Deprecate $SAFE support in ERB and let ERB.new take keyword arguments for it


Proc-level $SAFE change does no longer make sense. This should be deprecated and then removed. But the problem is that the position of safe_level argument is in the middle of argument list of ERB#initialize.

So to remove it safely, we should make them keyword arguments and deprecate the current non-keyword-argument interface. Having both interfaces would be backward-compatible because all of arguments don't take {}.

**before**

```erb
ERB.new("<%= 'foo' %>", 1, '%', '_erbout')
```

**after**

```erb
ERB.new("<%= 'foo' %>", 1, '%', '_erbout') # print deprecation in 2.6 if argc >= 2, and removed in later version
ERB.new("<%= 'foo' %>", trim_mode: '%', eoutvar: '_erbout') # no safe_level support from first
```

**Related issues:**

Related to Ruby master - Feature #14250: Make `$SAFE` process global state an...

**Associated revisions**

Revision cc777d09 - 02/22/2018 01:28 PM - k0kubun (Takashi Kokubun)

erb.rb: deprecate safe_level of ERB.new

Also, as it's in the middle of the list of 4 arguments, 3rd and 4th arguments (trim_mode, eoutvar) are changed to keyword arguments.

Old ways to specify arguments are deprecated and warned now.

bin/erb: deprecate -S option.

We'll remove all of deprecated ones at Ruby 2.7+

enc/make_encmake.rb: stopped using deprecated interface

ext/etc/mkconstants.rb: ditto

ext/socket/mkconstants.rb: ditto

sample/ripper/ruby2html.rb: ditto

spec/ruby/library/erb/defmethod/def_erb_method_spec.rb: ditto

spec/ruby/library/erb/new_spec.rb: ditto

test/erb/test_erb.rb: ditto
test/erb/test_erb_command.rb: ditto
tool/generic.erb.rb: ditto
tool/ruby_vm/helpers/dumper.rb: ditto
tool/transcode-ibigen.rb: ditto

lib/rdoc/erbio.rb: ditto

lib/rdoc/generator/darkfish.rb: ditto

[Feature #14256]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@62529 b2dd03c8-39d4-4d8f-98f8-823fe69b080e

Revision 62529 - 02/22/2018 01:28 PM - k0kubun (Takashi Kokubun)

erb.rb: deprecate safe_level of ERB.new

Also, as it's in the middle of the list of 4 arguments, 3rd and 4th arguments
(trim_mode, eoutvar) are changed to keyword arguments.
Old ways to specify arguments are deprecated and warned now.

bin/erb: deprecate -S option.

We'll remove all of deprecated ones at Ruby 2.7+.

enc/make_enencmake.rb: stopped using deprecated interface
test/encmconstants.rb: ditto
test/socket/mkconstants.rb: ditto
libdoc/erbio.rb: ditto
libdoc/generator/darkfish.rb: ditto

[Feature #14256]
Revision 62529 - 02/22/2018 01:28 PM - k0kubun (Takashi Kokubun)
erb.rb: deprecate safe_level of ERB.new

Also, as it's in the middle of the list of 4 arguments, 3rd and 4th arguments
(trim_mode, eoutvar) are changed to keyword arguments.
Old ways to specify arguments are deprecated and warned now.

bin/erb: deprecate -S option.

We'll remove all of deprecated ones at Ruby 2.7+.

enc/make_enencmake.rb: stopped using deprecated interface
test/encmconstants.rb: ditto
test/socket/mkconstants.rb: ditto
libdoc/erbio.rb: ditto
libdoc/generator/darkfish.rb: ditto

[Feature #14256]
Revision d12a08ab - 01/15/2022 09:46 PM - k0kubun (Takashi Kokubun)
[ruby/erb] Remove safe_level and further positional arguments (https://github.com/ruby/erb/pull/7)

[Feature #14256]
https://github.com/ruby/erb/commit/5133efa06f

History
#1 - 12/28/2017 05:52 AM - k0kubun (Takashi Kokubun)
- Subject changed from Deprecate $SAFE support in ERB to Deprecate $SAFE support in ERB and let ERB.new take keyword arguments for it

#2 - 12/28/2017 05:54 AM - k0kubun (Takashi Kokubun)
- Related to Feature #14250: Make "$SAFE" process global state and allow to set 0 again added

#3 - 02/22/2018 01:28 PM - k0kubun (Takashi Kokubun)
- Status changed from Open to Closed

Applied in changeset trunk/r62529.
erb.rb: deprecate safe_level of ERB.new

Also, as it's in the middle of the list of 4 arguments, 3rd and 4th arguments
(trim_mode, eoutvar) are changed to keyword arguments.
Old ways to specify arguments are deprecated and warned now.

bin/erb: deprecate -S option.

We'll remove all of deprecated ones at Ruby 2.7+.

enc/make_encmake.rb: stopped using deprecated interface
ext/etc/mkconstants.rb: ditto
ext/socket/mkconstants.rb: ditto
sample/ripper/ruby2html.rb: ditto
spec/ruby/library/erb/defmethod/def_erb_method_spec.rb: ditto
spec/ruby/library/erb/new_spec.rb: ditto
test/erb/test_erb.rb: ditto
test/erb/test_erb_command.rb: ditto
tool/generic_erb.rb: ditto
tool/ruby_vm/helpers/dumper.rb: ditto
tool/transcode-tblgen.rb: ditto
lib/rdoc/erbio.rb: ditto
lib/rdoc/generator/darkfish.rb: ditto

[Feature #14256]

#4 - 02/28/2018 12:21 PM - k0kubun (Takashi Kokubun)

I forgot to have a link to this ticket in r62612.

Ruby 2.5's ERB.new interface will continue to be available until Ruby 2.5 becomes EOL. So you don't need to branch ERB.new usage in many places. The deprecation warning is shown only when "-w" (or "-W2") flag is explicitly added.

If you're even against having deprecation warning on warn level 2, please add comments here. I may postpone it.