Ruby - Misc #20279

Is the implementation of `respond_to_missing?` in BasicObject documentation correct?

02/19/2024 05:18 AM - ioquatix (Samuel Williams)

Status:	Closed
Priority:	Normal
Assignee:	

Description

Considering the documentation here: https://ruby-doc.org/3.2.2/BasicObject.html

Introduced in: https://github.com/ruby/ruby/commit/3eb7d2b33e3f8555d81db5369eb6fb7100a91e63

I wondered if or super is correct in respond_to_missing?.

For example:

```
irb(main):001* class MyObjectSystem < BasicObject
irb(main):002*   DELEGATE = [:puts, :p]
irb(main):003*
irb(main):004*   def method_missing(name, *args, &block)
irb(main):005*     return super unless DELEGATE.include? name
irb(main):006*     ::Kernel.send(name, *args, &block)
irb(main):007*   end
irb(main):008*
irb(main):009*   public def respond_to_missing?(name, include_private = false)
irb(main):010*     DELEGATE.include?(name) or super
irb(main):011*   end
irb(main):012> end
=> :respond_to_missing?
irb(main):013> MyObjectSystem.new.respond_to_missing?(:foo)
(irb):5:in `method_missing': super: no superclass method `respond_to_missing?' for an instance of
MyObjectSystem (NoMethodError)
 from (irb):10:in `respond_to_missing?'
 from (irb):13:in `<main>'
 from <internal:kernel>:187:in `loop'
 from /home/samuel/.gem/ruby/3.3.0/gems/irb-1.11.2/exe/irb:9:in `<top (required)>'
 from /home/samuel/.gem/ruby/3.3.0/bin/irb:25:in `load'
 from /home/samuel/.gem/ruby/3.3.0/bin/irb:25:in `<main>'
```

It looks wrong to me.

In addition, I'd like to know in what situations BasicObject should define respond_to_missing? - because I was under the impression it was called by method_missing. Does BasicObject#method_missing have this behaviour? Maybe we can improve the documentation cc @burdettelamar (Burdette Lamar)

Associated revisions

Revision e127289632396f268099c9815a59bc7e7f13b3ec - 03/19/2024 12:49 PM - Earlopain (Earlopain _)

[Bug #20279] [DOC] Update for BasicObject

The current implementation raises on the call to super

Revision e127289632396f268099c9815a59bc7e7f13b3ec - 03/19/2024 12:49 PM - Earlopain (Earlopain _)

[Bug #20279] [DOC] Update for BasicObject

The current implementation raises on the call to super

Revision e1272896 - 03/19/2024 12:49 PM - Earlopain (Earlopain _)

[Bug #20279] [DOC] Update for BasicObject

The current implementation raises on the call to super

History

#1 - 02/19/2024 05:49 AM - ioquatix (Samuel Williams)

- Subject changed from `respond_to_missing?` in BasicObject documentation correct? to Is the implementation of `respond_to_missing?` in BasicObject documentation correct?

#2 - 02/19/2024 08:06 AM - byroot (Jean Boussier)

> because I was under the impression it was called by method_missing.

respond_to_missing? isn't called by method_missing but by Kernel#respond_to?, and BasicObject doesn't define respond_to?.

If you want to implement a delegator that responds to respond_to?, you need to copy the method over from Kernel:

```
class Proxy < BasicObject
  DELEGATE = [:puts, :p]

  define_method(:respond_to?, ::Kernel.instance_method(:respond_to?))

  private

  define_method(:respond_to_missing?, ::Kernel.instance_method(:respond_to_missing?))

  def method_missing(name, *args, &block)
    return super unless DELEGATE.include? name
    ::Kernel.send(name, *args, &block)
  end

  def respond_to_missing?(name, include_private = false)
    DELEGATE.include?(name) or super
  end
end

proxy = Proxy.new
p proxy.respond_to?(:puts)
proxy.puts "Hello"

true
Hello
```

#3 - 02/19/2024 08:08 AM - byroot (Jean Boussier)

So yes, the documentation is incorrect.

#4 - 02/19/2024 09:51 AM - ioquatix (Samuel Williams)

Do you want to submit a PR? You already wrote most of the code... however:

```
  define_method(:respond_to_missing?, ::Kernel.instance_method(:respond_to_missing?))

  def respond_to_missing?(name, include_private = false)
    DELEGATE.include?(name) or super
  end
```

Won't these clobber each other?

#5 - 02/19/2024 09:52 AM - byroot (Jean Boussier)

> Won't these clobber each other?

Yeah, my bad. You either need to copy it in an included module, or just not copy it over and not call super.

#6 - 03/19/2024 02:17 PM - Anonymous

- Status changed from Open to Closed

Applied in changeset git|e127289632396f268099c9815a59bc7e7f13b3ec.

[Bug #20279] [DOC] Update for BasicObject

The current implementation raises on the call to super