

## Ruby trunk - Bug #14324

### Should Exception#full\_message include escape sequences?

01/06/2018 10:27 AM - Eregon (Benoit Daloze)

<b>Status:</b>	Closed		
<b>Priority:</b>	Normal		
<b>Assignee:</b>	sorah (Sorah Fukumori)		
<b>Target version:</b>	2.6		
<b>ruby -v:</b>	ruby 2.5.0p0 (2017-12-25 revision 61468) [x86_64-linux]	<b>Backport:</b>	2.3: DONTNEED, 2.4: DONTNEED, 2.5: DONE
<b>Description</b>			
<pre>\$ ruby -e 'p RuntimeError.new("foo").full_message' "\e[1mTraceback \e[m(most recent call last):\n-e:1:in `full_message': \e[1mfoo (\e[4;1mRuntimeError\n\e[m\n\e[1m)\n\e[m"</pre>			
That is probably not intended.			
<pre>\$ ruby -e 'p RuntimeError.new("foo").full_message'  &amp; cat "-e:1:in `full_message': foo (RuntimeError)\n"</pre>			
So Exception#full_message depends if stderr is a tty which seems wrong. I think it should be the second result, i.e., no escape sequences.			
<b>Related issues:</b>			
Related to Ruby trunk - Feature #14141: Add a method to Exception for retriev...			<b>Closed</b>
Related to Ruby trunk - Feature #9918: Exception#cause should be shown in out...			<b>Open</b>

#### Associated revisions

##### Revision b9881083 - 03/22/2018 08:26 AM - nobu (Nobuyoshi Nakada)

error.c: full\_message options

- error.c (exc\_full\_message): add highlight: and reverse: keyword options. [Bug #14324]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@62894 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 62894 - 03/22/2018 08:26 AM - nobu (Nobuyoshi Nakada)

error.c: full\_message options

- error.c (exc\_full\_message): add highlight: and reverse: keyword options. [Bug #14324]

##### Revision 62894 - 03/22/2018 08:26 AM - nobu (Nobuyoshi Nakada)

error.c: full\_message options

- error.c (exc\_full\_message): add highlight: and reverse: keyword options. [Bug #14324]

##### Revision ed9c67c9 - 03/24/2018 10:57 AM - naruse (Yui NARUSE)

merge revision(s) 62548,62894: [Backport #14324]

```
eval_error.c: rb_error_write flags
```

```
* eval_error.c (rb_error_write): add highlight and reverse mode flags. defaulted to rb_stderr_tty_p() if Qnil.
```

```
error.c: full_message options
```

```
* error.c (exc_full_message): add highlight: and reverse: keyword options. [Bug #14324]
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby\_2\_5@62905 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 62905 - 03/24/2018 10:57 AM - naruse (Yui NARUSE)

merge revision(s) 62548,62894: [Backport #14324]

eval\_error.c: rb\_error\_write flags

```
* eval_error.c (rb_error_write): add highlight and reverse mode
  flags. defaulted to rb_stderr_tty_p() if Qnil.
```

error.c: full\_message options

```
* error.c (exc_full_message): add highlight: and reverse: keyword
  options. [Bug #14324]
```

## History

---

### #1 - 01/07/2018 05:55 AM - nobu (Nobuyoshi Nakada)

- Description updated

Since `Exception#full_message` is expected to return the message printed to `stderr`, escape sequences are intentional. But I think it's wrong that there is no way to disable them now.

[https://github.com/nobu/ruby/tree/bug/14324-full\\_message-options](https://github.com/nobu/ruby/tree/bug/14324-full_message-options)

### #2 - 01/08/2018 10:54 AM - Eregon (Benoit Daloze)

nobu (Nobuyoshi Nakada) wrote:

Since `Exception#full_message` is expected to return the message printed to `stderr`, escape sequences are intentional.

But how to know if `Exception#full_message` is later used to print to `$stderr`, a log or something else?

`Exception#backtrace` for instance is the same on all Ruby implementations, even though the top-level handler might print differently.

I think having `Exception#full_message` be the canonical (no TTY, as in 2.4) representation of message+backtrace would make sense. Reusing the new reversed MRI exception format here but only in some cases feels wrong.

I think libraries should do their own formatting if they want reversed exceptions. It would be more portable and reliable.

How should other Ruby implementations behave here?  
Print their own custom format by default? Is that good for usages of `#full_backtrace`?  
Or try to mimic the new MRI reverse+highlight backtrace format?

But I think it's wrong that there is no way to disable them now.

[https://github.com/nobu/ruby/tree/bug/14324-full\\_message-options](https://github.com/nobu/ruby/tree/bug/14324-full_message-options)

Options seems a good idea.

### #3 - 01/08/2018 10:54 AM - Eregon (Benoit Daloze)

- Related to Feature #14141: Add a method to `Exception` for retrieving formatted exception for logging purpose (`Exception#{formatted,display}`) added

### #4 - 02/20/2018 10:03 AM - nobu (Nobuyoshi Nakada)

- Related to Feature #9918: `Exception#cause` should be shown in output and `#inspect` added

### #5 - 03/22/2018 08:26 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

Applied in changeset [trunk|r62894](#).

---

error.c: full\_message options

- error.c (exc\_full\_message): add highlight: and reverse: keyword options. [Bug #14324]

### #6 - 03/22/2018 03:37 PM - nobu (Nobuyoshi Nakada)

- Backport changed from 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN to 2.3: DONTNEED, 2.4: DONTNEED, 2.5: REQUIRED

**#7 - 03/24/2018 10:57 AM - naruse (Yui NARUSE)**

- Backport changed from 2.3: *DONTNEED*, 2.4: *DONTNEED*, 2.5: *REQUIRED* to 2.3: *DONTNEED*, 2.4: *DONTNEED*, 2.5: *DONE*

ruby\_2\_5 r62905 merged revision(s) 62548,62894.