

Ruby master - Feature #8863

New error class: UndefinedBehaviorError

09/05/2013 04:30 AM - alexeymuranov (Alexey Muranov)

Status:	Feedback
Priority:	Normal
Assignee:	
Target version:	
Description	
=begin I propose to consider introducing a new error class: (<code>UndefinedBehaviorError</code>) (or (<code>UnspecifiedBehaviorError</code>)). It would be somewhat similar to (<code>NotImplementedError</code>), but it would mean that the behavior is actually unspecified, but may be defined in future versions (of Ruby or of the application). The purpose is to raise it in "edge cases" of a newly introduced feature, to avoid users' relying on some unintended behavior, and to be able to change the behavior or to raise a different type of error in a future version. I was thinking about a possible need for something like this in relation to (<code>Enumerable#to_h</code>) #7292 . =end	

History

#1 - 09/05/2013 04:34 AM - alexeymuranov (Alexey Muranov)

Well, maybe just leaving the behavior unspecified is enough, i don't know. But raising an error would reduce a chance of breaking a program of someone who has not read the documentation well.

#2 - 09/05/2013 12:19 PM - matz (Yukihiro Matsumoto)

- Status changed from Open to Feedback

Hmm, defining the behavior to raise `UndefinedBehaviorError` seems contradiction to me.

Matz.

#3 - 09/05/2013 01:23 PM - alexeymuranov (Alexey Muranov)

matz (Yukihiro Matsumoto) wrote:

 Hmm, defining the behavior to raise `UndefinedBehaviorError` seems contradiction to me.

 Matz.

It sounds like it, but my thought was that a user may catch other types of errors and use them meaningfully, but here it would be clearly advised not to rely on catching it. I am not sure it is useful, i agree with Feedback status. I thought first that maybe some other class of Errors can be appropriate, like `NotImplementedError`, but it does not match the meaning. It is to leave options open when introducing a new feature.

#4 - 01/11/2014 11:09 AM - alexeymuranov (Alexey Muranov)

After thinking about it, and since there is no feedback, i think this my feature request may be closed. I think handling special cases or catching exceptions to raise another exception is an unnecessary complication. Documentation should suffice.