

Ruby master - Feature #12944

Change Kernel#warn to call Warning.warn

11/15/2016 08:14 PM - jeremyevans0 (Jeremy Evans)

Status:	Closed
Priority:	Normal
Assignee:	
Target version:	
Description	
Currently, Kernel#warn is basically the same as \$stderr.puts. With the new Warning.warn support in ruby 2.4, it makes sense for Kernel#warn to call Warning.warn, otherwise you will not be able to use Warning.warn to filter/process warning messages generated by Kernel#warn.	
The Kernel#warn API is different than the Warning.warn API, the attached patch tries to get similar behavior, but there are probably corner cases where the behavior is different.	
Related issues:	
Related to Ruby master - Bug #14006: 2.5.0preview1 Warning.warn SystemS... Closed	

Associated revisions

Revision 50db992a - 01/19/2017 06:25 AM - nobu (Nobuyoshi Nakada)

Change Kernel#warn to call Warning.warn

This allows Warning.warn to filter/process warning messages generated by Kernel#warn. Currently, Warning.warn can only handle messages generated by the rb_warn/rb_warning C functions.

The Kernel#warn API is different than the Warning.warn API, this tries to get similar behavior, but there are probably corner cases where the behavior is different.

This makes str_end_with_asciichar in io.c no longer static so it can be called from error.c.

[Feature #12944]

Author: Jeremy Evans code@jeremyevans.net

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@57370 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 57370 - 01/19/2017 06:25 AM - nobu (Nobuyoshi Nakada)

Change Kernel#warn to call Warning.warn

This allows Warning.warn to filter/process warning messages generated by Kernel#warn. Currently, Warning.warn can only handle messages generated by the rb_warn/rb_warning C functions.

The Kernel#warn API is different than the Warning.warn API, this tries to get similar behavior, but there are probably corner cases where the behavior is different.

This makes str_end_with_asciichar in io.c no longer static so it can be called from error.c.

[Feature #12944]

Author: Jeremy Evans code@jeremyevans.net

Revision 57370 - 01/19/2017 06:25 AM - nobu (Nobuyoshi Nakada)

Change Kernel#warn to call Warning.warn

This allows Warning.warn to filter/process warning messages generated by Kernel#warn. Currently, Warning.warn can only handle messages generated by the rb_warn/rb_warning C functions.

The Kernel#warn API is different than the Warning.warn API, this tries to get similar behavior, but there are probably corner cases

where the behavior is different.

This makes `str_end_with_asciichar` in `io.c` no longer static so it can be called from `error.c`.

[Feature #12944]

Author: Jeremy Evans code@jeremyevans.net

Revision 57370 - 01/19/2017 06:25 AM - nobu (Nobuyoshi Nakada)

Change `Kernel#warn` to call `Warning.warn`

This allows `Warning.warn` to filter/process warning messages generated by `Kernel#warn`. Currently, `Warning.warn` can only handle messages generated by the `rb_warn/rb_warning` C functions.

The `Kernel#warn` API is different than the `Warning.warn` API, this tries to get similar behavior, but there are probably corner cases where the behavior is different.

This makes `str_end_with_asciichar` in `io.c` no longer static so it can be called from `error.c`.

[Feature #12944]

Author: Jeremy Evans code@jeremyevans.net

Revision 1d4976c0 - 01/20/2017 01:02 AM - nobu (Nobuyoshi Nakada)

`error.c`: print default RS

- `error.c (rb_warn_m)`: print the default RS instead of an empty string with a newline. [Feature #12944]

git-svn-id: [svn+ssh://ci.ruby-lang.org/ruby/trunk@57377](https://ci.ruby-lang.org/ruby/trunk@57377) b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 57377 - 01/20/2017 01:02 AM - nobu (Nobuyoshi Nakada)

`error.c`: print default RS

- `error.c (rb_warn_m)`: print the default RS instead of an empty string with a newline. [Feature #12944]

Revision 57377 - 01/20/2017 01:02 AM - nobu (Nobuyoshi Nakada)

`error.c`: print default RS

- `error.c (rb_warn_m)`: print the default RS instead of an empty string with a newline. [Feature #12944]

Revision 57377 - 01/20/2017 01:02 AM - nobu (Nobuyoshi Nakada)

`error.c`: print default RS

- `error.c (rb_warn_m)`: print the default RS instead of an empty string with a newline. [Feature #12944]

Revision ec7833ef - 04/26/2017 08:11 AM - nobu (Nobuyoshi Nakada)

`error.c`: splat warn arguments

- `error.c (rb_warn_m)`: print array arguments with splatting. it is often used with caller. [ruby-core:80849] [Feature #12944]

git-svn-id: [svn+ssh://ci.ruby-lang.org/ruby/trunk@58487](https://ci.ruby-lang.org/ruby/trunk@58487) b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 58487 - 04/26/2017 08:11 AM - nobu (Nobuyoshi Nakada)

`error.c`: splat warn arguments

- `error.c (rb_warn_m)`: print array arguments with splatting. it is often used with caller. [ruby-core:80849] [Feature #12944]

Revision 58487 - 04/26/2017 08:11 AM - nobu (Nobuyoshi Nakada)

`error.c`: splat warn arguments

- `error.c (rb_warn_m)`: print array arguments with splatting. it is often used with caller. [ruby-core:80849] [Feature #12944]

Revision 58487 - 04/26/2017 08:11 AM - nobu (Nobuyoshi Nakada)

`error.c`: splat warn arguments

- `error.c (rb_warn_m)`: print array arguments with splatting. it is often used with caller. [ruby-core:80849] [Feature #12944]

Revision d249b047 - 04/26/2017 08:13 PM - nobu (Nobuyoshi Nakada)

`error.c`: send as a single string

- `error.c (rb_warn_m)`: send the arguments as a single string concatenated with a newline, so it can be filtered easily. [ruby-core:80875] [Feature #12944]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@58490 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 58490 - 04/26/2017 08:13 PM - nobu (Nobuyoshi Nakada)

`error.c`: send as a single string

- `error.c (rb_warn_m)`: send the arguments as a single string concatenated with a newline, so it can be filtered easily. [ruby-core:80875] [Feature #12944]

Revision 58490 - 04/26/2017 08:13 PM - nobu (Nobuyoshi Nakada)

`error.c`: send as a single string

- `error.c (rb_warn_m)`: send the arguments as a single string concatenated with a newline, so it can be filtered easily. [ruby-core:80875] [Feature #12944]

Revision 58490 - 04/26/2017 08:13 PM - nobu (Nobuyoshi Nakada)

`error.c`: send as a single string

- `error.c (rb_warn_m)`: send the arguments as a single string concatenated with a newline, so it can be filtered easily. [ruby-core:80875] [Feature #12944]

History

#1 - 11/25/2016 09:32 AM - matz (Yukihiro Matsumoto)

Accepted.

Matz.

#2 - 11/25/2016 01:30 PM - nobu (Nobuyoshi Nakada)

`rb_str_cat` modifies the argument, it is unexpected to `Kernel#warn`, I think.
Also it doesn't work on a frozen string.

#3 - 11/26/2016 03:57 AM - jeremyevans0 (Jeremy Evans)

- *File 0001-Change-Kernel-warn-to-call-Warning.warn.patch added*

Here's an updated patch that uses `rb_str_dup` before calling `rb_str_cat`, so it doesn't modify a passed in string, and works with frozen strings.

#4 - 11/26/2016 05:23 AM - nobu (Nobuyoshi Nakada)

You can't compare and append the last byte simply in wide character encodings.
E.g., `str_end_with_asciichar` in `io.c`.

#5 - 11/26/2016 07:47 AM - jeremyevans0 (Jeremy Evans)

- *File 0001-Change-Kernel-warn-to-call-Warning.warn.patch added*

Here's an updated patch that uses `str_end_with_asciichar` instead of just manually checking the last byte for `\n`.

#6 - 12/12/2016 03:24 PM - jeremyevans0 (Jeremy Evans)

This didn't make 2.4.0rc1, but as matz has already accepted it, it should probably be committed before 2.4.0 final. If further changes are needed to the patch, please let me know.

#7 - 12/13/2016 05:08 AM - naruse (Yui NARUSE)

Jeremy Evans wrote:

This didn't make 2.4.0rc1, but as matz has already accepted it, it should probably be committed before 2.4.0 final. If further changes are needed to the patch, please let me know.

OK.
nobu, could you handle this?

#8 - 12/13/2016 05:10 AM - naruse (Yui NARUSE)

- Target version set to 2.4

#9 - 12/13/2016 05:34 AM - nobu (Nobuyoshi Nakada)

One thing to be confirmed, this patch lets warn ["foo", "bar"] print one line while two lines are printed now.
Is this OK?

#10 - 12/13/2016 04:27 PM - jeremyevans0 (Jeremy Evans)

- File 0001-Change-Kernel-warn-to-call-Warning.warn.patch added

The attached patch uses rb_ary_join to create the string to pass to Warning.warn, which should do a better job of handling corner cases such as warn ["foo", "bar"].

#11 - 12/14/2016 07:56 AM - nobu (Nobuyoshi Nakada)

Seems warn ["foo\n", "bar"] would print "foo\n\nbar\n".

#12 - 12/14/2016 03:42 PM - jeremyevans0 (Jeremy Evans)

- File 0001-Change-Kernel-warn-to-call-Warning.warn.patch added

In order to handle embedded arrays (similar to how puts handles them), the attached patch flattens the array, processes each string individually making sure it ends in a newline, then joins the array with the empty string.

#13 - 12/19/2016 05:10 PM - jeremyevans0 (Jeremy Evans)

Since the ruby_2_4 branch will be created on Friday, I think this patch needs to be merged by then. If additional changes are needed, please let me know.

#14 - 12/20/2016 01:10 AM - hsbt (Hiroshi SHIBATA)

We will discuss incompatible changes and feature details at <https://bugs.ruby-lang.org/projects/ruby/wiki/DevelopersMeeting20161221Japan> with Matz

#15 - 12/21/2016 01:41 PM - shyouhei (Shyouhei Urabe)

So we discussed at today's developer meeting. Attendees felt in common that Kernel#warn's complicated pitfall-ish behaviour is not useful very much. Does anybody actually pass array to it? Warning.warn is much straight-forward. It makes relatively less sense to emulate Kernel#warn's old corner cases using Warning.warn.

We want to re-think Kernel#warn API. In order to do so 2.4 is too immediate. Let us leave it as is now. Don't get me wrong; I basically think this feature is a good one to have. Just a bit bigger than it was thought to be.

#16 - 12/25/2016 08:27 PM - naruse (Yui NARUSE)

- Target version deleted (2.4)

#17 - 01/19/2017 05:31 AM - matz (Yukihiro Matsumoto)

Go ahead. I want to check compatibility issues early.

Matz.

#18 - 01/19/2017 06:25 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

Applied in changeset r57370.

Change Kernel#warn to call Warning.warn

This allows Warning.warn to filter/process warning messages generated by Kernel#warn. Currently, Warning.warn can only handle messages generated by the rb_warn/rb_warning C functions.

The Kernel#warn API is different than the Warning.warn API, this tries to get similar behavior, but there are probably corner cases

where the behavior is different.

This makes `str_end_with_asciichar` in `io.c` no longer static so it can be called from `error.c`.

[Feature [#12944](#)]

Author: Jeremy Evans code@jeremyevans.net

#19 - 04/24/2017 08:40 PM - Eregon (Benoit Daloze)

It seems it is a compatibility issue.

For instance `Thread.exclusive` warning is now printed incorrectly:

```
$ ruby -ve 'def m; Thread.exclusive {}; end; m'
ruby 2.4.0p0 (2016-12-24 revision 57164) [x86_64-linux]
Thread.exclusive is deprecated, use Thread::Mutex
-e:1:in `m'
-e:1:in `'
```

```
$ ruby -ve 'def m; Thread.exclusive {}; end; m'
ruby 2.5.0dev (2017-04-24 trunk 58466) [x86_64-linux]
Thread.exclusive is deprecated, use Thread::Mutex
["-e:1:in `m'", "-e:1:in `'"]
```

The fix is easy:

```
def self.exclusive
- warn "Thread.exclusive is deprecated, use Thread::Mutex", caller
+ warn "Thread.exclusive is deprecated, use Thread::Mutex", *caller
  MUTEX_FOR_THREAD_EXCLUSIVE.synchronize{
```

But this is not the only usage of `warn + caller`.

<https://github.com/search?l=Ruby&p=2&q=warn+caller&type=Code&utf8=%E2%9C%93>

I think it would be best to remain compatible and `join("\n")` `#warn`'s arguments.

#20 - 04/24/2017 08:49 PM - Eregon (Benoit Daloze)

Also, similar to [#13505](#), I think a single warning String (containing newlines) should be sent to `Warning.warn`.

Otherwise, `warn "my warning", caller` will appear as two different calls to `Warning.warn` and the second call cannot be filtered reliably without the necessary "my warning" context.

The original code also did a single `rb_io_puts` call so I think it makes most sense to make a single call to `Warning.warn` with the joined arguments.

#21 - 04/25/2017 12:10 PM - Eregon (Benoit Daloze)

- *File 0001-Restore-behavior-of-Kernel-warn-accepting-arrays-arg.patch added*

I attach a patch based on Jeremy "Use `rb_ary_join`" patch, rebased to current trunk.

I think using `args.join("\n")` is a good way to restore the needed compatibility as shown above and yet simple enough.

It's also what I intuitively used when redefining `Kernel#warn` in Ruby.

It fixes the wrong output for the `Thread.exclusive` warning, allows `ruby/spec` to no longer have different tests for 2.5's `#warn` and sends a single String to `Warning.warn` for proper filtering.

Can I commit this patch?

#22 - 04/26/2017 01:03 AM - nobu (Nobuyoshi Nakada)

Nitpicking an edge case, these two are different.

```
puts "foo", "" #=> "foo\n" and "\n"
```

```
puts ["foo", ""].join("\n") #=> "foo\n"
```

#23 - 04/26/2017 12:44 PM - Eregon (Benoit Daloze)

nobu (Nobuyoshi Nakada) wrote:

Nitpicking an edge case, these two are different.

I see, do you think it is important?

I think the second output is actually more natural for a warning.

`r58487` is not enough, it calls `Warning#write` and not `Warning.warn`:

```
ruby -ve 'def Warning.warn(msg); p msg; end; def m; Thread.exclusive {}; end; m'
```

should send to `Warning.warn` so it can be filtered.
I also think `Warning#write` should not be defined.

The entire warning should also be sent as a single String, so for instance I can filter based on where the warning was emitted:

```
def Warning.warn(message)
  case message
  when /Thread\.exclusive is deprecated.+\\n.+thread\\/exclusive_spec\\.rb/
    # ignore
  else
    super(message)
  end
end
```

#24 - 04/27/2017 11:24 AM - Eregon (Benoit Daloze)

Thank you nobu for fixing this, r58490 solves all my concerns for `Kernel#warn`.

#25 - 10/12/2017 04:57 AM - wanabe (_ wanabe)

- Related to Bug #14006: 2.5.0preview1 `Warning.warn` `SystemStackError` added

Files

0001-Change-Kernel-warn-to-call-Warning.warn.patch	1.88 KB	11/15/2016	jeremyevans0 (Jeremy Evans)
0001-Change-Kernel-warn-to-call-Warning.warn.patch	1.89 KB	11/26/2016	jeremyevans0 (Jeremy Evans)
0001-Change-Kernel-warn-to-call-Warning.warn.patch	2.36 KB	11/26/2016	jeremyevans0 (Jeremy Evans)
0001-Change-Kernel-warn-to-call-Warning.warn.patch	2.3 KB	12/13/2016	jeremyevans0 (Jeremy Evans)
0001-Change-Kernel-warn-to-call-Warning.warn.patch	3.23 KB	12/14/2016	jeremyevans0 (Jeremy Evans)
0001-Restore-behavior-of-Kernel-warn-accepting-arrays-arg.patch	1.23 KB	04/25/2017	Eregon (Benoit Daloze)