

## Ruby master - Feature #12026

### Support warning processor

01/26/2016 10:13 PM - jeremyevans0 (Jeremy Evans)

|  |        |               |
|--|--------|---------------|
| <b>Status:</b>   | Closed |               |
| <b>Priority:</b>   | Normal |               |
| <b>Assignee:</b>   |        |               |
| <b>Target version:</b>   |        |               |
| <b>Description</b>   |        |               |
| <p>This adds a simple way to filter warnings from being displayed. You just set a <code>\$WARNING_FILTER</code> with a regexp, and any warnings that match the regexp will not be displayed.</p> <p>I think this is a much simpler approach to filtering warnings than feature <a href="#">#11588</a>, while allowing the same type of capability.</p> <p>This is backwards compatible, in that you can set <code>\$WARNING_FILTER</code> in previous versions of ruby without problems.</p> <p>This should not cause any performance issues, as the regexp matching isn't done until the warning message is about to be displayed.</p> <p>It's possible to use something other than a global variable to store the warning filter, but a global variable was the easiest way to implement it, it has a global effect, and similar flags such as <code>\$VERBOSE</code> are also global variables, so I think a global variable makes sense.</p> |        |               |
| <b>Related issues:</b>   |        |               |
| Related to Ruby master - Feature #11588: Implement structured warnings   |        | <b>Open</b>   |
| Related to Ruby master - Feature #17122: Add category to Warning#warn  |        | <b>Closed</b> |

### History

#### #1 - 01/27/2016 03:58 PM - Eregon (Benoit Daloze)

I think this filter should be "one for the whole execution" and therefore something like an environment variable or a CLI flag.

Taking inspiration from javac's -Xlint: [1], having them associated by a group or short name would help to make it more effective across implementations, and maybe distinct mostly harmless warnings from more serious ones which happen to also match that filter.

[1] Scroll above <http://docs.oracle.com/javase/7/docs/technotes/tools/windows/javac.html#commandlineargfile>.

#### #2 - 03/14/2016 07:03 PM - jeremyevans0 (Jeremy Evans)

Since this will be discussed tomorrow at the developers meeting, here's a more detailed example of how this can be used.

Let's say you are using a library that is not free of verbose warnings, but you would like to use verbose warnings in your application. This is currently a pain as enabling warnings in your application will also print warnings for all libraries.

With this feature, you can easily filter specific types of warnings for specific libraries, while still getting the benefits of verbose warnings for your application code. Example:

```
$WARNING_FILTER = %r{/gem_name(-[\d\.]*)?/lib/.*.rb:\d+: warning: instance variable \@w+ not initialized\Z}
```

Having the filter be a global variable as opposed to an environment variable or CLI flag allows for additional flexibility, such as modifying the filter as runtime, which could be used for a block based filtering approach:

```
def filter_warnings(filter)
  prev = $WARNING_FILTER
  $WARNING_FILTER = filter
  yield
ensure
  $WARNING_FILTER = prev
end
```

Such an approach is not currently threadsafe, though I think \$WARNING\_FILTER could be made per-thread.

The attached patch is the minimally invasive way I could think of to support warning filters. While using regexps for filtering warnings has some issues in regards to warning text changing, the text of ruby's warnings does not change frequently and it is possible to handle multiple filters and variations easily using Regexp.union. So I don't expect the use of regexps for filtering warnings would be a problem in practice.

### #3 - 03/17/2016 06:51 AM - shyouhei (Shyouhei Urabe)

Yes we discussed this at the meeting, and someone pointed out that this request has something to do with <https://bugs.ruby-lang.org/issues/11588>. Everyone wants to control which warning to show and which to avoid. Proposed ways vary from a heavyweight complex mechanism to a lightweight grep-like filtering.

Maybe ruby should provide a primitive to cook warning strings before they are displayed. That way you can write your filter on top of it, or parse JSON-formatted warning string to operate, and so on.

### #4 - 03/17/2016 06:51 AM - shyouhei (Shyouhei Urabe)

- Related to Feature #11588: Implement structured warnings added

### #5 - 03/17/2016 02:28 PM - jeremyevans0 (Jeremy Evans)

Shyouhei Urabe wrote:

Yes we discussed this at the meeting, and someone pointed out that this request has something to do with <https://bugs.ruby-lang.org/issues/11588>. Everyone wants to control which warning to show and which to avoid. Proposed ways vary from a heavyweight complex mechanism to a lightweight grep-like filtering.

Maybe ruby should provide a primitive to cook warning strings before they are displayed. That way you can write your filter on top of it, or parse JSON-formatted warning string to operate, and so on.

One simple way to do this would be to add something like \$WARNING\_PROCESSOR, which would be any callable object, and if present it is called with all warnings and the result is displayed (or nothing is displayed if nil is returned). Here's an example if you just wanted a filter:

```
$WARNING_PROCESSOR = proc do |warning|
  unless warning =~ %r{/gem_name(-[\d\.]+)?/lib/.*.rb:\d+: warning: instance variable \@w+ not initialized\Z}
    warning
  end
end
```

This would make it simple to do warning message deduping using a hash (as requested by naruse):

```
seen_warnings = {}
$WARNING_PROCESSOR = proc do |warning|
  unless seen_warnings[warning]
    seen_warnings[warning] = true
    warning
  end
end
```

As well as allow arbitrary code to handle warnings, which should fix ko1's objection and handle your example of JSON output:

```
$WARNING_PROCESSOR = proc do |warning|
  if warning =~ %r{/A(.+\.rb):(\d+): warning: (.+)\Z}
    {'file'=>$1, 'line'=>$2, 'warning'=>$3}.to_json
  else
    {'warning_line'=>warning}.to_json
  end
end
```

I can work on an implementation of this if the response is positive.

### #6 - 03/17/2016 04:29 PM - Eregon (Benoit Daloze)

I think writing the warning to the screen should be part of the customizable behavior of that "warning processor". Therefore analyzing the return value is no longer necessary and it becomes more natural to redirect warnings to a file for example.

### #7 - 03/17/2016 04:40 PM - jeremyevans0 (Jeremy Evans)

Benoit Daloze wrote:

I think writing the warning to the screen should be part of the customizable behavior of that "warning processor". Therefore analyzing the return value is no longer necessary and it becomes more natural to redirect warnings to a file for example.

I like this idea. So this would change the filtering example to:

```
$WARNING_PROCESSOR = proc do |warning|
  unless warning =~ %r{/gem_name(-[\d\.]+)?/lib/.+.rb:\d+: warning: instance variable `@\w+ not initialized\Z}
    $stderr.puts warning
  end
end
```

#### #8 - 03/21/2016 05:20 PM - jeremyevans0 (Jeremy Evans)

Tsuyoshi Sawada wrote:

Can we have additional syntax that is analogous to rescue (but a little different)? Maybe something that takes a regex (instead of exception class) for matching, and an optional => that assigns message (and possibly backtrace):

```
begin
  some_routine
  catch_warning /some regex pattern to match a warning/ => message, backtrace
  # do whatever with message, backtrace
end
# continues
```

Note that, unlike rescue, catching a warning should not affect the code flow following the block; it only absorbs the warning message.

If you would like ruby to support that, please open a new feature request for it. I don't want to derail the discussion of this feature request with your proposal.

#### #9 - 04/09/2016 10:28 PM - jeremyevans0 (Jeremy Evans)

- File 0001-Add-WARNING\_PROCESSOR-for-processing-warnings.patch added

- Subject changed from Support warning filters to Support warning processor

Here is a patch that implements \$WARNING\_PROCESSOR support.

#### #10 - 04/13/2016 05:53 AM - matz (Yukihiro Matsumoto)

- Status changed from Open to Closed

Hi,

- I agree with the need for filtering/customizing warnings
- but I disagree with the use of a global variable
- casting warnings via warn method may work
- but there might be need for controlling warnings per gem/class basis

In summary, we cannot accept this proposal as it is.

Matz.

#### #11 - 08/14/2020 02:39 AM - shyouhei (Shyouhei Urabe)

- Related to Feature #17122: Add category to Warning#warn added

#### Files

|  |         |            |                             |
|--|---------|------------|-----------------------------|
| 0001-Support-warning-filters.patch                       | 4.12 KB | 01/26/2016 | jeremyevans0 (Jeremy Evans) |
| 0001-Add-WARNING_PROCESSOR-for-processing-warnings.patch | 5.13 KB | 04/09/2016 | jeremyevans0 (Jeremy Evans) |