# Ruby - Feature #3001

## Ruby stdlib: Benchmark::Tms #memberwise drops labels

03/24/2010 06:21 AM - DrErnie (Ernest Prabhakar)

| | |
|---|---|
| **Status:** | Feedback |
| **Priority:** | Normal |
| **Assignee:** | |
| **Target version:** | |

**Description**

=begin
http://ruby-doc.org/stdlib/libdoc/benchmark/rdoc/classes/Benchmark/Tms.html#M000016-source

The implementation of Benchmark::Tms#memberwise only passes the computed time values when creating a new instance, e.g.:

```
    Benchmark::Tms.new(utime.__send__(op, x),
                       stime.__send__(op, x),
                       cutime.__send__(op, x),
                       cstime.__send__(op, x),
                       real.__send__(op, x)
                       )
```

It would seem consistent (and more informative) if it also passed in the current label (especially since there's no other way to set it):

```
    Benchmark::Tms.new(utime.__send__(op, x),
                       stime.__send__(op, x),
                       cutime.__send__(op, x),
                       cstime.__send__(op, x),
                       real.__send__(op, x),
                       label
                       )
```

Example:

```
t =  Benchmark.measure("foo") { sleep 0.1 }
=> #<Benchmark::Tms:0x1011d03d8 @cstime=0.0, @total=0.0, @cutime=0.0, @label="foo", @stime=0.0,
@real=0.100059986114502, @utime=0.0>
t2 = t / 2
=> #<Benchmark::Tms:0x1011be6d8 @cstime=0.0, @total=0.0, @cutime=0.0, @label=nil, @stime=0.0,
@real=0.050029993057251, @utime=0.0>
=end
```

## History

### #1 - 04/09/2010 01:06 AM - mame (Yusuke Endoh)

=begin
Hi,

> It would seem consistent (and more informative) if it also passed in the current label (especially since there's no other way to set it):

It is confusing to preserve the entirely same label, I guess.

If you really need the feature, a "derived" mark would be better,
such as "foo (/ 2)"

Viewed in this light, this is a new feature rather than bug.
So I move this ticket into Feature tracker.

--
Yusuke Endoh mame@tsg.ne.jp
=end

**#2 - 04/09/2010 01:07 AM - mame (Yusuke Endoh)**

*- Target version set to 2.0.0*

=begin

=end

**#3 - 07/27/2011 04:28 PM - drbrain (Eric Hodel)**

*- Category set to lib*

**#4 - 07/27/2011 10:58 PM - Eregon (Benoit Daloze)**

Hi,

> It would seem consistent (and more informative) if it also passed in the current label (especially since there's no other way to set it):

What about (tms1 + tms2)'s label ? Should it be "#{tms1.label} #{op} #{tms2.label}" ?
It seems consistent for operations with both Tms and others ("foo / 2").

But I'm not sure how this would be useful, could you give me an example ?
Maybe the need is somewhere else.

**#5 - 03/18/2012 03:37 PM - nahi (Hiroshi Nakamura)**

*- Status changed from Open to Feedback*

**#6 - 10/27/2012 05:34 AM - ko1 (Koichi Sasada)**

*- Target version changed from 2.0.0 to 2.6*

I changed target to next minor because there is no discussion on it.
No feedback?

**#7 - 05/24/2016 01:42 PM - cjeon (CHEOLHO JEON)**

Ernest Prabhakar wrote:

> =begin
> http://ruby-doc.org/stdlib/libdoc/benchmark/rdoc/classes/Benchmark/Tms.html#M000016-source
>
> The implementation of Benchmark::Tms#memberwise only passes the computed time values when creating a new instance, e.g.:
>
> ```
>      Benchmark::Tms.new(utime.__send__(op, x),
>                         stime.__send__(op, x),
>                         cutime.__send__(op, x),
>                         cstime.__send__(op, x),
>                         real.__send__(op, x)
>                         )
> ```
>
> It would seem consistent (and more informative) if it also passed in the current label (especially since there's no other way to set it):
>
> ```
>      Benchmark::Tms.new(utime.__send__(op, x),
>                         stime.__send__(op, x),
>                         cutime.__send__(op, x),
>                         cstime.__send__(op, x),
>                         real.__send__(op, x),
>                         label
>                         )
> ```
>
> Example:
>
>> t =  Benchmark.measure("foo") { sleep 0.1 }
>> => #<Benchmark::Tms:0x1011d03d8 @cstime=0.0, @total=0.0, @cutime=0.0, @label="foo", @stime=0.0, @real=0.100059986114502, @utime=0.0>
>> t2 = t / 2
>> => #<Benchmark::Tms:0x1011be6d8 @cstime=0.0, @total=0.0, @cutime=0.0, @label=nil, @stime=0.0, @real=0.050029993057251, @utime=0.0>
>> =end

any update on this? I think it should not drop labels...