Ruby master - Feature #13332

Forwardable#def_instance_delegator nil

03/19/2017 10:52 PM - eike.rb (Eike Dierks)

Status: Feedback
Priority: Normal
Assignee: nobu (Nobuyoshi Nakada)
Target version:

Description
I'd like to suggest an extension to the signature of stdlib Forwardable#def_instance_delegator

Currently (ruby 2.3.3) it is declared as:

```ruby
def_instance_delegator(accessor, method, ali = method)
```

I'd like to add a nil: option like this:

```ruby
def_instance_delegator(accessor, method, ali = method, nil Behaviour: NoMethodError)
```

Rationale:
As of ruby2.3, delegated methods fail with
`#<NoMethodError: undefined methodfoo' for nil:NilClass>`
when the declared accessor returns nil

aka, when the delegator may be nil,
the Forwardable module can not be used,
but one has to ressort to implement the forward manually.

Adding an option to the signature could solve this.
I believe that this can be implemented in a fully backward compatible way.

I named this option :nil in my example,
(some people will cry out loud)

Specification scribble:

```ruby
if accessor.nil?
  case nil_behaviour
  when nil then nil  # return nil, when accessor was nil
  when Error         # raise that Error with proper args
  when Proc          # invoke that Proc with proper args
  when String        # return that string
  else               # return that object
  end
```

History
#1 - 03/19/2017 11:26 PM - nobu (Nobuyoshi Nakada)
- Description updated

I wonder how common such case is.

And you can override delegator/accessor methods by yourself.

#2 - 06/16/2017 07:40 AM - ko1 (Koichi Sasada)
- Assignee set to nobu (Nobuyoshi Nakada)

#3 - 09/25/2017 08:20 AM - matz (Yukihiro Matsumoto)
- Status changed from Open to Feedback

The proposed solution seems to be too generic. Show us the use-case except for nil for nil'. For example, it is acceptable that allow nil: true' keyword
argument we see in ActiveSupport.

Matz.