# Ruby - Bug #21368

## Moving objects with finalizer between Ractors crashes

05/23/2025 07:57 PM - peterzhu2118 (Peter Zhu)

| | | | |
|---|---|---|---|
| **Status:** | Assigned | | |
| **Priority:** | Normal | | |
| **Assignee:** | ractor | | |
| **Target version:** | | | |
| **ruby -v:** | | **Backport:** | 3.2: UNKNOWN, 3.3: UNKNOWN, 3.4: UNKNOWN |

**Description**

When an object is moved to a different Ractor, the finalizers are not copied to the new object, so it will have the FL_FINALIZE flag set but no entry in the finalizer table.

The following script crashes:

```
r = Ractor.new do
  loop { Ractor.receive }
end

1_000.times do
  o = Object.new
  ObjectSpace.define_finalizer(o, proc { |id| })
  r.send(o, move: true)
end
```

## History

**#1 - 05/23/2025 08:04 PM - peterzhu2118 (Peter Zhu)**

*- Assignee set to ractor*

**#2 - 05/27/2025 03:03 PM - zzak (zzak _)**

It seemed fun to patch, but I'm not sure this is correct:
https://github.com/ruby/ruby/pull/13452

**#3 - 05/28/2025 05:07 AM - hsbt (Hiroshi SHIBATA)**

*- Status changed from Open to Assigned*

**#4 - 05/28/2025 02:29 PM - osyoyu (Daisuke Aritomo)**

*- File 0001-Bug-21368-Transfer-finalizer-when-moving-between-Rac.patch added*

The script in the description didn't crash in my environments (the following). Explicitly calling GC.start did.

- Linux ruby 3.5.0dev (2025-05-28T04:34:40Z master d064fd067b) +PRISM [x86_64-linux]
- macOS ruby 3.5.0dev (2025-05-28T04:34:40Z master d064fd067b) +PRISM [arm64-darwin24]

```
r = Ractor.new do
  Ractor.receive # don't bind to a variable (let it be garbage collected)
  GC.start
end

o = Object.new
ObjectSpace.define_finalizer(o, proc { |id| })
r.send(o, move: true)

r.take
```

I have attached an updated version of @zzak (zzak _) 's patch which resolves this crash, but I suppose this is inappropiate since this will leak outer variables in the following case:

```
r = Ractor.new do
  Ractor.receive
  GC.start # `unshareable` happens to get read in this Ractor
end

o = Object.new
unshareable = +"hello"
ObjectSpace.define_finalizer(o, proc { |id| p unshareable })
r.send(o, move: true)

r.take
```

Maybe simply marking objects with a finalizer ineligible for moving is more appropiate.

#### #5 - 06/09/2025 07:09 AM - zzak (zzak _)

> Maybe simply marking objects with a finalizer ineligible for moving is more appropiate.

Thanks for checking, I've updated the PR to raise if the object has a finalizer. I'm not sure if we should do the check in like make_shareable_check_shareable instead, for example.

#### #6 - 06/11/2025 10:46 AM - osyoyu (Daisuke Aritomo)

Maybe this ticket should be merged with https://bugs.ruby-lang.org/issues/21315 ?

### Files

| | | | |
|---|---|---|---|
| 0001-Bug-21368-Transfer-finalizer-when-moving-between-Rac.patch | 883 Bytes | 05/28/2025 | osyoyu (Daisuke Aritomo) |