

Ruby master - Bug #11274

Equality inconsistency between Method and UnboundMethod

06/17/2015 08:01 AM - ko1 (Koichi Sasada)

Status: Open	
Priority: Normal	
Assignee: matz (Yukihiro Matsumoto)	
Target version:	
ruby -v: 2.3dev	Backport: 2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: UNKNOWN

Description

Method `#equality` inconsistency

```
module M1
  def foo; end
end

module M2
  include M1
  alias bar foo
end

class C
  include M1
  include M2
end

c = C.new
c_m1 = c.method(:foo)
c_m2 = c.method(:bar)
p [c_m1, c_m2, c_m1 == c_m2, c_m2 == c_m1]
```

```
[[
[#<Method: C(M1)#foo>, #<Method: C(M1)#bar(foo)>, true, true]
```

]

UnboundMethod#== inconsistency

```
module M1
  def foo; end
end

module M2
  include M1
  alias bar foo
end

ubm1 = M1.instance_method(:foo)
ubm2 = M2.instance_method(:bar)

p [ubm1, ubm2, ubm1==ubm2, ubm2==ubm1]
```

```
[[
[#<UnboundMethod: M1#foo>, #<UnboundMethod: M2(M1)#bar(foo)>, false, false]
```

]

]

Method#==

* Two method objects are equal if they are bound to the same object and refer to the same method definition and their owners are the same class or module.

```
module M1
  def foo; end
end
```

```
module M2
  include M1
  alias bar foo
end
```

```
class C
  include M1
  include M2
end
```

```
c = C.new
c_m1 = c.method(:foo)
c_m2 = c.method(:bar)
```

```
p c_m1.owner, c_m2.owner
```

M1
M2

Method#==

owner

alias

History

#1 - 06/17/2015 03:12 PM - matz (Yukihiro Matsumoto)

alias owner alias

Matz.