

# Ruby - Bug #5283

clone

09/06/2011 10:05 PM - shiba (satoshi shiba)

<b>Status:</b> Closed	
<b>Priority:</b> Normal	
<b>Assignee:</b> nobu (Nobuyoshi Nakada)	
<b>Target version:</b>	
<b>ruby -v:</b> -	<b>Backport:</b>

## Description

```
=begin
clone
clone clone
```

- String.class\_eval do  
def self.singleton\_method\_added(mid)  
puts("singleton\_method\_added: self = #{self}, mid = #{mid}")  
end  
end

```
puts("start clone")
StrClone = String.clone
puts("finish clone")
```

- singleton\_method\_added: self = String, mid = singleton\_method\_added  
start clone  
singleton\_method\_added: self = String, mid = try\_convert  
singleton\_method\_added: self = String, mid = singleton\_method\_added  
finish clone

```
String clone  
StrClone String  
singleton_method_added  
clone clone  
singleton_method_added
```

```
StrClone singleton_method_added  
singleton_method_added  
singleton_method_added
```

```
String singleton_method_added  
rb_singleton_class_clone(class.c)  
rb_singleton_class_clone  
"attached"  

```

```
VALUE  
rb_singleton_class_clone(VALUE obj)  
{  
...  
if (RCLASS_IV_TBL(klass)) {  
/* /  
RCLASS_IV_TBL(clone) = st_copy(RCLASS_IV_TBL(klass));  
}  
...  
RCLASS_M_TBL(clone) = st_init_numtable();  
data.tbl = RCLASS_M_TBL(clone);  
data.klass = (VALUE)clone;  
/* method_added */  
st_foreach(RCLASS_M_TBL(klass), clone_method,
```

```
(st_data_t)&data);
...
}

/* "attached" hook String
   singleton_method_added hook CALL_METHOD_HOOK
   "attached" hook
   singleton_method_added hook String hook */

#define CALL_METHOD_HOOK(klass, hook, mid) do {
const VALUE arg = ID2SYM(mid);
VALUE recv_class = (klass);
ID hook_id = (hook);
if (FL_TEST((klass), FL_SINGLETON)) {
recv_class = rb_ivar_get((klass), attached);
hook_id = singleton_##hook;
}
rb_funcall2(recv_class, hook_id, 1, &arg);
} while (0)

/* singleton_method_added hook
   rb_obj_clone hook
   RBASIC(clone)->klass = rb_singleton_class_clone_and_attach(obj, clone);
   hook
   hook */

/* hook */

```

- hook

## Index: object.c

```
--- object.c (revision 33202)
+++ object.c (working copy)
@@ -268,6 +268,7 @@
```

- the class.  
\*/

```
+VALUE rb_singleton_class_clone_and_attach(VALUE obj, VALUE attach);
VALUE
rb_obj_clone(VALUE obj)
{
@@ -277,9 +278,9 @@
rb_raise(rb_eTypeError, "can't clone %s", rb_obj_classname(obj));
}
clone = rb_obj_alloc(rb_obj_class(obj));
```

- RBASIC(clone)->klass = rb\_singleton\_class\_clone(obj);  
RBASIC(clone)->flags = (RBASIC(obj)->flags | FL\_TEST(clone, FL\_TAINT) | FL\_TEST(clone, FL\_UNTRUSTED)) &  
~(FL\_FREEZE|FL\_FINALIZE|FL\_MARK);  
init\_copy(clone, obj);
- RBASIC(clone)->klass = rb\_singleton\_class\_clone\_and\_attach(obj, clone);  
rb\_funcall(clone, id\_init\_clone, 1, obj);  
RBASIC(clone)->flags |= RBASIC(obj)->flags & FL\_FREEZE;

## Index: class.c

```
--- class.c (revision 33202)
+++ class.c (working copy)
@@ -219,9 +219,17 @@
```



**#3 - 12/29/2012 11:37 AM - nobu (Nobuyoshi Nakada)**

- Status changed from *Assigned* to *Closed*

- % Done changed from 0 to 100

This issue was solved with changeset r38648.  
satoshi, thank you for reporting this issue.  
Your contribution to Ruby is greatly appreciated.  
May Ruby be with you.

---

object.c: singleton class clone

- object.c (rb\_obj\_clone): attach clone to its singleton class during cloning singleton class so that singleton\_method\_added will be called on it. based on the patch by shiba (satoshi shiba)[Bug #5283] in [\[ruby-dev:44477\]](#). [Bug #5283]