Ruby - Feature #6315

handler to trace output of each line of code executed

04/18/2012 01:23 PM - ankopainting (Anko Painting)

Status:	Feedback	
Priority:	Normal	
Assignee:		
Target version:		

Description

using a similar mechanism to set_trace_func, I would like a way to get the line number and return value of each line. This would allow me to create a live debugger, much like light table from chris granger and/or bret victor's javascript demo.

It could be an extra return value of set_trace_func or we could use a new method.

History

#1 - 04/19/2012 03:23 PM - ko1 (Koichi Sasada)

(2012/04/18 13:23), ankopainting (Anko Painting) wrote:

> using a similar mechanism to set_trace_func, I would like a way to get the line number and return value of each line. This would allow me to create a live debugger, much like light table from chris granger and/or bret victor's javascript demo.
>
> It could be an extra return value of set_trace_func or we could use a new method.

It is interesting because I will fix debugger API.

However, I don't understand what you want and what set_trace_func lacks.
Could you give me examples?
(for example, your favorite "set_trace_func2", the extended set_trace_func)

Regards,
Koichi

// SASADA Koichi at atdot dot net

#2 - 04/20/2012 03:31 AM - mame (Yusuke Endoh)

- Status changed from Open to Feedback

#3 - 04/20/2012 04:46 AM - trans (Thomas Sawyer)

I was under the impression that set_trace_func would be deprecated. Last I checked it had a glaring bug in it which I reported.

#4 - 04/20/2012 02:33 PM - ankopainting (Anko Painting)

ko1 (Koichi Sasada) wrote:

> (2012/04/18 13:23), ankopainting (Anko Painting) wrote:
>
> > using a similar mechanism to set_trace_func, I would like a way to get the line number and return value of each line. This would allow me to create a live debugger, much like light table from chris granger and/or bret victor's javascript demo.
> >
> > It could be an extra return value of set_trace_func or we could use a new method.
>
> It is interesting because I will fix debugger API.

That's great - I see a good debugger as very important for a language.

> However, I don't understand what you want and what set_trace_func lacks.
> Could you give me examples?
> (for example, your favorite "set_trace_func2", the extended set_trace_func)

I asked the question on ruby forum here: http://www.ruby-forum.com/topic/4072087#new

It has an example, but I'll give you another one;

```
#!ruby


set_trace_func2 proc { |event, file, line, id, binding, classname, return_value|
if event == "line"
STDERR.puts "#{line}: -> #{return_value.inspect}"
end
}

i = 4
words = %w!brat cat apple!.sort
```

end of ruby

```
output;
10: -> 4
11: -> ["apple", "brat", "cat"]
```

This is my first, simple idea. It would allow your editor to show all return values next to your lines of code by evaluating it all in the browser.

I will put in a few more set_trace_func feature requests when I can give decent examples to explain them.

Thank you.

#5 - 04/20/2012 03:23 PM - ko1 (Koichi Sasada)

Hi,

(2012/04/20 14:33), ankopainting (Anko Painting) wrote:

> It has an example, but I'll give you another one;
>
> ```
> #!ruby
>
>
> set_trace_func2 proc { |event, file, line, id, binding, classname, return_value|
> if event == "line"
> STDERR.puts "#{line}: -> #{return_value.inspect}"
> end
> }
>
> i = 4
> words = %w!brat cat apple!.sort
> ```
>
> # end of ruby
>
> ```
> output;
> 10: -> 4
> 11: -> ["apple", "brat", "cat"]
> ```
>
> ---
>
> This is my first, simple idea. It would allow your editor to show all return values next to your lines of code by evaluating it all in the browser.
>
> I will put in a few more set_trace_func feature requests when I can give decent examples to explain them.

Thank you. I understand what you want. However, I think it needs
performance drawback. I'm not sure it is valuable feature or not
against slow down.

--
// SASADA Koichi at atdot dot net

#6 - 04/20/2012 07:24 PM - ankopainting (Anko Painting)

> Thank you. I understand what you want. However, I think it needs
> performance drawback. I'm not sure it is valuable feature or not
> against slow down.

--
// SASADA Koichi at atdot dot net

I understand your concern. I think power > speed. In ruby this feature is impossible currently.

please watch http://vimeo.com/40281991. It is 5 minutes but shows the power of such features.

I think it would be okay if it was kept separate to set_trace_func so current applications would not be affected.

thank you for taking the time to listen to my idea.

#7 - 04/20/2012 07:53 PM - ko1 (Koichi Sasada)

(2012/04/20 19:24), ankopainting (Anko Painting) wrote:

> I understand your concern. I think power > speed. In ruby this feature is impossible currently.
>
> please watch http://vimeo.com/40281991. It is 5 minutes but shows the power of such features.

I agree with "power > speed". However, I'm not sure how power grows
with this feature. I'm sorry I can't see your suggested video (because
I'm at outside). Could you write the power?

Or any other comments from other guys?

--
// SASADA Koichi at atdot dot net

#8 - 04/21/2012 07:01 AM - ankopainting (Anko Painting)

ko1 (Koichi Sasada) wrote:

> (2012/04/20 19:24), ankopainting (Anko Painting) wrote:
>
> > I understand your concern. I think power > speed. In ruby this feature is impossible currently.
> >
> > please watch http://vimeo.com/40281991. It is 5 minutes but shows the power of such features.
>
> I agree with "power > speed". However, I'm not sure how power grows
> with this feature. I'm sorry I can't see your suggested video (because
> I'm at outside). Could you write the power?
>
> Or any other comments from other guys?
>
> --
> // SASADA Koichi at atdot dot net

Maybe you can see the version on youtube? http://www.youtube.com/watch?v=H58-n7uldoU I am outside too.

Basically it's an IDE that evaluates your whole code when you press a key and shows the return values next to each line.

It is hard to write about but here is an ascii representation of an IDE

```
a = 3                  => 3
(0..3).each do |i|
if i == 1
a = 5             =>    | 5 |
else
a = i             =>  0 |   | 2 | 3
end
end
```

as you can see, the advantage that this has over irb is that you can see as values change over time in loops instantly, without stepping through them. You could see how changing one thing would ripple through your code.

#9 - 11/20/2012 10:35 PM - mame (Yusuke Endoh)

- Target version set to 2.6

#10 - 12/25/2017 06:15 PM - naruse (Yui NARUSE)

- Target version deleted (2.6)