

Ruby master - Feature #9025

Clarify the error message when calling a method with the wrong number of arguments

10/16/2013 03:15 AM - Nerian (Gonzalo Rodríguez)

Status:	Closed
Priority:	Normal
Assignee:	
Target version:	
Description	
Currently when calling a method with the wrong number of arguments we get a confusing error message:	
<code>ArgumentError: wrong number of arguments (1 for 0)</code>	
That means that the method was meant to accept 0 arguments, but 1 was provided instead. This error message is confusing, and a large number of people had to search for its meaning. For example [1] has 11000 views.	
I propose that we change the error message to something whose meaning is obvious. Examples:	
<ul style="list-style-type: none">• <code>ArgumentError: wrong number of arguments (expected: 1, provided: 0)</code>• <code>ArgumentError: wrong number of arguments (1 instead of 0)</code>	
This ticket originated from this pull request: https://github.com/ruby/ruby/pull/367	
[1] http://stackoverflow.com/questions/7537450/what-does-wrong-number-of-arguments-1-for-0-mean-in-ruby	
Related issues:	
Related to Ruby master - Bug #11617: Fix tests for Feature 9025	Closed

Associated revisions

Revision 927e18b3 - 10/24/2015 03:47 AM - duerst (Martin Dürst)

vm_inshelper.c: improved error message for "wrong number of arguments", distinguishing given and expected argument numbers clearly. [Feature #9025]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@52264 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 52264 - 10/24/2015 03:47 AM - duerst (Martin Dürst)

vm_inshelper.c: improved error message for "wrong number of arguments", distinguishing given and expected argument numbers clearly. [Feature #9025]

Revision 52264 - 10/24/2015 03:47 AM - duerst (Martin Dürst)

vm_inshelper.c: improved error message for "wrong number of arguments", distinguishing given and expected argument numbers clearly. [Feature #9025]

Revision 52264 - 10/24/2015 03:47 AM - duerst (Martin Dürst)

vm_inshelper.c: improved error message for "wrong number of arguments", distinguishing given and expected argument numbers clearly. [Feature #9025]

Revision 52264 - 10/24/2015 03:47 AM - duerst (Martin Dürst)

vm_inshelper.c: improved error message for "wrong number of arguments", distinguishing given and expected argument numbers clearly. [Feature #9025]

Revision 52264 - 10/24/2015 03:47 AM - duerst (Martin Dürst)

vm_inshelper.c: improved error message for "wrong number of arguments", distinguishing given and expected argument numbers clearly. [Feature #9025]

Revision e0ed0012 - 10/24/2015 06:42 AM - nobu (Nobuyoshi Nakada)

relax error message format

- bootstraptest/test_method.rb: relax error message format.
- test/ruby/test_arity.rb (err_mess): ditto. [Feature #9025], [ruby-core:71178] [Bug #11617]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@52265 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 52265 - 10/24/2015 06:42 AM - nobu (Nobuyoshi Nakada)

relax error message format

- bootstraptest/test_method.rb: relax error message format.
- test/ruby/test_arity.rb (err_mess): ditto. [Feature #9025], [ruby-core:71178] [Bug #11617]

Revision 52265 - 10/24/2015 06:42 AM - nobu (Nobuyoshi Nakada)

relax error message format

- bootstraptest/test_method.rb: relax error message format.
- test/ruby/test_arity.rb (err_mess): ditto. [Feature #9025], [ruby-core:71178] [Bug #11617]

Revision 52265 - 10/24/2015 06:42 AM - nobu (Nobuyoshi Nakada)

relax error message format

- bootstraptest/test_method.rb: relax error message format.
- test/ruby/test_arity.rb (err_mess): ditto. [Feature #9025], [ruby-core:71178] [Bug #11617]

Revision 52265 - 10/24/2015 06:42 AM - nobu (Nobuyoshi Nakada)

relax error message format

- bootstraptest/test_method.rb: relax error message format.
- test/ruby/test_arity.rb (err_mess): ditto. [Feature #9025], [ruby-core:71178] [Bug #11617]

Revision 52265 - 10/24/2015 06:42 AM - nobu (Nobuyoshi Nakada)

relax error message format

- bootstraptest/test_method.rb: relax error message format.
- test/ruby/test_arity.rb (err_mess): ditto. [Feature #9025], [ruby-core:71178] [Bug #11617]

Revision 96ff94c9 - 10/24/2015 12:16 PM - nobu (Nobuyoshi Nakada)

relax error message format

- test/fileutils/test_fileutils.rb (test_uptodate): relax error message format. [Feature #9025], [ruby-core:71178] [Bug #11617]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@52269 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 52269 - 10/24/2015 12:16 PM - nobu (Nobuyoshi Nakada)

relax error message format

- test/fileutils/test_fileutils.rb (test_uptodate): relax error message format. [Feature #9025], [ruby-core:71178] [Bug #11617]

Revision 52269 - 10/24/2015 12:16 PM - nobu (Nobuyoshi Nakada)

relax error message format

- test/fileutils/test_fileutils.rb (test_uptodate): relax error message format. [Feature #9025], [ruby-core:71178] [Bug #11617]

Revision 52269 - 10/24/2015 12:16 PM - nobu (Nobuyoshi Nakada)

relax error message format

- test/fileutils/test_fileutils.rb (test_uptodate): relax error message format. [Feature #9025], [ruby-core:71178] [Bug #11617]

Revision 52269 - 10/24/2015 12:16 PM - nobu (Nobuyoshi Nakada)

relax error message format

- test/fileutils/test_fileutils.rb (test_uptodate): relax error message format. [Feature #9025], [ruby-core:71178] [Bug #11617]

Revision 52269 - 10/24/2015 12:16 PM - nobu (Nobuyoshi Nakada)

relax error message format

- test/fileutils/test_fileutils.rb (test_uptodate): relax error message format. [Feature #9025], [ruby-core:71178] [Bug #11617]

Revision 94bdb010 - 10/25/2015 12:39 AM - nobu (Nobuyoshi Nakada)

update document [ci skip]

- error.c (ArgumentError): [DOC] update document following [Feature #9025]
- proc.c (proc_call, proc_curry): [DOC] ditto.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@52274 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 52274 - 10/25/2015 12:39 AM - nobu (Nobuyoshi Nakada)

update document [ci skip]

- error.c (ArgumentError): [DOC] update document following [Feature #9025]
- proc.c (proc_call, proc_curry): [DOC] ditto.

Revision 52274 - 10/25/2015 12:39 AM - nobu (Nobuyoshi Nakada)

update document [ci skip]

- error.c (ArgumentError): [DOC] update document following [Feature #9025]
- proc.c (proc_call, proc_curry): [DOC] ditto.

Revision 52274 - 10/25/2015 12:39 AM - nobu (Nobuyoshi Nakada)

update document [ci skip]

- error.c (ArgumentError): [DOC] update document following [Feature #9025]
- proc.c (proc_call, proc_curry): [DOC] ditto.

Revision 52274 - 10/25/2015 12:39 AM - nobu (Nobuyoshi Nakada)

update document [ci skip]

- error.c (ArgumentError): [DOC] update document following [Feature #9025]
- proc.c (proc_call, proc_curry): [DOC] ditto.

Revision 52274 - 10/25/2015 12:39 AM - nobu (Nobuyoshi Nakada)

update document [ci skip]

- error.c (ArgumentError): [DOC] update document following [Feature #9025]
- proc.c (proc_call, proc_curry): [DOC] ditto.

Revision 898fd2ed - 10/25/2015 12:43 AM - nobu (Nobuyoshi Nakada)

use rb_check_arity [ci skip]

- ext: use `rb_check_arity` and `rb_error_arity` to raise `ArgumentError`. [Feature #9025]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@52275 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 52275 - 10/25/2015 12:43 AM - nobu (Nobuyoshi Nakada)

use `rb_check_arity` [ci skip]

- ext: use `rb_check_arity` and `rb_error_arity` to raise `ArgumentError`. [Feature #9025]

Revision 52275 - 10/25/2015 12:43 AM - nobu (Nobuyoshi Nakada)

use `rb_check_arity` [ci skip]

- ext: use `rb_check_arity` and `rb_error_arity` to raise `ArgumentError`. [Feature #9025]

Revision 52275 - 10/25/2015 12:43 AM - nobu (Nobuyoshi Nakada)

use `rb_check_arity` [ci skip]

- ext: use `rb_check_arity` and `rb_error_arity` to raise `ArgumentError`. [Feature #9025]

Revision 52275 - 10/25/2015 12:43 AM - nobu (Nobuyoshi Nakada)

use `rb_check_arity` [ci skip]

- ext: use `rb_check_arity` and `rb_error_arity` to raise `ArgumentError`. [Feature #9025]

Revision 52275 - 10/25/2015 12:43 AM - nobu (Nobuyoshi Nakada)

use `rb_check_arity` [ci skip]

- ext: use `rb_check_arity` and `rb_error_arity` to raise `ArgumentError`. [Feature #9025]

History

#1 - 10/16/2013 11:34 AM - sawa (Tsuyoshi Sawada)

"instead of" is even worse than "for". It is ambiguous: "1 argument was given instead of the expected 0" or "1 argument should be given instead of 0 that was given". "for" at least does not have that ambiguity.

#2 - 10/16/2013 11:53 AM - fuadksd (Fuad Saud)

I like the "expected" wording. Also, wouldn't help a little to print the method name? May help to identify what's wrong faster.

--

Fuad Saud

Sent with Sparrow (<http://www.sparrowmailapp.com/?sig>)

#3 - 10/16/2013 11:58 AM - phluid61 (Matthew Kerwin)

This is sort of a bike-shedding issue, and I'm always happy to throw paint at bike sheds!

In the past, in various languages and environments, I've tended to use:

"wrong number of arguments (given 1, expected 0)"

However, note that that StackOverflow question only has 9 upvotes. Other questions from the user are things like "What does the "+=" operator do in Java?", and a couple of spoon-feedy "please fix/explain this code" questions. While I agree that it would be nice to support ruby newbies with nice meaningful errors, I believe that the current message already conveys all the information required to comprehend it. Critically "wrong number of arguments" is pretty universal for programmers, even without relying on English per se, and the smallest amount of debugging will show that one of those numbers corresponds with the invocation causing the error, and therefore the other number must be the correct "number of arguments" for that method.

#4 - 10/16/2013 12:02 PM - phluid61 (Matthew Kerwin)

fuadksd (Fuad Saud) wrote:

I like the "expected" wording. Also, wouldn't help a little to print the method name? May help to identify what's wrong faster.

The first line of the error's backtrace gives the filename and line number of the bad call. If that line has so many method calls that you can't work out which one is wrong, perhaps *that* is the true cause of the error.

#5 - 10/16/2013 12:12 PM - fuadksd (Fuad Saud)

Maybe. Sometimes, if you do chain a couple of methods, you may get confused. I'm not saying there's no way you can figure out what's happening in the current situation, I just feel like the method name belongs here, as it belongs in NoMethodError messages.

#6 - 10/16/2013 01:03 PM - duerst (Martin Dürst)

phluid61 (Matthew Kerwin) wrote:

This is sort of a bike-shedding issue, and I'm always happy to throw paint at bike sheds!

In the past, in various languages and environments, I've tended to use:
"wrong number of arguments (given 1, expected 0)"

That's a good way to put it, and very close to Nerian's
ArgumentError: wrong number of arguments (expected: 1, provided: 0)
(close enough to really just be bikeshedding)

While I agree that it would be nice to support ruby newbies with nice meaningful errors, I believe that the current message already conveys all the information required to comprehend it. Critically "wrong number of arguments" is pretty universal for programmers, even without relying on English per se, and the smallest amount of debugging will show that one of those numbers corresponds with the invocation causing the error, and therefore the other number must be the correct "number of arguments" for that method.

Ruby is all about making it easier for the programmer. Even the smallest amount of debugging is too much when it can be eliminated by tweaking the error message. This is definitely finetuning, but it's not bikeshedding.

#7 - 10/16/2013 02:07 PM - phluid61 (Matthew Kerwin)

duerst (Martin Dürst) wrote:

phluid61 (Matthew Kerwin) wrote:

[...] I believe that the current message already conveys all the information required to comprehend it. [...] the smallest amount of debugging will show [...]

Ruby is all about making it easier for the programmer. Even the smallest amount of debugging is too much when it can be eliminated by tweaking the error message.

I think that statement accidentally reached a point of absurdity by using a very precise definition of "debugging" that I don't share.

My argument is that the programmer still has to visit the erroneous line of code, inspect it, and probably modify it to have the correct number of arguments. That amount of debugging is always required, and would be if the error message said "ArgumentError" or "ArgumentError: the method `frob' on a String object does not accept any arguments, however it was passed a Fixnum".

I agree that someone not familiar with more technical English writing might be confused by the "(1 for 0)" part, but only when reading it in isolation. If the programmer sees the message, they have to fix it, in which case they are already in the code and can understand the message using contextual information (e.g. how many arguments there are in the call, what the API docs say about the function's arity and semantics, etc.); if an end user sees the message, they don't have to understand it and couldn't do anything about it if they did, they just pass it on to the programmer verbatim as a bug report.

#8 - 10/22/2015 09:11 AM - robb (Robb Shecter)

"(expected: 1, provided: 0)" is excellent. This is a big usability issue. Python's message is similar, and includes the function name:

TypeError: fun() takes exactly 1 argument (0 given)

I agree that someone not familiar with more technical English writing might be confused by the "(1 for 0)" part, but only when reading it in isolation.

My experience is different. I have a computer science degree and 20 years' of experience, and I'm always frustrated by this error message. I'm never clear on which number is which. I even got it wrong a few minutes ago in a comment I left here: "Request for comments about error messages"
<https://bugs.ruby-lang.org/issues/11295>

I program in many languages, and I don't have the mental capacity to remember idiosyncracies like this. Finally, consider the Python example, above. This is how modern languages perform, and users' expectations are rightly geared to match.

Ruby's dynamic nature defeats mere static analysis which you suggest would fix this: looking at one's code, it is not always possible to know how many arguments were passed.

I'll submit a Pull Request in order to make the conversation more concrete.

#9 - 10/22/2015 09:51 PM - duerst (Martin Dürst)

Matthew Kerwin wrote:

duerst (Martin Dürst) wrote:

Ruby is all about making it easier for the programmer. Even the smallest amount of debugging is too much when it can be eliminated by tweaking the error message.

I think that statement accidentally reached a point of absurdity by using a very precise definition of "debugging" that I don't share.

It might have been clearer for me to write "Even the smallest reduction in the amount of debugging is worth it when it can be achieved by tweaking the error message."

Anyway, if it doesn't make life easier for you, that's no problem. It won't hurt you, and it will help others, even if you might not believe it.

#10 - 10/22/2015 11:48 PM - phluid61 (Matthew Kerwin)

On 23 October 2015 at 07:51, duerst@it.aoyama.ac.jp wrote:

Issue [#9025](#) has been updated by Martin Dürst.

Matthew Kerwin wrote:

duerst (Martin Dürst) wrote:

Ruby is all about making it easier for the programmer. Even the smallest amount of debugging is too much when it can be eliminated by tweaking the error message.

I think that statement accidentally reached a point of absurdity by using a very precise definition of "debugging" that I don't share.

It might have been clearer for me to write "Even the smallest reduction in the amount of debugging is worth it when it can be achieved by tweaking the error message."

Anyway, if it doesn't make life easier for you, that's no problem. It won't hurt you, and it will help others, even if you might not believe it.

That's true, and my position has mellowed in the past two years. At worst harmless*, at best an improvement.

*Does anyone depend on the precise wording of this error message? I know it's an anti-pattern, but sometimes unavoidable, especially when dealing with these core exceptions that can't easily be modified to include programatically-retrievable data.

#11 - 10/24/2015 03:48 AM - duerst (Martin Dürst)

- *Status changed from Open to Closed*

Applied in changeset r52264.

vm_inshelper.c: improved error message for "wrong number of arguments", distinguishing given and expected argument numbers clearly. [Feature [#9025](#)]

#12 - 10/24/2015 03:49 AM - duerst (Martin Dürst)

- *Related to Bug #11617: Fix tests for Feature 9025 added*