

Ruby master - Feature #15833

Some refactors for shared-root array

05/06/2019 10:29 AM - wanabe (_ wanabe)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	
Description	
<p>I wrote some patches for shared-root array. But I can't decide whether to commit them because they include the modification of public header include/ruby/ruby.h.</p> <p>To ruby core team (I'm not assuming anyone in particular): Can I modify include/ruby/ruby.h by the patches?</p> <p>The following is descriptions of the patches.</p> <p>0001-Name-RArray-member-to-count-reference-of-shared-root.patch names reference count of shared-root array. So far, as.heap.aux.capa is used as reference count but it is different from the reality, reference count. This is a follow up of r19824.</p> <p>0002-Export-RARRAY_SHARED_ROOT_FLAG.patch exports RARRAY_SHARED_ROOT_FLAG. I followed other flags as in RARRAY_EMBED_FLAG or RARRAY_TRANSIENT_FLAG.</p> <p>0003-Fix-object-info-of-shared-root-array.patch is not a refactor but feature request patch based on above ones. rb_obj_info outputs shared-root array info as same as normal array currently, but its capa is not a really "capa" as I said.</p>	

History

#1 - 05/13/2019 10:42 PM - Eregon (Benoit Daloze)

I think nowadays RArray should be considered internal and C extensions should not access members of struct RArray directly, as that's very likely going to be incorrect or very complex.

I don't think many C extensions use RArray directly, but it would be good to check.

#2 - 05/14/2019 12:08 AM - wanabe (_ wanabe)

Eregon (Benoit Daloze) wrote:

I think nowadays RArray should be considered internal and C extensions should not access members of struct RArray directly, as that's very likely going to be incorrect or very complex.

I don't think many C extensions use RArray directly, but it would be good to check.

Thank you to comment.

I didn't think about C extension but the readability of MRI itself.

struct RArray definition can't move from include/ruby/ruby.h, so it can't be avoided to modify include/ruby/ruby.h if I want to rename struct RArray member for readability.

But I change my mind that it may not be good to export RARRAY_SHARED_ROOT_FLAG.

I will drop 0002-Export-RARRAY_SHARED_ROOT_FLAG.patch.

And I will also drop 0003-Fix-object-info-of-shared-root-array.patch because it is based on above 0002.

#3 - 05/15/2019 09:05 PM - Eregon (Benoit Daloze)

I think RARRAY_SHARED_ROOT_FLAG could be in internal.h, if you want to expose it outside of array.c but not as public C API.

Longer-term I think we should move struct RArray to internal.h, but that's a much bigger/riskier change so not for this issue obviously :)

#4 - 07/29/2019 07:13 AM - nobu (Nobuyoshi Nakada)

Eregon (Benoit Daloze) wrote:

I think RARRAY_SHARED_ROOT_FLAG could be in internal.h, if you want to expose it outside of array.c but not as public C API.

Longer-term I think we should move struct RArray to internal.h, but that's a much bigger/riskier change so not for this issue obviously :)

I agree both.

Files

0001-Name-RArray-member-to-count-reference-of-shared-root.patch	1.42 KB	05/06/2019	wanabe (_ wanabe)
0002-Export-RARRAY_SHARED_ROOT_FLAG.patch	1.54 KB	05/06/2019	wanabe (_ wanabe)
0003-Fix-object-info-of-shared-root-array.patch	1.04 KB	05/06/2019	wanabe (_ wanabe)