Ruby - Bug #21298

`ObjectSpace.allocation_class_path` returns inconsistent results depending on `TracePoint` state

05/01/2025 07:43 AM - mame (Yusuke Endoh)

Status:	Assigned		
Priority:	Normal		
Assignee:	tenderlovemaking (Aaron Patterson)		
Target version:			
ruby -v:		**Backport:**	3.2: UNKNOWN, 3.3: UNKNOWN, 3.4: UNKNOWN

Description

ObjectSpace.allocation_class_path is an API that returns the class of self in the context where an object was allocated. However, due to recent optimizations in Class#new (#21254), the return value now changes depending on whether TracePoint is enabled.

```
require "objspace"

class Foo
  def test
    obj = Object.new
    ObjectSpace.allocation_class_path(obj)
  end
end

ObjectSpace.trace_object_allocations_start

p Foo.new.test #=> 3.4.2: "Class", master: "Foo"
```

Previously, this returned "Class" (from the Class#new call frame), but in the master branch, the result is now "Foo" because that frame is gone.

I am ok for the incompatibility itself because I find the new behavior more intuitive and useful. However, there's an inconsistency: the optimization is disabled when TracePoint is enabled, causing the result to revert to the old behavior.

```
p Foo.new.test #=> master: "Foo"

TracePoint.new {}.enable do
  p Foo.new.test #=> master: "Class", expected: "Foo"
end
```

This makes behavior dependent on whether TracePoint is enabled, which can lead to confusion.

@ko1 (Koichi Sasada) @tenderlovemaking (Aaron Patterson) Can we make ObjectSpace.allocation_class_path consistently return the class from the .new call context, regardless of the TracePoint state?

I am facing a failure of the following test when code coverage (which uses TracePoint) is enabled:

https://github.com/ruby/ruby/blob/e8ad728209ee22136e61054fea74096b49088b8a/test/objspace/test_objspace.rb#L206

As a short-term workaround, I'm considering commenting out this test.

Related issues:

Related to Ruby - Feature #21254: Inlining Class#new	**Closed**

Associated revisions

Revision 5cee3329 - 05/01/2025 08:21 AM - mame (Yusuke Endoh)

Skip test affected by TracePoint-dependent allocation_class_path

These assertions fail when TracePoint is enabled due to differing
allocation context. Commented out for now until behavior is fixed.

See [Bug #21298]

History

#1 - 05/01/2025 07:43 AM - mame (Yusuke Endoh)

- Related to Feature #21254: Inlining Class#new added

#2 - 05/01/2025 08:21 AM - mame (Yusuke Endoh)

- Status changed from Open to Closed

Applied in changeset git|5cee3329df2963667d958cc7bb091f77ae9172aa.

Skip test affected by TracePoint-dependent allocation_class_path

These assertions fail when TracePoint is enabled due to differing
allocation context. Commented out for now until behavior is fixed.

See [Bug #21298]

#3 - 05/01/2025 08:22 AM - mame (Yusuke Endoh)

- Status changed from Closed to Open

#4 - 05/01/2025 07:15 PM - tenderlovemaking (Aaron Patterson)

mame (Yusuke Endoh) wrote:

> I am ok for the incompatibility itself because I find the new behavior more intuitive and useful. However, there's an inconsistency: the optimization is disabled when TracePoint is enabled, causing the result to revert to the old behavior.

Yes, we can keep the behavior consistent. I would rather keep the behavior consistent, but some TracePoint tests were expecting the frame so I tried to maintain backwards compatibility.

I sent a PR to make the behavior consistent here.

#5 - 05/01/2025 07:30 PM - tenderlovemaking (Aaron Patterson)

I remember the problem now. power_assert expects to find the Class#new frame:
https://github.com/ruby/ruby/actions/runs/14781787226/job/41502113192?pr=13232

I want to make the behavior consistent, but I'm not sure what to do about power_assert

#6 - 05/02/2025 12:14 AM - mame (Yusuke Endoh)

@tenderlovemaking (Aaron Patterson) Oh thank you!

@ktsj (Kazuki Tsujimoto) Would it be possible for power_assert to avoid depending on Class#new being present in the stack trace?

#7 - 05/05/2025 05:27 AM - ktsj (Kazuki Tsujimoto)

> Would it be possible for power_assert to avoid depending on Class#new being present in the stack trace?

I updated power_assert.
https://github.com/ruby/power_assert/pull/56

Now, it should work.

#8 - 05/12/2025 11:16 PM - hsbt (Hiroshi SHIBATA)

- Status changed from Open to Assigned

#9 - 06/03/2025 05:47 AM - mame (Yusuke Endoh)

Can we go ahead? This will bring incompatibility, so I think this should be fixed by the release. @ktsj (Kazuki Tsujimoto) @tenderlovemaking (Aaron Patterson)