Ruby trunk - Feature #3908

private constant

10/06/2010 12:19 AM - mame (Yusuke Endoh)

Status:	Closed	
Priority:	Normal	
Assignee:	mame (Yusuke Endoh)	
Target version:	1.9.3	

Description

=begin
Hi,

I'd propose "private constant."
Private constant provides method-like visibility for constant.

=== Sample code ===

module SomeModule
class PublicClass
...
end

```
class PrivateClass
  ...
end

# you can make "PrivateClass" private by private_constant method
private_constant :PrivateClass
```

end

we can refer access constant as is conventionally done
p SomeModule::PublicClass #=> SomeModule::PublicClass

a RuntimeError is raised when we attempt to refer private constant
p SomeModule::PrivateClass #=> private constant (RuntimeError)

we can even refer private constant from its parent scope
module SomeModule
p PrivateClass #=> SomeModule::PrivateClass
end

=== Background ===

Traditionally, there is no way to prevent users from using your classes.
It is too easy for user to access any internal class (e.g., CGI::Html3,
Enumerator::Generator, Matrix::Scalar, URI::Util, etc).
We can only write a document to ask users not to use them.

RubySpec inspired me to propose this feature. RubySpec declares the
policy that no spec should be written for private method. Nevertheless,
there were some specs for internal classes, such as CGI::Html3 (FYI,
such specs are already deleted).
I thought this was because there is no way to explicitly declare that
the constants are "internal use only."

=== Proposal ===

Private constant is a scoped constant that can be referred only from
its parent scope. It can be used for declaring "the constant is
for internal use," like private method.

When users try to refer private constant, they can realize that they

are going to use non-guaranteed feature, because the constant cannot
be referred so easily. Even so, they can use such a feature with
self-responsibility, by explicitly opening its parent scope.

Since the default visibility is public, this feature does not break
any compatibility.

=== Current status ===

I first suggested this at [ruby-dev:39685].
Matz approved my proposal [ruby-dev:39686]
Yugui has also approved [ruby-dev:40254], but said that it is needed
to discuss in ruby-core list before commit.

The patches are attached. make check and make test-rubyspec are all
passed.

What do you think?
I'll commit the patch unless there is objection.

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

History

#1 - 10/06/2010 12:41 AM - now (Nikolai Weibull)

=begin
On Tue, Oct 5, 2010 at 17:19, Yusuke Endoh redmine@ruby-lang.org wrote:

> What do you think?

Yes! Finally! Thanks!

=end

#2 - 10/27/2010 02:37 AM - mame (Yusuke Endoh)

- Status changed from Open to Closed

- % Done changed from 0 to 100

=begin
This issue was solved with changeset r29603.
Yusuke, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

=end

#3 - 10/27/2010 06:30 AM - runpaint (Run Paint Run Run)

=begin
I'm sorry I missed the deadline on this ticket...

1) Is it intentional that const_get :K, where :K is a private constant, raises a NameError? In the context of private methods, the reflection API only enforces visibility constraints if they have been explicitly requested (e.g. #methods c.f. #private_methods).

2) Should private constants be returned by #constants? They currently are, but #methods excludes private methods. That is, all methods returned by #methods may be objectified with #method, yet the same is not true for constants. Further, as there is no predicate for determining whether a constant is public or private, the programmer must resort to rescuing NameError when using #const_get. This represents an incompatibility in that existing code of the form K.constants.map{|c| K.const_get c}, where K contains one or more private constants, will now raise an exception.

3) Is it intended that, given the name of a private constant, #set_const makes it public?

```
 class K
   C = true
   private_constant :C
 end
K::C
```

```
#=> NameError: private constant K::C referenced
     from (irb):6
     from /usr/local/bin/irb:12:in `<main>'
K.const_set :C, false
(irb):7: warning: already initialized constant C
#=> false
K::C #=> false
```

=end

#4 - 10/27/2010 10:30 PM - nagachika (Tomoyuki Chikanaga)

=begin
I found a typo in error message in r29603.

Index: variable.c
===
--- variable.c (revision 29612)
+++ variable.c (working copy)
@@ -1895,7 +1895,7 @@

```
  if (rb_safe_level() >= 4 && !OBJ_UNTRUSTED(mod)) {
rb_raise(rb_eSecurityError,
```

- "Insecure: can't change method visibility");

- [redacted]

  }

  for (i = 0; i < argc; i++) {

=end

#5 - 10/28/2010 07:34 AM - mame (Yusuke Endoh)

=begin
Hi,

2010/10/27 Run Paint Run Run redmine@ruby-lang.org:

> Issue #3908 has been updated by Run Paint Run Run.
>
> I'm sorry I missed the deadline on this ticket...

No problem. Thank you always for your detailed spec review.
Say, I wanna work on File.write...

> 1) Is it intentional that const_get :K, where :K is a private constant, raises a NameError?

No. I think #const_get' does not raise an exception for
private constants because#method' and `#send' does not
for private methods.

Matz, may I introduce a new method #public_const_get', like
#public_send'?

> 2) Should private constants be returned by #constants?

I think no. I'm attaching a patch.

Matz, may I introduce two new methods, #public_constants'
and#private_constants'?

> 3) Is it intended that, given the name of a private constant, #set_const makes it public?

No. I'm attaching a patch.

NOTES:

- Matz is now casting doubt on this feature ;-(because
  we've not estimated the impact of this feature yet.
  Though I thought that I certainly got matz's approval,
  I might make a quick judgment.
  We're discussing still now, but this feature may be
  reverted.

- The first patch slightly changes YARV insn format for
  defineclass, to distinguish constant access style in
  class definition, e.g.,
  OK: class Foo::PrivateClass
  NG: class PrivateClass
  Thus, ko1's approval is needed.

- I have to work on defined?(Foo::PrivateConstant)

--
Yusuke Endoh mame@tsg.ne.jp

Attachment: 0001-add-rb_const_get_-_visi-functions.patch
Attachment: 0002-const_set-should-not-change-constant-visibility.patch
Attachment: 0003-Module-constant-should-exclude-private-constants.patch
=end

#6 - 10/28/2010 07:37 AM - mame (Yusuke Endoh)

- Status changed from Closed to Assigned

=begin
Hi,

2010/10/27 Tomoyuki Chikanaga redmine@ruby-lang.org:

> I found a typo in error message in r29603.

Thank you always for your detailed code review.
I'll import your patch after matz's decision.

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

#7 - 10/28/2010 07:53 AM - runpaint (Run Paint Run Run)

=begin

> Matz is now casting doubt on this feature ;-(because we've not estimated the impact of this feature yet.
> Though I thought that I certainly got matz's approval, I might make a quick judgment. We're discussing
> still now, but this feature may be reverted.

It would be a shame if it's reverted. There are a few rough edges, but nothing insurmountable. FWIW, I agree with your changes.
=end

#8 - 12/22/2010 09:07 AM - Anonymous

- Status changed from Assigned to Closed

=begin
This issue was solved with changeset r30290.
Yusuke, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

=end

#9 - 12/22/2010 11:22 AM - nahi (Hiroshi Nakamura)

=begin
Hi,

Yusuke, patches in the mail I'm replying are not applied and we still need discuss about followings, right?

- Module#constants includes a private constant?
- Module#const_get raises NameError for a private constant?
- add Module#public_constants and Module#private_constants?
- Module#const_set changes the constant visibility to public? Setting visibility via const_set?

Just as a notification since I see this ticket closed today.

Regards,
// NaHi

=end

#10 - 12/22/2010 12:02 PM - mame (Yusuke Endoh)

- Status changed from Closed to Assigned

=begin
Hi, NaHi

> Just as a notification since I see this ticket closed today.

Kazu committed a change of NEWS with ML ref, which closed this
ticket. I think that he did not intend to close this ticket.
That is, it is just accident. I'm reopening this ticket.

> Yusuke, patches in the mail I'm replying are not applied

Yes, just because of my laziness. I will work on.

> and we still need discuss about followings, right?

I think that what we needed is not discussion, but matz's final
approval.
The reason why this ticket stopped was because matz cancelled
his approval and opposed this feature at [ruby-dev:42469].
And, matz has finally re-approved it at [ruby-dev:42587].
So we can now move on.

> - Module#constants includes a private constant?

It should not. I will.

> - Module#const_get raises NameError for a private constant?

It should not. I will.

> - add Module#public_constants and Module#private_constants?

Plus, Module#public_const_get. But these may be need matz's
additional approval.

Matz, can I add these methods:

- Module#public_constants, which returns an array that contains
  only public constant names.

- Module#private_constants, which returns an array that contatins
  only private constant names.

- Module#public_const_get, which returns a value assigned to the
  constant if it is public, or raise a NameError if it is private.

?

> - Module#const_set changes the constant visibility to public? Setting visibility via const_set?

It is just a bug. I will.

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

#11 - 01/29/2011 12:07 PM - mame (Yusuke Endoh)

=begin
Hi, matz

May I introduce these methods in relation to private constants?

- Module#public_constants, as alias to Module#constants
(corresponding to Ojbect#public_methods)

- Module#private_constants, which returns an array that contains only private constant names.
(corresponding to Ojbect#private_methods)

- Module#public_const_get, which return the constant value or raise a NameError if the constant is private
(corresponding to Object#public_send)

- Module#public_const_defined?, which tells whether the public constant is defined or not
(corresponding to Module#public_method_defined?)

- Module#private_const_defined?, which tells whether the private constant is defined or not
(corresponding to Module#private_method_defined?)

A patch and tests are attached. Thank you.

Note: I fixed other bugs which RPRR and nagachika reported (r30713, r30714, r30715, r30716 and r30718). Sorry for late action.

2010/12/22 Yusuke Endoh redmine@ruby-lang.org:

> Issue #3908 has been updated by Yusuke Endoh.
>
> Status changed from Closed to Assigned
>
> Hi, NaHi
>
>> Just as a notification since I see this ticket closed today.
>
> Kazu committed a change of NEWS with ML ref, which closed this ticket. I think that he did not intend to close this ticket.
> That is, it is just accident. I'm reopening this ticket.
>
>> Yusuke, patches in the mail I'm replying are not applied
>
> Yes, just because of my laziness. I will work on.
>
>> and we still need discuss about followings, right?
>
> I think that what we needed is not discussion, but matz's final approval.
> The reason why this ticket stopped was because matz cancelled his approval and opposed this feature at [ruby-dev:42469].
> And, matz has finally re-approved it at [ruby-dev:42587].
> So we can now move on.
>
>> - Module#constants includes a private constant?
>
> It should not. I will.
>
>> - Module#const_get raises NameError for a private constant?

It should not. I will.

> - add Module#public_constants and Module#private_constants?

Plus, Module#public_const_get. But these may be need matz's
additional approval.

Matz, can I add these methods:

 - Module#public_constants, which returns an array that contains
   only public constant names.

 - Module#private_constants, which returns an array that contatins
   only private constant names.

 - Module#public_const_get, which returns a value assigned to the
   constant if it is public, or raise a NameError if it is private.

?

> - Module#const_set changes the constant visibility to public? Setting visibility via const_set?

It is just a bug. I will.

--

Yusuke Endoh mame@tsg.ne.jp

http://redmine.ruby-lang.org/issues/show/3908

---

http://redmine.ruby-lang.org

--
Yusuke Endoh mame@tsg.ne.jp

Attachment: private-constant-omake.patch
=end

#12 - 07/05/2011 01:42 AM - mame (Yusuke Endoh)

- Assignee changed from mame (Yusuke Endoh) to matz (Yukihiro Matsumoto)

- Target version changed from 1.9.3 to 2.0.0

#13 - 03/18/2012 01:38 PM - nahi (Hiroshi Nakamura)

- Status changed from Assigned to Closed

- Assignee changed from matz (Yukihiro Matsumoto) to mame (Yusuke Endoh)

- Target version changed from 2.0.0 to 1.9.3

Closing as implemented at 1.9.3.

#14 - 03/21/2012 07:53 PM - mame (Yusuke Endoh)

Hello,

2012/3/18, Hiroshi Nakamura nakahiro@gmail.com:

> Closing as implemented at 1.9.3.

Thanks.

When I think about it now, 2.0.0 were enough for
this feature. No one thought 2.0.0 would become
close to reality so soon :-)

--

Yusuke Endoh mame@tsg.ne.jp

#15 - 03/23/2012 02:08 AM - trans (Thomas Sawyer)

What was this deemed significant? I can't think of single reason why anyone would actually have to have a "private", as opposed to a "public", constant. Constants are CONSTANT so they aren't supposed to be changed after they are defined anyway --indeed normal channels of doing so will cause a warning. And constants aren't methods, so they aren't something you can call to effect object state. So what's the point? Why add all this new complexity for working with constants, e.g. #private_constants vs. #public_constants? I just see additional headache with nothing at all gained.

#16 - 03/23/2012 02:23 AM - now (Nikolai Weibull)

On Thu, Mar 22, 2012 at 18:08, trans (Thomas Sawyer)
transfire@gmail.com wrote:

> I can't think of single reason why anyone would actually have to have a "private", as opposed to a "public", constant.

Private modules and classes.

#17 - 03/23/2012 02:53 AM - aprescott (Adam Prescott)

On Thu, Mar 22, 2012 at 17:08, trans (Thomas Sawyer) transfire@gmail.comwrote:

> I can't think of single reason why anyone would actually have to have a
> "private", as opposed to a "public", constant. Constants are CONSTANT so
> they aren't supposed to be changed after they are defined anyway --indeed
> normal channels of doing so will cause a warning. And constants aren't
> methods, so they aren't something you can call to effect object state. So
> what's the point?

You can have constants defined which are purely an implementation detail
that you don't want to be public-facing, because they aren't intended to be
relied upon or modified. Hiding private modules and classes aside, I can
see a benefit from hiding that information.

#18 - 03/23/2012 08:46 AM - trans (Thomas Sawyer)

Isn't that really best left to a documentation detail?

While these private classes might be an implementation detail in your design, what if someone comes along and wants to build off the work and thus needs to subclass one of these implementation detail classes?

I can understand the thought behind it. On the surface we clearly think of certain classes and modules in our projects being the public interface and others, the majority of them in fact, being implementation details. But codifying that, rather than just documenting it, is bound to be more frustrating than useful.

#19 - 03/23/2012 09:53 AM - aprescott (Adam Prescott)

On Thu, Mar 22, 2012 at 23:46, trans (Thomas Sawyer) transfire@gmail.comwrote:

> But codifying that, rather than just documenting it, is bound to be more
> frustrating than useful.

Isn't that the point of trying to crack open an implementation detail to
rely on? It should be difficult if it's just a detail that can change at
any moment. It's not really different from having private methods. If you
really want to get at them, you still can.

#20 - 03/23/2012 07:15 PM - trans (Thomas Sawyer)

Methods are different b/c they can effect state --they can be *dangerous*. If it wasn't for that, there would be little reason to have private methods either.

So, what does this "if you really want to get at them, you still can" look like?

As I think about the scenarios of use for this feature, I just don't see it bringing anything really helpful to the table. But I do see it annoying people when there programs break as a new version of some library decided all those "implementation details" should be private.

Files

0001-separate-RCLASS_CONST_TBL-from-RCLASS_IV_TBL.patch	10.7 KB	10/06/2010	mame (Yusuke Endoh)
0002-use-rb_constant_entry_t-as-entry-of-RCLASS_CONST_TBL.patch	10.6 KB	10/06/2010	mame (Yusuke Endoh)

0003-add-Module-public_constant-and-private_constant.patch	3.25 KB	10/06/2010	mame (Yusuke Endoh)