

Ruby - Bug #3445

1.9.2-preview3 Object#extend public_methods(false)

06/16/2010 02:00 AM - mame (Yusuke Endoh)

Status:	Closed	Backport:
Priority:	Normal	
Assignee:	mame (Yusuke Endoh)	
Target version:	1.9.2	
ruby -v:	1.9.2-preview3	
Description		
<pre>=begin [ruby-dev:41553] redmine ML </pre>		
<pre> 1.9.2-preview3 trunk Object#extend public_methods(false) 1.9.1 1.9.2-preview1 + extend public NEWS module M def m1; end def m2; end module_function :m2 end class A def a; end end obj = A.new p obj.public_methods(false) # => [:a] obj.extend(M) p obj.public_methods(false) # => trunk, 1.9.2-preview3 => [] # => 1.9.1, 1.9.2-preview1 => [:m1, :a] -- Chikanaga Tomoyuki Nippon Control System Corp. =end</pre>		

History

#1 - 06/16/2010 02:08 AM - mame (Yusuke Endoh)

- Assignee set to mame (Yusuke Endoh)

```
=begin

```

```

false
#2993

```

```
Module#public_instance_methods(false):
Object#public_methods(false):

```

```


```

```


```

```

diff --git a/class.c b/class.c
index 9a3984a..b6f0999 100644
--- a/class.c
+++ b/class.c
@@ -833,7 +833,7 @@ method_entry(ID key, const rb_method_entry_t *me, st_table *list)
}

static VALUE
-class_instance_method_list(int argc, VALUE *argv, VALUE mod, int (*func) (ID, long, VALUE))
+class_instance_method_list(int argc, VALUE *argv, VALUE mod, int obj, int (*func) (ID, long, VALUE))
{
    VALUE ary;
    int recur;
    @@ -852,6 +852,7 @@ class_instance_method_list(int argc, VALUE *argv, VALUE mod, int (*func) (ID, lo
    for (; mod; mod = RCLASS_SUPER(mod)) {
        st_foreach(RCLASS_M_TBL(mod), method_entry, (st_data_t)list);
        if (BUILTIN_TYPE(mod) == T_ICLASS) continue;

        • if (obj && FL_TEST(mod, FL_SINGLETON)) continue;
          if (!recur) break;
        }
        ary = rb_ary_new();
        @@ -891,7 +892,7 @@ class_instance_method_list(int argc, VALUE *argv, VALUE mod, int (*func) (ID, lo
        VALUE
        rb_class_instance_methods(int argc, VALUE *argv, VALUE mod)
        {

        • return class_instance_method_list(argc, argv, mod, ins_methods_i);

        • return class_instance_method_list(argc, argv, mod, 0, ins_methods_i);
        }

/*
@@ -906,7 +907,7 @@ rb_class_instance_methods(int argc, VALUE *argv, VALUE mod)
VALUE
rb_class_protected_instance_methods(int argc, VALUE *argv, VALUE mod)
{

    • return class_instance_method_list(argc, argv, mod, ins_methods_prot_i);

    • return class_instance_method_list(argc, argv, mod, 0, ins_methods_prot_i);
    }

/*
@@ -929,7 +930,7 @@ rb_class_protected_instance_methods(int argc, VALUE *argv, VALUE mod)
VALUE
rb_class_private_instance_methods(int argc, VALUE *argv, VALUE mod)
{

    • return class_instance_method_list(argc, argv, mod, ins_methods_priv_i);

    • return class_instance_method_list(argc, argv, mod, 0, ins_methods_priv_i);
    }

/*
@@ -944,7 +945,93 @@ rb_class_private_instance_methods(int argc, VALUE *argv, VALUE mod)
VALUE
rb_class_public_instance_methods(int argc, VALUE *argv, VALUE mod)
{

    • return class_instance_method_list(argc, argv, mod, ins_methods_pub_i);

    • return class_instance_method_list(argc, argv, mod, 0, ins_methods_pub_i);
    +}

+/*

    ◦ call-seq:

    ◦ obj.methods -> array

    ◦ Returns a list of the names of methods publicly accessible in

```

- *obj*. This will include all the methods accessible in
- *obj*'s ancestors.

```

◦ class Klass

◦   def kMethod()

◦   end

◦ end

◦ k = Klass.new

◦ k.methods[0..9]    #=> [:kMethod, :freeze, :nil?, :is_a?,

◦                   #   :class, :instance_variable_set,

◦                   #   :methods, :extend, :__send__, :instance_eval]

◦ k.methods.length   #=> 42

• */

```

+VALUE

+rb_obj_methods(int argc, VALUE *argv, VALUE obj)

+{

- retry:
- if (argc == 0) {
- VALUE args[1];
- args[0] = Qtrue;
- return class_instance_method_list(argc, argv, CLASS_OF(obj), 1, ins_methods_i);
- }
- else {
- VALUE recur;
- rb_scan_args(argc, argv, "1", &recur);
- if (RTEST(recur)) {

- XXXXXXXXXX

- XXXXXXXXXX

- }
- return rb_obj_singleton_methods(argc, argv, obj);
- }
- +}

+/*

- call-seq:
- `obj.protected_methods(all=true)` -> array
- Returns the list of protected methods accessible to *obj*. If
- the *all* parameter is set to false, only those methods
- in the receiver will be listed.
- */

+VALUE

```

+rb_obj_protected_methods(int argc, VALUE *argv, VALUE obj)
+{
    • return class_instance_method_list(argc, argv, CLASS_OF(obj), 1, ins_methods_prot_i);
    +}

+/*
    ◦ call-seq:

    ◦ obj.private_methods(all=true) -> array

    ◦ Returns the list of private methods accessible to obj. If

    ◦ the all parameter is set to false, only those methods

    ◦ in the receiver will be listed.
    • */

+VALUE
+rb_obj_private_methods(int argc, VALUE *argv, VALUE obj)
+{
    • return class_instance_method_list(argc, argv, CLASS_OF(obj), 1, ins_methods_priv_i);
    +}

+/*
    ◦ call-seq:

    ◦ obj.public_methods(all=true) -> array

    ◦ Returns the list of public methods accessible to obj. If

    ◦ the all parameter is set to false, only those methods

    ◦ in the receiver will be listed.
    • */

+VALUE
+rb_obj_public_methods(int argc, VALUE *argv, VALUE obj)
+{
    • return class_instance_method_list(argc, argv, CLASS_OF(obj), 1, ins_methods_pub_i);
    }

/*
diff --git a/object.c b/object.c
index 414c3a9..f52522d 100644
--- a/object.c
+++ b/object.c
@@ -1751,109 +1751,10 @@ rb_mod_const_defined(int argc, VALUE *argv, VALUE mod)
return RTEST(recur) ? rb_const_defined(mod, id) : rb_const_defined_at(mod, id);
}

-/*
    ◦ call-seq:

    ◦ obj.methods -> array

    ◦ Returns a list of the names of methods publicly accessible in

    ◦ obj. This will include all the methods accessible in

```

- *obj*'s ancestors.

```

◦ class Klass
◦   def kMethod()
◦   end
◦ end

◦ k = Klass.new

◦ k.methods[0..9]    #=> [:kMethod, :freeze, :nil?, :is_a?,
◦                    #   :class, :instance_variable_set,
◦                    #   :methods, :extend, :__send__, :instance_eval]

◦ k.methods.length   #=> 42

```

• */

-static VALUE

-rb_obj_methods(int argc, VALUE *argv, VALUE obj)

-{

- retry:
- if (argc == 0) {
- VALUE args[1];
- args[0] = Qtrue;
- return rb_class_instance_methods(1, args, CLASS_OF(obj));
- }
- else {
- VALUE recur;
- rb_scan_args(argc, argv, "1", &recur);
- if (RTEST(recur)) {

• [REDACTED]

• [REDACTED]

- }
- return rb_obj_singleton_methods(argc, argv, obj);
- }
- }

/*

- call-seq:

```
◦ obj.protected_methods(all=true) -> array
```

- Returns the list of protected methods accessible to *obj*. If
- the *all* parameter is set to false, only those methods
- in the receiver will be listed.

• */

-static VALUE

-rb_obj_protected_methods(int argc, VALUE *argv, VALUE obj)

-{

- if (argc == 0) { /* hack to stop warning */
- VALUE args[1];
- args[0] = Qtrue;
- return rb_class_protected_instance_methods(1, args, CLASS_OF(obj));
- }
- return rb_class_protected_instance_methods(argc, argv, CLASS_OF(obj));
- }

/*

◦ call-seq:

◦ `obj.private_methods(all=true)` → array

◦ Returns the list of private methods accessible to *obj*. If

◦ the *all* parameter is set to false, only those methods

◦ in the receiver will be listed.

• */

-static VALUE

-rb_obj_private_methods(int argc, VALUE *argv, VALUE obj)

{

- if (argc == 0) { /* hack to stop warning */
- VALUE args[1];
- args[0] = Qtrue;
- return rb_class_private_instance_methods(1, args, CLASS_OF(obj));
- }
- return rb_class_private_instance_methods(argc, argv, CLASS_OF(obj));
- }

/*

◦ call-seq:

◦ `obj.public_methods(all=true)` → array

◦ Returns the list of public methods accessible to *obj*. If

◦ the *all* parameter is set to false, only those methods

◦ in the receiver will be listed.

• */

-static VALUE

-rb_obj_public_methods(int argc, VALUE *argv, VALUE obj)

{

- if (argc == 0) { /* hack to stop warning */
- VALUE args[1];
- args[0] = Qtrue;
- return rb_class_public_instance_methods(1, args, CLASS_OF(obj));
- }
- return rb_class_public_instance_methods(argc, argv, CLASS_OF(obj));
- }
- +VALUE rb_obj_methods(int argc, VALUE *argv, VALUE obj);
- +VALUE rb_obj_protected_methods(int argc, VALUE *argv, VALUE obj);
- +VALUE rb_obj_private_methods(int argc, VALUE *argv, VALUE obj);
- +VALUE rb_obj_public_methods(int argc, VALUE *argv, VALUE obj);

/*

- call-seq:

--

Yusuke Endoh mame@tsg.ne.jp

=end

#2 - 06/23/2010 10:29 PM - mame (Yusuke Endoh)

- *Status changed from Open to Closed*

=begin

~~~~~

r28357 ~~~~~ close ~~~~~

--

Yusuke Endoh [mame@tsg.ne.jp](mailto:mame@tsg.ne.jp)

=end