

Ruby master - Feature #6277

Hash#convert_key

04/11/2012 01:12 AM - trans (Thomas Sawyer)

Status:	Assigned
Priority:	Normal
Assignee:	matz (Yukihiro Matsumoto)
Target version:	
Description	
<pre>=begin Many times a hash with uniform keys is needed (or is at least preferable) for a particular usecase. To this end I propose ((%Hash#convert_key%)). h = {} h.convert_key{ k k.to_sym } h['a'] = 1 h.update('b'=>2) h #=> {:a=>1, :b=>2} The idea is similar in concept to ((%#default_proc%)). Others solutions to fill this need have been tried and used, but almost exclusively are in the form of a new class. Most notable is Rails HashWithIndifferentAccess. But ((%#convert_key%)) has much greater flexibility since keys can be converted to anything. =end</pre>	

History

#1 - 04/13/2012 07:54 AM - mame (Yusuke Endoh)

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

#2 - 05/11/2012 03:08 AM - trans (Thomas Sawyer)

It occurs to me that this issue could be solved in another way, via Subclassing. A subclass of Hash could be made with certain methods overridden to convert the keys as needed. This would offer complete flexibility (where as callbacks are limited). The only problem is, Ruby's Hash class is not very internally connascent --many different methods access the underlying data --there is no one, or few, specific methods that can be overridden to handle all cases. This leads to a very unDRY and bloated subclass.

If Ruby's Hash could be modified to follow essentially a CRUD model, with single points-of-entry for the creation, reading, updating and deletion of entries, and all other methods route through these core methods for access to the underlying data, then it would be easy enough to subclass and implement the key conversion behaviour.

#3 - 05/17/2012 12:34 AM - trans (Thomas Sawyer)

I am going to create a separate issue fro the CRUDify of Hash. After working on the Hashery (<http://github.com/rubyworks/hashery>) project all this week, it is clear to me that both a key casting proc and a CRUDified Hash class are independently useful.

#4 - 10/25/2012 05:35 PM - yhara (Yutaka HARA)

- Target version changed from 2.0.0 to 2.6

#5 - 12/25/2017 06:15 PM - naruse (Yui NARUSE)

- Target version deleted (2.6)