Ruby master - Feature #17401

The class `Ractor::MovedObject` should not be frozen

12/17/2020 12:59 AM - marcandre (Marc-Andre Lafontune)

Status: Closed
Priority: Normal
Assignee: ko1 (Koichi Sasada)
Target version: 

Description

If Ractor::MovedObject can be not frozen that would be more helpful than currently. For example, to implement a helpful inspect in pure Ruby 3.0 like in #17393

It is currently possible to go around the fact that it is frozen with refinements:

```ruby
using Module

refine Ractor::MovedObject do
  def inspect
    "I was moved! Don't use me!"
  end
end

o = Object.new
Ractor.new{ receive }.send(o, move:true)
puts o.inspect # => "I was moved! Don't use me!"
```

Another (ugly) way to bypass the fact that it is frozen without refinements is to modify its ancestor:

```ruby
module Ext
  def inspect
    return super unless Ractor::MovedObject === self

    "I was moved! Don't use me!"
  end
end

BasicObject.prepend Ext

o = Object.new
Ractor.new{ receive }.send(o, move:true)
p o #=> "I was moved! Don't use me!"
```

My point remain that this "security" can be bypassed unless it is strictly necessary it, it would be nice to unfreeze the class. It could be documented that this is experimental / not recommended to modify the class in production, for example.

Associated revisions

Revision 76e88480 - 12/18/2020 08:57 PM - ko1 (Koichi Sasada)

unfreeze Ractor::MovedObject

Matz prefers to unfreeze the class.
[Feature #17401]

History

#1 - 12/18/2020 08:56 PM - ko1 (Koichi Sasada)
Matz prefers to provide a free to shoot the foot.

#2 - 12/18/2020 08:59 PM - ko1 (Koichi Sasada)
- Status changed from Open to Closed

Applied in changeset gili76e88480371469400346fca609efe67096813915.
unfreeze Ractor::MovedObject

Matz prefers to unfreeze the class.
[Feature #17401]