Redmine - Feature #26561

Enable frozen string literals

2017-07-27 22:48 - Pavel Rosický

Status:	Closed		**Start date:**	
Priority:	Normal		**Due date:**	
Assignee:	Go MAEDA		**% Done:**	0%
Category:	Performance		**Estimated time:**	0.00 hour
Target version:	4.1.0			
Resolution:				

Description

Hi, Ruby 2.3 introduced support for frozen string literals

Pros

- less memory consumption
- performance (less GC preasure)
- safer code

Cons

- mutable functions like force_encoding or gsub! can't be called on frozen strings anymore, this may cause incompatibility with some plugins, but it's very easy to fix and still maintain the backwards compatibility
- a magic comment has to be added at the top of each rb file :-(

More information
https://www.pluralsight.com/blog/software-development/ruby-2-3--working-with-immutable-strings-

Many popular gems and libraries already did this - for example:
https://github.com/rails/rails/pull/29655
https://github.com/mikel/mail/pull/970

I attached a patch based on master that passes all tests.
There're some garbage changes like removed whitespaces, I'll clean it up if it's approved.

What do you think?

Related issues:

Related to Redmine - Patch #31004: Decode hexadecimal-encoded literals in ord...	**Closed**
Related to Redmine - Defect #31052: Download unified diff throws "can't modif...	**Closed**
Related to Redmine - Patch #31059: Use #b shortcut instead of #force_encoding	**Closed**
Related to Redmine - Defect #32071: FrozenError in assert_save in test/test_h...	**Closed**
Related to Redmine - Defect #31508: Add missing frozen strings and copyrights	**Closed**
Related to Redmine - Defect #33021: [v.4.0.5-stable] Internal Server Error 50...	**Closed**

Associated revisions

Revision 17947 - 2019-03-15 02:32 - Go MAEDA

Add "frozen_string_literal: false" for all files (#26561).

This will be changed to true in the future.

Revision 17974 - 2019-03-16 03:20 - Go MAEDA

Fix that methods in Redmine::CodesetUtil destroy the original string (#26561).

Revision 17975 - 2019-03-16 05:43 - Go MAEDA

Fix RDMwriteFormattedCell destroying an argument (#26561).

Patch by Pavel Rosický.

Revision 17977 - 2019-03-16 10:37 - Go MAEDA

Enable frozen_string_literal for some files under app and lib directory (#26561).

Revision 17978 - 2019-03-16 16:03 - Go MAEDA

Enable frozen_string_literal for some files under test directory (#26561).

Revision 17979 - 2019-03-17 01:23 - Go MAEDA

Enable frozen_string_literal for app/views/*/*.builder (#26561).

Revision 17980 - 2019-03-17 01:29 - Go MAEDA

Support frozen_string_literal in issue, journal, and query (#26561).

Patch by Pavel Rosický.

Revision 17981 - 2019-03-17 02:47 - Go MAEDA

Enable frozen_string_literal for files which don't contain string literals (#26561).

Revision 17982 - 2019-03-17 03:04 - Go MAEDA

Enable frozen_string_literal for all files under config directory (#26561).

Revision 17983 - 2019-03-17 04:29 - Go MAEDA

Support frozen_string_literal in config/initializers/10-patches.rb (#26561).

Patch by Pavel Rosický.

Revision 17985 - 2019-03-17 07:10 - Go MAEDA

Support frozen_string_literal in lib/redmine/export/*.rb (#26561).

Contributed by Pavel Rosický.

Revision 17987 - 2019-03-17 14:36 - Go MAEDA

Support frozen_string_literal in app/**/*.rb (#26561).

Contributed by Pavel Rosický.

Revision 17988 - 2019-03-17 17:36 - Go MAEDA

Support frozen_string_literal in lib/**/*.rb (#26561).

Contributed by Pavel Rosický.

Revision 17990 - 2019-03-19 03:42 - Go MAEDA

Fix FrozenError in RepositoriesController#diff (#31052, #26561).

Patch by Mizuki ISHIKAWA.

Revision 17994 - 2019-03-20 16:02 - Go MAEDA

Fix FrozenError in BazaarAdapter.branch_conf_path (#26561).

Patch by Go MAEDA.

Revision 17995 - 2019-03-21 01:45 - Go MAEDA

Remove extra #dup (#26561).

Revision 17996 - 2019-03-21 04:27 - Go MAEDA

Support frozen_string_literal in test/**/*.rb (#26561).

Contributed by Pavel Rosický.

Revision 17997 - 2019-03-21 07:06 - Go MAEDA

Remove frozen_string_literal magic comment from files which are not used when running the application or tests (#23630).

Revision 18474 - 2019-09-16 11:23 - Go MAEDA

Fix FrozenError during tests (#32071, #26561).

Patch by Go MAEDA.

History

#1 - 2017-08-18 19:56 - Toshi MARUYAMA

- Description updated

#2 - 2017-08-18 20:11 - Toshi MARUYAMA

- Target version set to 4.1.0

#3 - 2019-02-05 23:39 - Marius BĂLTEANU

I've added your patch to my [GitLab repository](#) in order to run the tests and to see what is missing.

Do you see any value in adding the # frozen_string_literal: true to db migration files?

#4 - 2019-02-06 00:31 - Pavel Rosický

Hi Marius,
the patch is more than a year old, so there'll be some conflicts and missing files.

Do you see any value in adding the # frozen_string_literal: true to db migration files?
There was a discussion about making it default for Ruby 3, so it was historically enforced everywhere by formatters like rubocop
https://github.com/rubocop-hq/rubocop . But in terms of performance or memory usage there probably no point for adding the comment to migration files.

one more thing, if Ruby 2.2 support is dropped [#30356](#)
instead of "".dup we can use +"" . That wasn't an option a year ago when even ruby 1.x was still supported.

should I update the patch and retest it on the recent redmine version?

#5 - 2019-02-06 07:23 - Marius BĂLTEANU

Pavel Rosický wrote:

> Hi Marius,
>
> one more thing, if Ruby 2.2 support is dropped [#30356](#)
> instead of "".dup we can use +"" . That wasn't an option a year ago when even ruby 1.x was still supported.
>
> should I update the patch and retest it on the recent redmine version?

Yes, if you can do it, it will be very useful and I think we should have in mind that the support for Ruby 2.2 will be dropped in 4.1.

I'm in favour of this change, but this doesn't mean necessary that the patch will be committed. Most probably, it will be Jean-Philippe decision, but considering that almost all gems use frozen_string_literal, I don't see a real reason for not including it.

#6 - 2019-02-06 21:49 - Pavel Rosický

ok, rebased and pushed to the github
https://github.com/redmine/redmine/pull/113

there're many changes but most of them is just adding a magic comment

#7 - 2019-03-06 14:22 - Pavel Rosický

since ruby 2.2 support was dropped this should be ready for 4.1

let me know if there's anything else I can do

#8 - 2019-03-06 18:09 - Go MAEDA

Since the patch is too big to review for me, I would like to take a step-by-step approach. It means that I will enable frozen_string_literal in each file one after another.

Today, I confirmed that we can enable frozen_string_literal for test/unit/activity_test.rb by just adding a magic comment and for test/unit/attachment_test.rb by applying Pavel Rosický's patch.

#9 - 2019-03-06 20:29 - Pavel Rosický

- File frozen.patch added

https://github.com/redmine/redmine/pull/115
this is a pull request without frozen_string_literal comment - 69 vs 936 changes, it should be much easier to review

you can apply the comment afterwards using rubocop like this
> rubocop --only Style/FrozenStringLiteralComment --auto-correct -c ruboconfig.yml

ruboconfig.yml

```
AllCops:
  TargetRubyVersion: 2.3
```

#10 - 2019-03-07 04:04 - Go MAEDA

Pavel Rosický wrote:

> https://github.com/redmine/redmine/pull/115
> this is a pull request without frozen_string_literal comment - 69 vs 936 changes, it should be much easier to review

Thank you for updating the patch. I would like to enable frozen_string_literal in files under the ./test directory first for the following reasons:

- Changes don't affect production environments
- We can ensure that all code is executed with frozen_string_literals enabled by simply running bin/rails test

After that, I want to apply the rest of your patch.

Comments welcome.

#11 - 2019-03-07 21:37 - Pavel Rosický

Go MAEDA wrote:

> Thank you for updating the patch. I would like to enable frozen_string_literal in files under the ./test directory first

I can do that, but there're some problematic methods
User#find_by_login
https://github.com/redmine/redmine/blob/33522bf2e88026d0ca4f3d304d016dddf5966044/app/models/user.rb#L489
Redmine::FieldFormat#parse_keyword
https://github.com/redmine/redmine/blob/4770fa5b6a12f78a5c055efc8a7bf47555496426/lib/redmine/field_format.rb#L200
and the whole codeset_utils

they're are used heavily in tests

consider this example

```
login = 'test'.force_encoding('ascii-8bit')
puts "before: #{login.encoding}" # => ascii-8bit
user = User.find_by_login(login)
puts "after: #{login.encoding}" # => utf-8
```

or in my case

```
login = 'test'.freeze
User.find_by_login(login)
=> FrozenError: can't modify frozen String
```

What is the prefered way how to deal with these issues? I would rather fix those methods to avoid side-effects, what do you think?

#12 - 2019-03-08 00:58 - Go MAEDA

Pavel Rosický wrote:

> What is the prefered way how to deal with these issues? I would rather fix those methods to avoid side-effects, what do you think?

You don't have to work on splitting the patch. I will do that :)

#13 - 2019-03-08 01:12 - Pavel Rosický

great, thanks :)

#14 - 2019-03-10 14:45 - Go MAEDA

- Related to Patch #31004: Decode hexadecimal-encoded literals in order to be frozen string literals friendly added

#15 - 2019-03-19 03:37 - Go MAEDA

- Related to Defect #31052: Download unified diff throws "can't modify frozen String" error. added

#16 - 2019-03-20 00:24 - Go MAEDA

- Related to Patch #31059: Use #b shortcut instead of #force_encoding added

#17 - 2019-03-21 07:23 - Go MAEDA

Thanks to Pavel Rosický, I have enabled frozen_string_literal for almost all files used when running the application or tests.

But I have not enabled frozen_string_literal for some files yet for the following reasons:

- lib/redmine/imap.rb and lib/redmine/pop3.rb are not covered by existing tests
- I don't have environments to test Redmine.pm or OpenId

Here is a list of files for which frozen_string_literal is still "false":

- extra/svn/reposman.rb
- extra/mail_handler/rdm-mailhandler.rb
- lib/redmine/imap.rb
- lib/redmine/pop3.rb
- lib/plugins/open_id_authentication/init.rb
- lib/plugins/open_id_authentication/lib/open_id_authentication/mem_cache_store.rb
- lib/plugins/open_id_authentication/lib/open_id_authentication/request.rb
- lib/plugins/open_id_authentication/lib/open_id_authentication/db_store.rb
- test/extra/redmine_pm/repository_git_test_pm.rb
- test/extra/redmine_pm/test_case.rb
- test/extra/redmine_pm/repository_subversion_test_pm.rb

However, I think I can close this issue because frozen_string_literal is enabled for 98% of files and features implemented by the above files are not so popular.

I hope someone tests the above files and enables frozen_string_literal for those in the future.

#18 - 2019-03-21 07:23 - Go MAEDA

- Status changed from New to Resolved

#19 - 2019-03-21 07:24 - Go MAEDA

- Subject changed from Frozen string literals to Enable frozen string literals

- Assignee set to Go MAEDA

#20 - 2019-03-24 02:24 - Go MAEDA

- Status changed from Resolved to Closed

#21 - 2019-09-16 11:12 - Go MAEDA

- Related to Defect #32071: FrozenError in assert_save in test/test_helper.rb added

#22 - 2019-11-02 14:53 - Marius BĂLTEANU

- Related to Defect #31508: Add missing frozen strings and copyrights added

#23 - 2022-03-22 07:36 - Go MAEDA

- Related to Defect #33021: [v.4.0.5-stable] Internal Server Error 500 when accessing 'repository' tab added

Files

frozen_string_literals.patch	541 KB	2017-07-27	Pavel Rosický
frozen.patch	86.8 KB	2019-03-06	Pavel Rosický