Ruby master - Feature #16336

Allow private constants to be accessed with absolute references

11/08/2019 08:54 PM - dylants (Dylan Thacker-Smith)

Status:	Open	
Priority:	Normal	
Assignee:		
Target version:		

Description

The purpose of constant privacy is to keep a constant from being accessed from outside the namespace. As such, it can be surprising at first when something like the following doesn't work

```
# in foo/a.rb
module Foo
  class A
  end
  private_constant :A
end

# in foo/b.rb
module Foo
  class B
    Foo::A # => NameError: private constant Foo::A referenced
  end
end
```

Once the unexpected behaviour is understood, it can be worked around by using a relative constant reference (changing Foo::A to A in the example). However, the developer isn't just writing the code for the machine to understand, they are also writing it so it will be clear to other developers, where using the parent namespace in the constant reference may be appropriate for the readability of the code.

Instead, I think a reference to a private constant should be allowed through its parent namespace (as is the case with absolute constant references) if the parent namespace is in Module.nesting and the constant reference isn't made through a dynamic value (e.g. foo = Foo; foo::A wouldn't be allowed in the above example).

This will have the additional advantage that it will be easier to make a constant private, since these private constant references through their parent namespace won't have to be updated to remove the parent namespace from the constant reference.

I've opened pull request https://github.com/ruby/ruby/pull/2664 with a fix for this issue.

History

#1 - 11/08/2019 11:26 PM - jeremyevans0 (Jeremy Evans)

This would make private constant handling inconsistent with private method handling:

```
class A
  def self.b
    1
  end
  private_class_method :b

  A.b
  # NoMethodError (private method `b' called for A:Class)
end
```

That by itself isn't a reason to reject this feature, but it is something that should be considered.

#2 - 11/14/2019 06:21 PM - dylants (Dylan Thacker-Smith)

https://bugs.ruby-lang.org/issues/16123 added support for making private methods calls explicitly through self, so in class A self.b could be used to call a private class method b. If consistency with private methods is desired, then we could support referring the self through a static constant reference, although I can't think of any reason why that would be useful.