Ruby master - Bug #1983

Struct doesn't define instance methods for valid method names

08/24/2009 11:18 AM - bitsweat (Jeremy Daer)

Status:	Rejected		
Priority:	Normal		
Assignee:	matz (Yukihiro Matsumoto)		
Target version:	1.9.2		
ruby -v:	ruby 1.9.2dev (2009-08-16) [i386-darwin9.8.0]	**Backport:**	

Description

=begin

```
    foo = Struct.new(:foo?, :foo!).new(true, true)
    => #
    foo.foo?
    NoMethodError: undefined method foo?' for #<struct :foo?=true, :foo!=true>
    from (irb):3
    from /usr/local/bin/irb19:12:in'
    foo.foo!
    NoMethodError: undefined method foo!' for #<struct :foo?=true, :foo!=true>
    from (irb):4
    from /usr/local/bin/irb19:12:in'
    foo[:foo?]
    => true
    foo[:foo!]
    => true
```

I think this is because foo?= and foo!= are not valid setter methods.

By why not generate the reader and skip the setter, then?
=end

History

#1 - 08/24/2009 12:30 PM - tmat (Tomas Matousek)

=begin
Since

```
irb(main):012:0> class C
irb(main):013:1>   attr_accessor :foo?
irb(main):014:1> end
NameError: invalid attribute name `foo?'
```

I would rather think that the same error should be raised by Struct.new.

BTW, even more interesting names for attributes are operator names:

```
irb(main):007:0> Struct.new(:===, :+, :-)[1,2,3]
=> # :====1, :+=2, :-=3>
```

=end

#2 - 08/24/2009 01:24 PM - bitsweat (Jeremy Daer)

=begin
Good point; struct is consistent with attr_*. Both use the same low-level check: (rb_is_local_id(id) || rb_is_const_id(id))

I'd like attr_reader :foo? to work too. It's a common convention for booleans. And for Struct, this is a nice way to stub out a duck-typed object.
=end

#3 - 12/21/2009 11:56 AM - marcandre (Marc-Andre Lafortune)

- Assignee set to matz (Yukihiro Matsumoto)

=begin

=end

#4 - 03/02/2010 10:42 PM - mame (Yusuke Endoh)

- Status changed from Open to Rejected

=begin
Hi Jeremy,

> By why not generate the reader and skip the setter, then?

In [ruby-core:4404], matz decided to prohibit such a field name once.
And in [ruby-core:4577], he relaxed the restriction:

> having field names with "?" can be valid if field
> accessor is disabled for such cases.

So it is currently intended, not a bug. I close this ticket.

You can re-register this as Feature ticket.
If you do, please elaborate the motivation example.

> I'd like attr_reader :foo? to work too. It's a common convention for booleans.

We cannot define @foo? even by using reflection:

```
instance_variable_set(:"@foo?", 42)
#=> `@foo?' is not allowed as an instance variable name (NameError)
```

It is never a common convention.

IMO, if such a field name is allowed, we would next want to use foo?=
and @foo?. I think Ruby design is sane because it limits ourselves
from such earthly desires.

--
Yusuke Endoh mame@tsg.ne.jp
=end

#5 - 03/03/2010 02:20 PM - bitsweat (Jeremy Daer)

=begin
Thanks Yusuke. It's a reasonable restriction, considering my only motivation is to use Struct to make stubs for testing other code.
=end