

## Ruby - Feature #13252

### C API for creating strings without copying

02/25/2017 12:19 AM - tenderlovmaking (Aaron Patterson)

<b>Status:</b>	Assigned	
<b>Priority:</b>	Normal	
<b>Assignee:</b>	ko1 (Koichi Sasada)	
<b>Target version:</b>		
<b>Description</b>		
Hi,		
I'd like to have a C API that allows me to create String objects without copying the underlying char *. Basically a C API similar to the rb_str_new_static, but have the GC free the underlying char * when the object dies. The use case is that at work we have C libraries that allocate char * and we want to pass those to Ruby land without copying the buffer. Here is an example of what we're doing now:		
<a href="https://github.com/arthurmn/memcached/commit/1886546944b420dc6953096ba1f5eae772001e31#diff-f508f9b8263ea397534b2b3f8efed987R147">https://github.com/arthurmn/memcached/commit/1886546944b420dc6953096ba1f5eae772001e31#diff-f508f9b8263ea397534b2b3f8efed987R147</a>		
I'd like it if there was a public API for doing something like this.		
Thank you!		
P.S. I am sure I can't be the first to ask for this, but I couldn't find a similar issue in RedMine, so if this has been answered I apologize.		
P.P.S. I've added a patch for kind of what I want.		

#### History

##### #1 - 02/25/2017 01:01 AM - nobu (Nobuyoshi Nakada)

- Description updated

It is not guaranteed that ruby\_xfree can free a pointer allocated by other than ruby\_xmalloc and so on.  
You can't mix bare malloc and ruby\_xfree.

##### #2 - 03/01/2017 04:12 AM - shyouhei (Shyouhei Urabe)

Nobuyoshi Nakada wrote:

It is not guaranteed that ruby\_xfree can free a pointer allocated by other than ruby\_xmalloc and so on.  
You can't mix bare malloc and ruby\_xfree.

Agreed. I heard that DLLs can have their own malloc() implementation on Windows. Even on POSIX variants, memory regions (e.g. mmap()-ed ones) are not always guaranteed to be free()-able.

This proposed C APIs are too easy to be misused when publicized. And misuse of them directly results in SEGV. I don't think it's a wise idea.

##### #3 - 03/01/2017 08:11 AM - normalperson (Eric Wong)

[shyouhei@ruby-lang.org](mailto:shyouhei@ruby-lang.org) wrote:

Nobuyoshi Nakada wrote:

It is not guaranteed that ruby\_xfree can free a pointer allocated by other than ruby\_xmalloc and so on.  
You can't mix bare malloc and ruby\_xfree.

Agreed. I heard that DLLs can have their own malloc() implementation on Windows. Even on POSIX variants, memory regions (e.g. mmap()-ed ones) are not always guaranteed to be free()-able.

This proposed C APIs are too easy to be misused when publicized. And misuse of them directly results in SEGV. I don't think it's a wise idea.

One place we can use this in C Ruby is ruby\_getcwd() on GNU libc.

Yes, it's dangerous if misused. Perhaps having a way to plug-in

and redefine alloc/free/realloc functions would be safe on a per-T\_STRING basis.

We can maybe use FL\_USER{3,4,5}, and STR\_NOFREE flags for non-embed strings and allow up to 14 non-standard plug-in \*alloc implementations for users to define.

(4 bits; 16 implementations possible, 2 of which are built-in:  
1: normal malloc 2: no-free, replacing existing STR\_NOFREE cases)

**#4 - 04/17/2017 07:03 AM - nobu (Nobuyoshi Nakada)**

normalperson (Eric Wong) wrote:

We can maybe use FL\_USER{3,4,5}, and STR\_NOFREE flags for non-embed strings and allow up to 14 non-standard plug-in \*alloc implementations for users to define.

FL\_USER{3,4,5} are for RSTRING\_EMBED\_LEN\_MASK.

**#5 - 04/17/2017 07:22 AM - shyouhei (Shyouhei Urabe)**

- Status changed from Open to Assigned

**#6 - 04/17/2017 07:22 AM - shyouhei (Shyouhei Urabe)**

- Assignee set to ko1 (Koichi Sasada)

**Files**

out.diff	1.66 KB	02/25/2017	tenderlovemaking (Aaron Patterson)
----------	---------	------------	------------------------------------