Ruby master - Feature #7359
#eql? and #equal? naming
11/15/2012 10:00 PM - aef (Alexander E. Fischer)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Rejected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>matz (Yukihiro Matsumoto)</td>
</tr>
<tr>
<td>Target version:</td>
<td>3.0</td>
</tr>
</tbody>
</table>

Description
In my opinion the difference between @#eql?! and @#equal?! is really unintuitive. How about making their difference more obvious by giving one of them a more accurate name?

My proposal is to rename @#equal?! to @#identic?!.

If you deprecate #equal? at the same time, maybe in the far future it can have a comeback as an alias for #eql? to make those people happy who dislike to use abbreviations just to reduce the character count by two and simultaneously making it harder to read in a classical sense.

If you like it, let me know. Then I will provide a patch.

History
#1 - 11/16/2012 07:31 AM - matz (Yukihiro Matsumoto)
- Status changed from Open to Rejected

Making them little more intuitive does not worth breaking millions of existing programs.

Matz.

#2 - 11/16/2012 09:29 AM - trans (Thomas Sawyer)
"Making them little more intuitive does not worth breaking millions of existing programs."

That's true, but why does it have to be one or the other? Just,

alias identical? equal?

And let that be for a year or two while getting word out to people they should start to use #identical? instead of #equal? for future. After a year or two of that, add a warning to #equal?. And let that be for another couple of years. Only after that, in a new major version, e.g. Ruby 2.2, 2.3 or Ruby 3.0 or whatever, would #equal? actually change. Plenty of time for an orderly managed transition.

It may seem minor, but little things add up. Why reject what is clearly an improvement just b/c it requires a managed transition?