang): support key swapping in  
Hash#transform_keys!  
[Bug #14380] [ruby-core:84951]
```

```
* test/ruby/test_hash.rb (test_transform_keys_bang):  
add assertions for this change
```

```
Fix rubyspec against the change in Hash#transform_keys!
```

```
[Bug #14380] [ruby-core:84951]
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby\_2\_5@62889 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

#### Revision 62889 - 03/22/2018 07:00 AM - naruse (Yui NARUSE)

merge revision(s) 62042,62044: [Backport #14380]

```
hash.c: support key swapping in Hash#transform_keys!
```

```
* hash.c (rb_hash_transform_keys_bang): support key swapping in  
Hash#transform_keys!  
[Bug #14380] [ruby-core:84951]
```

```
* test/ruby/test_hash.rb (test_transform_keys_bang):  
add assertions for this change
```

```
Fix rubyspec against the change in Hash#transform_keys!
```

```
[Bug #14380] [ruby-core:84951]
```

## History

---

## #1 - 01/20/2018 07:44 PM - shevegen (Robert A. Heiler)

Here is the current documentation:

[http://ruby-doc.org/core-2.5.0/Hash.html#method-i-transform\\_keys](http://ruby-doc.org/core-2.5.0/Hash.html#method-i-transform_keys)

After looking at the example you gave, I can not say whether this is deliberate or not - but either way, I think IF the behaviour is retained as-is, then it should at the least be documented too.

In particular this code surprised me too:

```
hash = {1 => :a, -1 => :b}
pp hash.transform_keys{|k| -k} # => {-1=>:a, 1=>:b}
hash.transform_keys!{|k| -k}
pp hash # => {1=>:a}
```

I would have assumed that the method with the ! acts as the method without ! but this is not the case. It may probably not be a bug (I don't know though), but it should be documented either way. People may look at the above code and ask "what happened to key -1, if all keys are flipped from negative to positive and vice versa?"

To your code example of an alternative, I am confused too.

Why do you have a method called "transform\_values!" if the values that are modified are on the keys, as exemplified by the name "transform\_keys!"?

## #2 - 01/21/2018 10:24 PM - taw (Tomasz Wegrzanowski)

Oops, I meant to suggest this, accidentally said "values" instead of "keys" :

```
class Hash
  def transform_keys!(&block)
    replace transform_keys(&block)
  end
end
```

## #3 - 01/21/2018 11:24 PM - phluid61 (Matthew Kerwin)

taw (Tomasz Wegrzanowski) wrote:

Oops, I meant to suggest this, accidentally said "values" instead of "keys" :

```
class Hash
  def transform_keys!(&block)
    replace transform_keys(&block)
  end
end
```

IIRC this was discussed when the feature was originally proposed. What happens to a break inside the block?

## #4 - 01/22/2018 03:42 AM - shyouhei (Shyouhei Urabe)

phluid61 (Matthew Kerwin) wrote:

IIRC this was discussed when the feature was originally proposed.

Here you are. <https://bugs.ruby-lang.org/issues/13583>

## #5 - 01/24/2018 08:02 AM - mrkn (Kenta Murata)

- Assignee set to mrkn (Kenta Murata)
- Status changed from Open to Assigned

## #6 - 01/26/2018 01:33 AM - mrkn (Kenta Murata)

- Status changed from Assigned to Closed

Applied in changeset [trunk|r62042](https://github.com/ruby/ruby/commit/trunk|r62042).

---

hash.c: support key swapping in Hash#transform\_keys!

- hash.c (rb\_hash\_transform\_keys\_bang): support key swapping in Hash#transform\_keys!  
[Bug #14380] [ruby-core:84951]
- test/ruby/test\_hash.rb (test\_transform\_keys\_bang):  
add assertions for this change

**#7 - 01/29/2018 02:27 PM - Eregon (Benoit Daloze)**

Will this be backported to 2.5?

**#8 - 01/29/2018 08:33 PM - marcandre (Marc-Andre Lafortune)**

- Assignee changed from mrkn (Kenta Murata) to matz (Yukihiro Matsumoto)
- Status changed from Closed to Open

I raised this issue previously <https://bugs.ruby-lang.org/issues/13583#note-8>

This is a spec change. Moreover it introduces incompatibilities with ActiveSupport.

Matz: final verdict please?

**#9 - 02/03/2018 10:42 PM - taw (Tomasz Wegrzanowski)**

marcandre (Marc-Andre Lafortune) wrote:

I raised this issue previously <https://bugs.ruby-lang.org/issues/13583#note-8>

This is a spec change. Moreover it introduces incompatibilities with ActiveSupport.

I'd say that ActiveSupport is testing group for possible new ruby features, and when getting them into ruby core library, it's good time to clean up their edge cases.

**#10 - 03/15/2018 07:21 AM - usa (Usaku NAKAMURA)**

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

**#11 - 03/20/2018 02:18 PM - naruse (Yui NARUSE)**

- Status changed from Open to Closed

Considered a bug and the behavior change will also be applied in ActiveSupport.

**#12 - 03/20/2018 03:41 PM - rafaelfranca (Rafael França)**

change will also be applied in ActiveSupport

Where did you get that information?

This change silently breaks applications when upgrading to 2.6 version. Backward compatibility not only with active support but with an already released version of Ruby should be took in consideration here.

I don't think the Rails team would accept this change as a bug fix.

**#13 - 03/22/2018 07:00 AM - naruse (Yui NARUSE)**

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

ruby\_2\_5 r62889 merged revision(s) 62042,62044.