

Ruby master - Bug #10466

rb_eval_string_wrap does not actually wrap in a module binding

11/02/2014 03:39 PM - silverhammermba (Max Anselm)

Status: Assigned	
Priority: Normal	
Assignee: ko1 (Koichi Sasada)	
Target version:	
ruby -v: ruby 2.1.4p265 (2014-10-27 revision 48166) [x86_64-linux]	Backport: 2.0.0: UNKNOWN, 2.1: UNKNOWN

Description

rb_eval_string_wrap says that it "evaluates the given string under a module binding in an isolated binding", but this isn't true.

Run the following:

```
#include <ruby.h>

int main(int argc, char* argv[])
{
    ruby_init();

    int state;
    rb_eval_string_protect("X = 2", &state);
    rb_eval_string_wrap("X = 3", &state);
    rb_eval_string_protect("puts X", &state);

    return ruby_cleanup(0);
}
```

Expected:

outputs 2

Actual:

outputs

```
eval:1: warning: already initialized constant X
eval:1: warning: previous definition of X was here
3
```

It looks like rb_eval_string_wrap tries to wrap it

```
th->top_wrapper = rb_module_new();
th->top_self = rb_obj_clone(rb_vm_top_self());
rb_extend_object(th->top_self, th->top_wrapper);
```

But it ends up calling ruby_eval_string_from_file which uses rb_vm_top_self() as self, thus undoing the wrapping.

rb_load can perform similar wrapping, but there it works properly.

History

#1 - 12/08/2014 02:00 AM - retro (Josef Šimánek)

You can reproduce this without C code also.

```
Module.new {X = 5} #=> #<Module:0x000000012186c0>
Module.new {X = 5} #=> #<Module:0x00000001209da0>
(irb):2: warning: already initialized constant X
(irb):1: warning: previous definition of X was here
```

And since rb_eval_string_wrap is really wrapping code into module, this works without warning:

```
#include <ruby.h>

int main(int argc, char* argv[])
{
    ruby_init();

    int state;
    rb_eval_string_wrap("self::Y = 'wrap'", &state);
    rb_eval_string_wrap("self::Y = 'wrap'", &state);

    return ruby_cleanup(0);
}
```

#2 - 07/23/2019 03:12 AM - jeremyevans0 (Jeremy Evans)

- File *rb_eval_string_wrap_cref.patch* added

I agree this a bug. The documentation states for `rb_eval_string_wrap` that This is same as the binding for loaded libraries on `"load('foo', true)"`, but that doesn't appear to be the case. `load('foo', true)` evaluates with an anonymous module in `Module.nesting`, but `rb_eval_string_wrap` uses an empty `Module.nesting`.

The attached patch should fix the issue by setting a `cref`, so that `rb_eval_string_wrap` evaluates the string with an anonymous module in `Module.nesting`. I'm not sure if this is the best way to fix it, and would appreciate if another committer could review this patch. Unfortunately, there are no tests for the `rb_eval_string_wrap` function and nothing internally seems to use this method.

#3 - 07/30/2019 07:54 AM - ko1 (Koichi Sasada)

- Assignee set to *ko1* (Koichi Sasada)

- Status changed from *Open* to *Assigned*

Files

rb_eval_string_wrap_cref.patch	1.79 KB	07/23/2019	jeremyevans0 (Jeremy Evans)
--	---------	------------	-----------------------------