Ruby master - Feature #3131
add Kernel#Hash() method like Kernel#Array()
04/12/2010 08:32 AM - sunaku (Suraj Kurapati)

Status: Closed
Priority: Normal
Assignee: matz (Yukihiro Matsumoto)
Target version: 2.0.0

Description
=begin
Hello,

There is an imbalance of power in the Ruby core API (when it comes
to arrays and hashes) because it is easier to convert nil values
into empty arrays, thanks to Kernel#Array(), than it is to convert
nil values into empty hashes, due to lack of Kernel#Hash().

To correct this asymmetry and restore a balance of power, please
add a Kernel#Hash() method for converting nil, Array, and Hash
values into hashes:

module Kernel
  def Hash(value)
    if value.respond_to?(:to_hash)
      value.to_hash
    elsif value.respond_to?(:to_ary)
      Hash[*value.to_ary]
    elsif value.nil?
      {}
    else
      raise ArgumentError, "invalid value for Hash: #{value}" end
  end
end

For example, here is how I would use the above API:

# CASE 1: to_hash
real_hash = {:real => true}
Hash(real_hash) # => {:real=>true}
fake_hash = Object.new
def fake_hash.to_hash
  {:fake => true}
end
Hash(fake_hash) # => {:fake=>true}

# CASE 2: to_ary
real_array = [:real, true]
Hash(real_array) # => {:real=>true}
fake_array = Object.new
def fake_array.to_ary
  [:fake, true]
end
Hash(fake_array) # => {:fake=>true}
# CASE 3: nil

Hash(nil) # => {}

# CASE 4: unsupported arguments

>> Hash(true)
ArgumentError: invalid value for Hash: true
  from (irb):74:in 'Hash'
  from (irb):80
  from /usr/bin/irb:12:in `<main>'

>> Hash(false)
ArgumentError: invalid value for Hash: false
  from (irb):74:in 'Hash'
  from (irb):81
  from /usr/bin/irb:12:in `<main>'

>> Hash(123)
ArgumentError: invalid value for Hash: 123
  from (irb):74:in 'Hash'
  from (irb):82
  from /usr/bin/irb:12:in `<main>'

Thanks for your consideration.
=.end

Related issues:

Follows Ruby master - Feature #5008: Equal rights for Hash (like Array, Strin...  Rejected  07/10/2011

Associated revisions

Revision 498838c6 - 01/24/2012 04:02 AM - matz (Yukihiro Matsumoto)

* object.c (rb_Hash): add Kernel#Hash conversion method like Array() or Float(). a patch from Run Paint Run Run. Fix #3131
git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@34367 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 34367 - 01/24/2012 04:02 AM - matz (Yukihiro Matsumoto)

* object.c (rb_Hash): add Kernel#Hash conversion method like Array() or Float(). a patch from Run Paint Run Run. Fix #3131

Revision 34367 - 01/24/2012 04:02 AM - matz (Yukihiro Matsumoto)

* object.c (rb_Hash): add Kernel#Hash conversion method like Array() or Float(). a patch from Run Paint Run Run. Fix #3131

Revision 34367 - 01/24/2012 04:02 AM - matz (Yukihiro Matsumoto)

* object.c (rb_Hash): add Kernel#Hash conversion method like Array() or Float(). a patch from Run Paint Run Run. Fix #3131

Revision 34367 - 01/24/2012 04:02 AM - matz (Yukihiro Matsumoto)

* object.c (rb_Hash): add Kernel#Hash conversion method like Array() or Float(). a patch from Run Paint Run Run. Fix #3131

Revision 34367 - 01/24/2012 04:02 AM - matz (Yukihiro Matsumoto)

* object.c (rb_Hash): add Kernel#Hash conversion method like Array() or Float(). a patch from Run Paint Run Run. Fix #3131
I forgot to show that case 2 supports empty arrays:

```ruby
Hash([])
=> {}
```

Just like Kernel#Array() supports empty hashes:

```ruby
Array({})
=> []
```

Thanks for your consideration. And sorry for the noise.

Hi,

In message "Re: [ruby-core:29462] [Feature #3131] add Kernel#Hash() method like Kernel#Array()" on Mon, 12 Apr 2010 08:32:52 +0900, Suraj Kurapati <redmine@ruby-lang.org> writes:

> There is an imbalance of power in the Ruby core API (when it comes to arrays and hashes) because it is easier to convert nil values into empty arrays, thanks to Kernel#Array(), than it is to convert nil values into empty hashes, due to lack of Kernel#Hash().

Having Hash() might be a good idea. But since conversion from arrays only meaningful for specific case (array of 2-element arrays), I am not sure whether Hash() should support conversion from Array in general or not.

matz.
Hi,

matz wrote:

since conversion from arrays only meaningful for specific case (array of 2-element arrays), I am not sure whether Hash() should support conversion from Array in general or not.

Hash::[] does not support conversion from odd-length Array (because it is uncertain what Ruby must do with the 2n+1'th element) and will raise an error in such case. So, in my view, Kernel#Hash() need not support conversion from odd-length Array either.

If you wish to avoid even-length Array conversion in Kernel#Hash() because odd-length Array conversion is not supported, then I request that Kernel#Hash() must still support conversion from empty Array because Kernel#Array() already supports conversion from empty Hash:

```ruby
Array([])
=> []
```

In summary, I request that Kernel#Hash() supports conversion from:

- nil
- Hash
- empty Array

It would be nice if Kernel#Hash() also supports conversion from:

- even-length Array

Thanks for your consideration.

#4 - 04/13/2010 09:19 AM - sunaku (Suraj Kurapati)

=begin
Hi,

Please allow me to clarify.

Suraj Kurapati wrote:

Hash::[] does not support conversion from odd-length Array (because it is uncertain what Ruby must do with the 2n+1'th element) and will raise an error in such case.

I was referring to this particular behavior:

```ruby
Hash[1,2,3,4,5,6]
=> {1=>2, 3=>4, 5=>6}

Hash[1,2,3,4,5,6,7]
ArgumentError: odd number of arguments for Hash
from (irb):3:in '['
from (irb):3
from /usr/bin/irb:12:in'
```

Avoided even-length Array conversion in Kernel#Hash() because odd-length Array conversion is not supported.

I forgot that Hash::[] can convert 2-element Array into Hash:

```ruby
Hash[1,2,3,4,5,6,7] # non 2-element Array ignored
=> {1=>2, 3=>4, 5, 6}=>[7]

Hash[1,2,3,4,5,6] # implicit pair value is nil
=> {5=>6, 7=>nil}

Hash[1,2,3,4,5,6,7,8] # previous pair is overwritten
=> {5=>6, 7=>nil}
```
Thanks for your consideration.

=begin

Hi Matz,

To avoid delaying this request forever due to unnecessary features, I have narrowed the requirements for Kernel#Hash() to the following:

Hash() must convert (1) nil, (2) Hash, and (3) empty Array into Hash.

module Kernel
def Hash(value)
  if value.respond_to? :to_hash
    value.to_hash
  elsif value.nil? or Array(value).empty?
    {}
  else
    raise ArgumentError, "invalid value for Hash: #{value}"
  end
end
end

Thanks for your consideration.

=end

#6 - 04/21/2010 10:23 PM - matz (Yukihiro Matsumoto)

=begin

Hi,

In message "Re: [ruby-core:29644] [Feature #3131] add Kernel#Hash() method like Kernel#Array()" on Tue, 20 Apr 2010 13:36:00 +0900, Suraj Kurapati redmine@ruby-lang.org writes:

|To avoid delaying this request forever due to unnecessary features, I have narrowed the requirements for Kernel#Hash() to the following: |
|Hash() must convert (1) nil, (2) Hash, and (3) empty Array into Hash. |

Sounds reasonable. But the new feature window for 1.9.2 was closed. So, I will consider (positively) adding this feature after 1.9.2 release.

matz.

=end

#7 - 04/22/2010 06:53 AM - shyouhei (Shyouhei Urabe)

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

=begin

=begin

#8 - 04/23/2010 09:54 AM - sunaku (Suraj Kurapati)

=begin

Hi,

Matz wrote:

I will consider (positively) adding this feature after 1.9.2 release.

Thanks Matz! In the mean time, I have created a gem with this functionality so that people can start using it, if they wish:

http://rubygems.org/gems/kernel_hash

=begin

09/14/2021

5/7
The source code (and unit tests) is available here:

http://github.com/sunaku/kernel_hash

Cheers.
=end

#9 - 05/25/2010 03:57 PM - sunaku (Suraj Kurapati)

=begin
Hi,

Please set the "Done %" for this issue to 80%.

The remaining 20% is for (possibly) rewriting my proposed Kernel#Hash() Ruby code in C.

Thanks for your consideration.
=\end

#10 - 09/02/2010 11:03 PM - runpaint (Run Paint Run Run)

File kernel-hash.patch added

=begin
I believe the attached patch reflects the consensus between matz and Suraj. The only material difference is that a TypeError is raised instead of an ArgumentError for consistency with Integer(), Float(), etc.
=\end

#11 - 04/23/2011 08:52 AM - sunaku (Suraj Kurapati)

=begin
Integer() and Float() in Ruby 1.9.2 raise TypeError and ArgumentError:

$ irb
## ruby 1.9.2p180 (2011-02-18 revision 30909) [x86_64-linux]
Float(nil)
TypeError: can't convert nil into Float
from (irb):6:in Float'
from (irb):6
from /usr/bin/irb:12:in'
Float('meow')
ArgumentError: invalid value for Float(): "meow"
from (irb):8:in Float'
from (irb):8
from /usr/bin/irb:12:in'

I don't know which error type we should raise in Kernel#Hash().
=\end

#12 - 04/23/2011 10:23 PM - matz (Yukihiro Matsumoto)

=begin
Hi,

In message "Re: [ruby-core:35857] [Ruby 1.9 - Feature #3131] add Kernel#Hash() method like Kernel#Array()" on Sat, 23 Apr 2011 08:52:12 +0900, Suraj Kurapati sunaku@gmail.com writes:

|Integer() and Float() in Ruby 1.9.2 raise TypeError and ArgumentError:

I think they should raise TypeError, if incompatibly here is not an issue, I will fix.

matz.
=\end

#13 - 09/26/2011 05:22 AM - sunaku (Suraj Kurapati)

Any chance of this getting into Ruby 1.9.3? Thanks.
On Mon, Sep 26, 2011 at 1:52 AM, Suraj Kurapati sunaku@gmail.com wrote:

Issue #3131 has been updated by Suraj Kurapati.

Any chance of this getting into Ruby 1.9.3? ÂThanks.

Doesn't look like it. See ruby-core 39693.

--
Anurag Priyam

#15 - 09/27/2011 08:17 AM - sunaku (Suraj Kurapati)
Thanks for the pointer. I'll just wait for Ruby 1.9.4 then.

#16 - 01/24/2012 12:36 PM - matz (Yukihiro Matsumoto)
I accept this patch for trunk, except that Hash(nil) should return an empty hash, not nil.

Matz.

#17 - 02/25/2012 02:46 PM - ko1 (Koichi Sasada)
Who should work on it?

#18 - 02/25/2012 03:13 PM - naruse (Yui NARUSE)
- Status changed from Assigned to Closed

This has already done, only missed to close.

Files

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>kernel-hash.patch</td>
<td>2.06 KB</td>
<td>09/02/2010</td>
</tr>
</tbody>
</table>