

Ruby trunk - Feature #9239

Array#to_h ignores flat arrays

12/11/2013 03:26 AM - sawa (Tsuyoshi Sawada)

Status:	Rejected	
Priority:	Normal	
Assignee:	naruse (Yui NARUSE)	
Target version:	2.1.0	
Description		
=begin My understanding was that the newly introduced Array#to_h was to be equivalent to Hash[]. But when I give a flat array, it returns an empty hash: <pre>[:a, :b].to_h # => {} [:a].to_h # => {}</pre> I expected { :a => :b } for the first example, and ArgumentError: odd number of arguments for Hash for the second example, as is with Hash[]. Was it intended to be different? If so, why is that the case? Even if it is to be different, it should at least raise an error when a flat array is given. =end		
Related issues:		
Related to Backport21 - Backport #9270: Array#to_h should not ignore badly fo...		Closed 12/21/2013

Associated revisions

Revision 9471f418 - 12/23/2013 03:42 AM - marcandre (Marc-Andre Lafortune)

- array.c: Have to_h raise on elements that are not key-value pairs [#9239]
- enum.c: ditto

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@44354 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 44354 - 12/23/2013 03:42 AM - marcandre (Marc-Andre Lafortune)

- array.c: Have to_h raise on elements that are not key-value pairs [#9239]
- enum.c: ditto

Revision 44354 - 12/23/2013 03:42 AM - marcandre (Marc-Andre Lafortune)

- array.c: Have to_h raise on elements that are not key-value pairs [#9239]
- enum.c: ditto

Revision 44354 - 12/23/2013 03:42 AM - marcandre (Marc-Andre Lafortune)

- array.c: Have to_h raise on elements that are not key-value pairs [#9239]
- enum.c: ditto

Revision 44354 - 12/23/2013 03:42 AM - marcandre (Marc-Andre Lafortune)

- array.c: Have to_h raise on elements that are not key-value pairs [#9239]
- enum.c: ditto

Revision 44354 - 12/23/2013 03:42 AM - marcandre (Marc-Andre Lafortune)

- array.c: Have to_h raise on elements that are not key-value pairs [#9239]
- enum.c: ditto

Revision 44354 - 12/23/2013 03:42 AM - marcandre (Marc-Andre Lafortune)

- array.c: Have to_h raise on elements that are not key-value pairs [#9239]
- enum.c: ditto

Revision 0f7e52f9 - 12/23/2013 07:12 AM - naruse (Yui NARUSE)

merge revision(s) 44354: [Backport #9270]

```
* array.c: Have to_h raise on elements that are not key-value pairs  
[#9239]
```

```
* enum.c: ditto
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_1@44361 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 12/11/2013 03:41 AM - marcandre (Marc-Andre Lafortune)

- *Tracker changed from Bug to Feature*

#2 - 12/11/2013 03:49 AM - marcandre (Marc-Andre Lafortune)

- *Assignee set to matz (Yukihiro Matsumoto)*

Moving to "feature", as the current documentation states that elements that are not 2-element arrays are ignored, so it is different from Hash.[]

My opinion is that the fact that Hash[1,2,3,4] # => {1 => 2, 3 => 4} is a misfeature and that this usage should not be encouraged.

On the other hand, Array#to_h could raise an error on elements that are not 2-element arrays. I argued otherwise before, but maybe that would be safer.

#3 - 12/11/2013 05:14 AM - alexeymuranov (Alexey Muranov)

If [:a, :b].to_h returned {:a => :b}, then what should [a.b](#), [\[:c, :d\].to_h](#) return? {[:a, :b] => [:c, :d]} ?

#4 - 12/11/2013 11:22 AM - sawa (Tsuyoshi Sawada)

marcandre (Marc-Andre Lafortune) wrote:

My opinion is that the fact that Hash[1,2,3,4] # => {1 => 2, 3 => 4} is a misfeature and that this usage should not be encouraged.

On the other hand, Array#to_h could raise an error on elements that are not 2-element arrays. I argued otherwise before, but maybe that would be safer.

I agree.

#5 - 12/15/2013 04:06 AM - marcandre (Marc-Andre Lafortune)

One reason I think I was wrong is that current form could encourage code like:

```
enum.map{|x| [x.foo, x.bar] if x.baz? }.to_h
```

using the fact that any nil will be ignored. I'm not sure that it's a good idea.

It would probably be safer to raise an Exception for elements that are not a key-value pair. It also satisfies fail-early principle.

Matz, what do you think?

#6 - 12/18/2013 07:07 AM - marcandre (Marc-Andre Lafortune)

- Category set to core

- Assignee changed from matz (Yukihiko Matsumoto) to marcandre (Marc-Andre Lafortune)

- Target version set to 2.1.0

I got the green light from Matz, thanks :-)

Naruse-san: could you please confirm that there is no problem on your end for me to commit the following patch:

https://github.com/marcandre/ruby/compare/to_h_raise

Thanks

#7 - 12/18/2013 08:18 AM - zzak (Zachary Scott)

I dont think we should commit this before 2.1, is there good reason?

#8 - 12/18/2013 12:45 PM - marcandre (Marc-Andre Lafortune)

Since this is a new feature, there's no risk of regression and there is no incompatibility. Committing after 2.1 could create incompatibility.

#9 - 12/18/2013 12:57 PM - zzak (Zachary Scott)

How would committing after 2.1 create incompatibility?

If we commit this feature to trunk after ruby_2_1 branch is created, then it won't be released until 2.2, right?

#10 - 12/18/2013 01:21 PM - marcandre (Marc-Andre Lafortune)

zzak (Zachary Scott) wrote:

How would committing after 2.1 create incompatibility?

If some rubyists write code for Ruby 2.1 using this new `Array#to_h` and rely on the fact that elements that are not key value pairs are ignored, then this change in 2.2 will break their code. This is what I mean by incompatibility.

#11 - 12/18/2013 10:54 PM - zzak (Zachary Scott)

I think you should ask for permission from naruse before you commit

#12 - 12/19/2013 12:06 AM - sawa (Tsuyoshi Sawada)

marcandre (Marc-Andre Lafortune),

Thanks for the patch to reflect my request, and thanks for ALREADY HAVING ASKED Naruse-san for the commit.

#13 - 12/19/2013 02:26 PM - marcandre (Marc-Andre Lafortune)

- Assignee changed from marcandre (Marc-Andre Lafortune) to naruse (Yui NARUSE)

- Priority changed from Normal to 6

#14 - 12/21/2013 01:02 AM - naruse (Yui NARUSE)

- Status changed from Open to Rejected

As `Array#to_h` says and `test_to_h` describes, `Array#to_h` is different from `Hash[]`.
The behavior is expected one.

Returns the result of interpreting ary as an array of [key, value] pairs. Elements other than pairs of values are ignored.

#15 - 12/21/2013 03:05 AM - marcandre (Marc-Andre Lafortune)

Yui, I'm wondering if you actually read the whole thread. In any case, I probably don't have to point out that Matz usually has the privilege of rejecting feature requests.

I've summarized it as [#9270](#).