

Ruby - Feature #2709

\$VERBOSE, \$DEBUG and Kernel#sprintf

02/03/2010 07:00 PM - paddor (Patrik Wenger)

Status:	Rejected	
Priority:	Normal	
Assignee:	shyouhei (Shyouhei Urabe)	
Target version:	3.0	
Description		
<p>=begin</p> <p>Kernel#sprintf is very useful. I think the power lies within the ignorance of too many arguments. So one is free to decide which arguments he uses and which one he doesn't. This applies to a Ruby runtime where \$VERBOSE and \$DEBUG are false.</p> <p>Following the POLS (Principle of the least surprise), one would assume that changing the variables \$VERBOSE and \$DEBUG wouldn't change the behavior of Kernel#sprintf. But they do.</p> <p>\$VERBOSE causes warnings when there are too many arguments.</p> <p>\$DEBUG causes an ArgumentError when there are too many arguments. This isn't useful at all.</p> <p>I like the effect of \$DEBUG according to threads, to raise uncaught exceptions from the thread.</p> <p>But in the case of Kernel#sprintf, it's just not useful. This behavior prohibits some power of Ruby.</p> <p>My suggestion:</p> <p>Remove the special behavior of \$VERBOSE and \$DEBUG in Kernel#sprintf.</p> <p>Thank you very much.</p> <p>=end</p>		
Related issues:		
Related to Ruby - Bug #5763: sprintf not throwing error for wrong number of a...		Closed 12/15/2011

History

#1 - 02/03/2010 08:08 PM - shyouhei (Shyouhei Urabe)

=begin

I think the power lies within the ignorance of too many arguments. So one is free to decide which arguments he uses and which one he doesn't.

Can you be a bit more verbose here? Please illustrate a tiny example where ignoring sprintf argumens is powerful.

=end

#2 - 02/03/2010 10:31 PM - paddor (Patrik Wenger)

=begin

Can you be a bit more verbose here? Please illustrate a tiny example where ignoring sprintf argumens is powerful.

At our company we have a framework to execute commands in telnet sessions on many hosts in parallel. This framework is used in many scripts. Depending on the scripts, the needs for log files are different. Some scripts write one log file per host and append to it in a second run. Other scripts write one log file per telnet session, so the name of the host and some date/time integers are involved.

And that's exactly what the framework provides to the freely configurable log file format string:
hostname, year, month, day, hour, minute, second

The former scripts could then use a log path like this:
"~/log/%s"

The latter one something like this (a directory each day, a file per host and session):
"/path/to/session/logs/%2\$i-%3\$02i-%4\$02i/%1\$s_%5\$02i%6\$02i%7\$02i"

I think this is power of Ruby. And it shouldn't be prohibited, just because one is debugging. It wouldn't change the default behavior (\$DEBUG and \$VERBOSE are false) anyway. :-)

=end

#3 - 02/09/2010 07:29 AM - paddor (Patrik Wenger)

=begin
Any updates here?
=end

#4 - 02/09/2010 10:18 AM - matz (Yukihiro Matsumoto)

=begin
Hi,

In message "Re: [\[ruby-core:28029\]](#) [Bug [#2709](#)] \$VERBOSE, \$DEBUG and Kernel#sprintf"
on Wed, 3 Feb 2010 19:00:23 +0900, Patrik Wenger redmine@ruby-lang.org writes:

[Following the POLS (Principle of the least surprise), one would assume that changing the variables \$VERBOSE and \$DEBUG wouldn't change the behavior of Kernel#sprintf. But they do.

We don't accept any proposal with the POLS reason by default, since the background vary person to person. So don't be so lazy. Explain why YOU have assumed Kernel#sprintf not to change behavior according to the variables \$VERBOSE and \$DEBUG.

matz.

=end

#5 - 02/09/2010 12:22 PM - shyouhei (Shyouhei Urabe)

=begin
Yukihiro Matsumoto wrote:

We don't accept any proposal with the POLS reason by default, since the background vary person to person. So don't be so lazy. Explain why YOU have assumed Kernel#sprintf not to change behavior according to the variables \$VERBOSE and \$DEBUG.

Take a look at [\[ruby-core:28032\]](#). My current feeling is neutral ... I have been helped by printf's argument checks many time so it is useful for me at least, but it seems the reporter does have needs to ignore it.

What about it: raise exceptions as it is when arguments are too *few*, but not for too *many*, even when \$DEBUG is true.

Signed-off-by: Urabe, Shyuohei shyouhei@ruby-lang.org

sprintf.c | 2 +-
1 files changed, 1 insertions(+), 1 deletions(-)

diff --git a/sprintf.c b/sprintf.c
index b5cad94..ebb26eb 100644
--- a/sprintf.c
+++ b/sprintf.c
@@ -1057,7 +1057,7 @@ rb_str_format(int argc, const VALUE *argv, VALUE fmt)
*/
if (posarg >= 0 && nextarg < argc) {
const char *mesg = "too many arguments for format string";

• if (RTEST(ruby_debug)) rb_raise(rb_eArgError, "%s", mesg);

• if (RTEST(ruby_debug)) rb_warn("%s", mesg);
if (RTEST(ruby_verbose)) rb_warn("%s", mesg);
}
rb_str_resize(result, blen);

Attachment: signature.asc
=end

#6 - 02/10/2010 02:05 AM - paddor (Patrik Wenger)

=begin
Yukihiro Matsumoto wrote:

We don't accept any proposal with the POLS reason by default, since the background vary person to person. So don't be so lazy. Explain why YOU have assumed Kernel#sprintf not to change behavior according to the variables \$VERBOSE and \$DEBUG.

I assumed Kernel#sprintf doesn't change behavior according to the variables \$VERBOSE and \$DEBUG because there's absolutely no reason to raise if there are too *many* arguments.

The work can be done by Kernel#sprintf without errors (like when \$DEBUG evaluates to false).

Shyouhei Urabe wrote:

What about it: raise exceptions as it is when arguments are too *few*, but not for too *many*, even when \$DEBUG is true.

Thank you, Shyouhei Urabe. That's what I meant. That would be great.

Sorry if I'm wrong, but the change set shown in your post, wouldn't it cause two messages to be printed, if \$VERBOSE and \$DEBUG are true? Maybe the line should look like this:

```
if (RTEST(ruby_debug) || RTEST(ruby_verbose)) rb_warn("%s", msg);
```

By the way, is this warning really needed? Same reason: If there are too many arguments, there's nothing to complain about. The functionality of Kernel#sprintf isn't affected by too many arguments. So removing the warning as well would be wise I think.

```
=end
```

#7 - 02/13/2010 05:11 PM - matz (Yukihiro Matsumoto)

```
=begin  
Hi,
```

In message "Re: [\[ruby-core:28130\]](#) [Feature [#2709](#)] \$VERBOSE, \$DEBUG and Kernel#sprintf" on Wed, 10 Feb 2010 02:05:11 +0900, Patrik Wenger redmine@ruby-lang.org writes:

|Thank you, Shyouhei Urabe. That's what I meant. That would be great.

OK, I am happy to know what you want. Could you explain why you want? Too many arguments are error anyway, so why do you want to keep it warning, *when you are debugging*?

```
matz.
```

```
=end
```

#8 - 02/17/2010 10:05 PM - paddor (Patrik Wenger)

```
=begin  
Because, in my opinion, too many arguments aren't an error. Why is this an error for you?  
Too many arguments have no impact on the functionality of Kernel#sprintf.  
Too few arguments are an error, however.
```

My suggestion:

1. never raise or warn on too many arguments
2. always raise on too few arguments

I figured out that if you provide absolute position arguments ("%1\$s"), no warnings are issued, even if \$VERBOSE or \$DEBUG. That's the way it should work with relative position arguments ("%s"), too.

```
=end
```

#9 - 04/04/2010 01:37 AM - znz (Kazuhiro NISHIYAMA)

- Category set to core
- Target version set to 2.0.0

```
=begin  
  
=end
```

#10 - 03/18/2012 03:16 PM - nahi (Hiroshi Nakamura)

- Description updated
- Assignee set to shyouhei (Shyouhei Urabe)

#11 - 03/18/2012 06:46 PM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Assigned

#12 - 10/27/2012 05:54 AM - shyouhei (Shyouhei Urabe)

- Target version changed from 2.0.0 to 3.0

I personally like this feature so I really hesitate to reject, but as matz do not like it so much, there are relatively few chances for the proposal now.

We have to persuade him.

#13 - 10/27/2012 09:54 AM - trans (Thomas Sawyer)

+1 The behavior seems too surprising. In general I do not think \$DEBUG and \$VERBOSE should alter execution, but rather only serve to change information provided to user/developer. In other words, it should be an error, or not, independent of \$DEBUG.

#14 - 10/22/2017 01:08 PM - mame (Yusuke Endoh)

Closing this because of no discussion for five years. Shyouhei-san or anyone, feel free to reopen and facilitate the discussion if you still have an interest.

#15 - 11/27/2017 12:38 PM - mame (Yusuke Endoh)

- Status changed from Assigned to Rejected