Ruby - Feature #15112

Introducing the short form of `STDERR.puts expr.inspect`.

09/13/2018 06:15 AM - mrkn (Kenta Murata)

Status:	Assigned	
Priority:	Normal	
Assignee:	matz (Yukihiro Matsumoto)	
Target version:		

Description

I sometimes write STDERR.puts obj.inspect to print debug message to standard error.
I want to write it as STDERR.p obj.
It can be realized by introducing p singleton method in STDERR object.

Related issues:

Related to Ruby - Feature #14609: Let `Kernel#p` without an argument print th...	**Rejected**

History

#1 - 09/13/2018 07:04 AM - duerst (Martin Dürst)

#warn writes to STDERR. What about warn_p or something similar?

#2 - 09/13/2018 07:28 AM - jeremyevans0 (Jeremy Evans)

mrkn (Kenta Murata) wrote:

> I sometimes write STDERR.puts obj.inspect to print debug message to standard error.
> I want to write it as STDERR.p obj.
> It can be realized by introducing p singleton method in STDERR object.

I've heard it is preferable to use $stderr (the current standard error stream) instead of STDERR (the original standard error stream) as warn and similar methods use $stderr. As $stderr can be reassigned, there is no guarantee that $stderr.p would be valid. I suppose you could have STDERR.p write to $stderr, but that may be confusing. It may be more generally useful to add a method to IO that did the equivalent of puts obj.inspect.

#3 - 09/13/2018 04:07 PM - shevegen (Robert A. Heiler)

I agree with both what Kenta Murata and what Jeremy Evans wrote (to explain,
I have no personal preference; I think both .p() would be useful anyway and
I think this was the spirit of the original suggestion; I love p and pp).

Martin suggested warn_p so I think this addresses more Kenta Murata, but I
think, personally, the net benefit of warn_p may be significantly smaller
than the "puts" versus "p" situation, where I think the gain is quite
significant here, if you write a lot of code that makes use of p for
output. (This is a bit similar to the proposal to do "require 'pp'" by
default, which was a good change IMO).

Anyway, my personal opinion is +1 to the idea behind the proposal - I have
no real preference on it being on STDERR or $stderr or both; I think it may
be useful on its own, no matter where the method ultimately resides.

#4 - 09/24/2018 11:18 PM - nobu (Nobuyoshi Nakada)

Any object which has #write method can be assigned to $stderr, so we should not expect $stderr to have p, I think.

#5 - 10/10/2018 05:30 AM - mrkn (Kenta Murata)

- Related to Feature #14609: Let `Kernel#p` without an argument print the receiver added

#6 - 10/10/2018 05:51 AM - usa (Usaku NAKAMURA)

I propose more generalized method on IO instead of STDERR.p.
My proposal is IO#putp.

#7 - 10/12/2018 02:02 AM - mrkn (Kenta Murata)

- Subject changed from Introdudce the new singleton method STDERR.p to Introducing the short form of `STDERR.puts expr.inspect`.

I totally agree with Jeremy and Usaku, so I want to change the title of this issue.

#8 - 04/03/2024 03:50 AM - hsbt (Hiroshi SHIBATA)

- Status changed from Open to Assigned