

Ruby trunk - Feature #4862

Struct#to_hash

06/10/2011 11:29 AM - postmodern (Hal Brodigan)

Status:	Rejected	
Priority:	Normal	
Assignee:	matz (Yukihiro Matsumoto)	
Target version:		
Description		
Occasionally, it is necessary to convert a Struct to a Hash (especially when generating JSON from an Array of Structs). A Struct#to_hash method would be very useful in this situation.		
<pre>class Struct # # Returns the Hash representation of the members and values within the struct. # def to_hash new_hash = {} each_pair do member, value new_hash[member] = value end new_hash end end</pre>		
Related issues:		
Has duplicate Ruby trunk - Feature #5008: Equal rights for Hash (like Array, ...		Rejected 07/10/2011

Associated revisions

Revision 1dbe0f06 - 04/16/2012 03:16 AM - marcandre (Marc-Andre Lafortune)

- struct.c: Add Struct#to_h [Feature #6276] [ref #4862] [rubyspec:2082ef46d46e]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@35341 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 35341 - 04/16/2012 03:16 AM - marcandre (Marc-Andre Lafortune)

- struct.c: Add Struct#to_h [Feature #6276] [ref #4862] [rubyspec:2082ef46d46e]

Revision 35341 - 04/16/2012 03:16 AM - marcandre (Marc-Andre Lafortune)

- struct.c: Add Struct#to_h [Feature #6276] [ref #4862] [rubyspec:2082ef46d46e]

Revision 35341 - 04/16/2012 03:16 AM - marcandre (Marc-Andre Lafortune)

- struct.c: Add Struct#to_h [Feature #6276] [ref #4862] [rubyspec:2082ef46d46e]

Revision 35341 - 04/16/2012 03:16 AM - marcandre (Marc-Andre Lafortune)

- struct.c: Add Struct#to_h [Feature #6276] [ref #4862] [rubyspec:2082ef46d46e]

Revision 35341 - 04/16/2012 03:16 AM - marcandre (Marc-Andre Lafortune)

- struct.c: Add Struct#to_h [Feature #6276] [ref #4862] [rubyspec:2082ef46d46e]

Revision 35341 - 04/16/2012 03:16 AM - marcandre (Marc-Andre Lafortune)

- struct.c: Add Struct#to_h [Feature #6276] [ref #4862] [rubyspec:2082ef46d46e]

- struct.c: Add Struct#to_h [Feature #6276] [ref #4862] [rubyspec:2082ef46d46e]

History

#1 - 11/24/2011 04:08 AM - rossmeissl (Andy Rossmeissl)

```
def to_hash() Hash[members.zip(values)] end
```

#2 - 11/24/2011 05:03 AM - trans (Thomas Sawyer)

A Struct doesn't respond to hardly any Hash methods in general. So #to_h seems more appropriate.

#3 - 11/24/2011 05:09 AM - trans (Thomas Sawyer)

Then again maybe I'm wrong b/c I was just surprised by the fact that a Struct doesn't have (or at least act like it has) instance_variables.

```
Struct.new(:a).new(1).instance_variables #=> []
```

#4 - 11/24/2011 07:31 AM - Eregon (Benoit Daloze)

Thomas Sawyer wrote:

Then again maybe I'm wrong b/c I was just surprised by the fact that a Struct doesn't have (or at least act like it has) instance_variables.

```
Struct.new(:a).new(1).instance_variables #=> []
```

Yeah, it is implemented with an array stored in the RStruct struct in MRI.

A Struct#to_hash method would be very useful in this situation.

I agree.

#to_hash is the right method name because there is no #to_h (but indeed it is somewhat inconsistent with #to_i, #to_int, ...)

#5 - 11/24/2011 08:33 AM - trans (Thomas Sawyer)

Despite it's implementation, I thought Struct was intended to provide a convenient way to make quick classes. Which is why I assume it would have instance variables. I can understand though that it would be implemented in another fashion for speed. But I would still think the public interface to appear as if it were a typical sort of class.

Given it's current interface it hardly resembles a Hash at all. I think #each at least should act the same as a Hash if #to_hash would be added to it, but instead it acts like an Array:

```
s = Struct.new(:a, :b).new(1,2)
=> #
s.map{ |a| a }
=> [1, 2]
```

"because there is no #to_h (but indeed it is somewhat inconsistent with #to_i, #to_int, ...)"

Not just "somewhat". So I ignore all that heehawing and define #to_h wherever it makes sense when I need it. Why #to_h should be carved out as some odd exception seems to me just a lot of needless fuss over... what?

#6 - 11/24/2011 08:45 AM - drbrain (Eric Hodel)

Struct does use instance variables, you just can't see them from ruby because they don't start with @

#7 - 03/25/2012 04:18 PM - mame (Yusuke Endoh)

- Status changed from Open to Assigned

- Assignee set to matz (Yukihiro Matsumoto)

#8 - 04/10/2012 06:48 PM - matz (Yukihiro Matsumoto)

- Status changed from Assigned to Rejected

#to_hash protocol expects the object to be hash-compatible. Struct is not the case.

Matz.