

## Ruby trunk - Bug #15674

### Regression: ObjectSpace.allocation\_sourcefile produces the wrong location for allocations, it shows the end of the method instead of the line where the object was created

03/18/2019 03:49 PM - schneems (Richard Schneeman)

<b>Status:</b> Open	
<b>Priority:</b> Normal	
<b>Assignee:</b>	
<b>Target version:</b>	
<b>ruby -v:</b> 2.6.2	<b>Backport:</b> 2.4: UNKNOWN, 2.5: UNKNOWN, 2.6: UNKNOWN

#### Description

Originally I opened this up as an issue in memory profiler [https://github.com/SamSaffron/memory\\_profiler/issues/67](https://github.com/SamSaffron/memory_profiler/issues/67), however I was able to isolate the issue to just Ruby 2.6.

ObjectSpace.allocation\_sourcefile is reporting the line where a method ends rather than the line where the object was actually allocated.

Here's the script. You would expect that object space would tell me the allocation happens on line 2 but instead it gets reported as line 3:

```
def my_concat(a, b)
  return a + b # <=====
end

require 'fileutils'
require 'objspace'
FileUtils.mkdir_p("tmp")

def run
  ObjectSpace.trace_object_allocations do
    10.times do
      my_concat("hello".freeze, "world".freeze)
    end

    10.times do
      my_concat("hello".freeze, "world".freeze)
    end
  end

  ObjectSpace.each_object do |obj|
    file = ObjectSpace.allocation_sourcefile(obj)
    line = ObjectSpace.allocation_sourcefile(obj)
    next unless file && line
    puts "#{file}:#{line}"
  end
end

run
```

If you add space to the method body it keeps incrementing the line number of the output. For example, changing the method body to this, should not change the allocation location:

```
def my_concat(a, b)
  return a + b # <=====

end
```

However, it is reported as a different line:

```
script.rb:8
```

I tested with Ruby 2.3, 2.4, and 2.5 and all those versions produce a correct result, this regression is limited to only 2.6.