

## ChiliProject - Bug # 952: Engines tests are broken in current versions of Ruby

<b>Status:</b>	Closed	<b>Priority:</b>	Normal
<b>Author:</b>	Jan Vlnas	<b>Category:</b>	Testing
<b>Created:</b>	2012-03-25	<b>Assignee:</b>	Felix Schäfer
<b>Updated:</b>	2012-06-23	<b>Due date:</b>	
<b>Remote issue URL:</b>			
<b>Affected version:</b>	master		
<b>Description:</b>	While working at #944, I've found that tests for Engines are broken both in MRI 1.8.7 and 1.9.X " in 1.8.7, the problem is with exception_notification plugin (used by test app); 1.9 has problems with @require@.		
	I've merged Redmine/ChiliProject changes with the original repository and extracted the plugin to the standalone repo at <a href="https://github.com/jrv/engines">https://github.com/jrv/engines</a> for easier testing.		
	Note that these changes shouldn't affect functionality of Engines.		
	Once ChiliProject migrates to Rails 3.2 (or newer), Engines plugin will be effectively obsolete. Until then, ChiliProject should provide stable and compatible testing environment for current plugins " and Engines plugin should be tested too.		

### Associated revisions

#### 2012-06-23 09:23 pm - Jan Vlnas

Correct engine tests to work on Ruby 1.9 #952 #944

### History

#### 2012-03-26 02:32 pm - Jan Vlnas

- Assignee set to Jan Vlnas

- Status changed from Open to Ready for review

<https://github.com/chiliproject/chiliproject/pull/176>

#### 2012-05-20 07:17 pm - Felix Schäfer

- Status changed from Ready for review to Needs more information

I see this updates the Engines to 1.2, what's the difference to the current version?

Furthermore: What does @upgrade\_engines\_migration.rb@ do? It seems to me it does things that should already be there (e.g. the @plugin\_assets@ thing) and the plugin migrations are not in @engine\_schema\_info@ making @rename\_table(:engine\_schema\_info, :plugin\_schema\_info) rescue nil@ unneeded too.

#### 2012-05-20 08:33 pm - Jan Vlnas

As mentioned in description, I have stitched ChiliProject's/Redmine's changes with the original Engines repo, so the @version.rb@ file and addition of @upgrade\_engines@ generator are side effects of this -- as you can see from the diff, there are no functional changes to the plugin itself and these files have no actual effect on Rake or ChiliProject anyway. My idea was to have the last official version of Engines in ChiliProject; I actually think it'd be better to have plugins as git submodules or merged subtrees (or better yet, as Bundler dependencies) so the modifications can be easily tracked and extracted, however I can see it's not worth the effort at this point, as most of plugins aren't maintained anymore.

Anyway, if you really don't like those files, I can remove them from pull request.

Note that the real reason I did this was to test Engines due to #944 -- I have this fixed in Engines repo at <https://github.com/jrv/engines> so you can use

that instead.

**2012-05-21 05:55 am - Felix Schäfer**

Jan Vlnas wrote:

> As mentioned in description, I have stitched ChiliProject's/Redmine's changes with the original Engines repo, so the `@version.rb@` file and addition of `@upgrade_engines@` generator are side effects of this -- as you can see from the diff, there are no functional changes to the plugin itself and these files have no actual effect on Rake or ChiliProject anyway. My idea was to have the last official version of Engines in ChiliProject;

Well, the one thing is that the engines plugin has been patched so much I'd wager it's not compatible with the original one anymore, the other thing is the first commit says "Engines version 2.1.0", so 1.2.0 seems weird to me

<https://github.com/chiliproject/chiliproject/commits/master/vendor/plugins/engines>

I can see how the `load_error` initializer is needed, the other things are at best "noise" (are the changes to how the paths for the requires needed?), at worst misleading (wrong version (as far as I can tell), mentions to not-applicable migrations).

> I actually think it'd be better to have plugins as git submodules or merged subtrees (or better yet, as Bundler dependencies) so the modifications can be easily tracked and extracted, however I can see it's not worth the effort at this point, as most of plugins aren't maintained anymore.

You don't know how much we'd love to throw out code in favor of gems, most stuff in `@vendor/plugins@` has been patched so much that replacing it with the original versions will be hard. In the case of the engines, as the original version isn't maintained anymore and as it's such a cornerstone of the way plugins work, I'd as much want to keep it there until we don't need it anymore to avoid weird problems.

> Anyway, if you really don't like those files, I can remove them from pull request.

I'd rather have a minimal pull request that (only) fixes existing problems than trying to sync to an abandoned plugin, i.e. if you could rework the pull request to only include actual fixes (e.g. the `load_error` initializer) it would be great :-). (sorry to be so conservative on this one, but we can't afford errors on the engines because it would affect all installed plugins, and the engines have been patched so much I'm weary of "it's from upstream so it's ok" changes)

> Note that the real reason I did this was to test Engines due to #944 -- I have this fixed in Engines repo at <https://github.com/jnv/engines> so you can use that instead.

Yeah, I had seen #944, the pull request 176 doesn't fix that though, does it?

**2012-05-21 09:55 am - Jan Vlnas**

> I can see how the `load_error` initializer is needed, the other things are at best "noise" (are the changes to how the paths for the requires needed?), at worst misleading (wrong version (as far as I can tell), mentions to not-applicable migrations).

Now that you made me think about that, I see what mistakes I've made. `@version.rb@` and `@upgrade_engines@` were actually removed from the original repository, stitching the commits brought them back. Also, you are right about `@expand_path@`, I think I've run into tests not finding `@test_helper@` at one point, and overreacted a bit. :)

I've removed the unneeded changes from the pull request and now it is. Rather short: <https://github.com/chiliproject/chiliproject/pull/176/files>

> Yeah, I had seen #944, the pull request 176 doesn't fix that though, does it?

No, I will send a separate pull request for this ASAP, including additional engines tests.

**2012-05-21 01:00 pm - Felix Schäfer**

On a side-note: Do you think we should run the tests for the engines plugin with the core tests? And if so, can they be run from rake?

**2012-05-21 01:48 pm - Jan Vlnas**

I don't think that Engines' tests should be part of the core test suite given the changes to this plugin are rare. It would just complicate the testing

process.

You could, however, extract Engines plugin to a standalone repository under ChiliProject organization and integrate it into main source tree "using subtree":<http://apenwarr.ca/log/?m=200904#30> "nothing will change for end users and developers (no need to run @git submodule @), but changes to Engines can be easily tracked and tested, including CI.

**2012-05-21 02:09 pm - Holger Just**

Jan Vlnas wrote:

> You could, however, extract Engines plugin to a standalone repository under ChiliProject organization and integrate it into main source tree "using subtree":<http://apenwarr.ca/log/?m=200904#30> "nothing will change for end users and developers (no need to run @git submodule @), but changes to Engines can be easily tracked and tested, including CI.

Given that we will retire the old engines with (or very soon after) the upgrade to Rails 3, I don't feel comfortable with overly messing with it now. We should keep it running as long as required, but I don't think this component deserves this amount of care (and possibilities of breakage) anymore.

**2012-05-21 08:28 pm - John Yani**

Jan Vlnas wrote:

> Once ChiliProject migrates to Rails 3.2 (or newer), Engines plugin will be effectively obsolete.

Holger Just wrote:

> Given that we will retire the old engines with (or very soon after) the upgrade to Rails 3

I would advise against doing so. Firstly, because there is no built-in possibility to migrate Engines plugin's plugins database to Rails 3 Engines'. Secondly, maintaining a Redmine and Chiliproject compatible plugin would be even harder if not impossible (Redmine still uses Engines).

**2012-05-22 05:13 pm - Jan Vlnas**

John Yani wrote:

> I would advise against doing so. Firstly, because there is no built-in possibility to migrate Engines plugin's plugins database to Rails 3 Engines'. Secondly, maintaining a Redmine and Chiliproject compatible plugin would be even harder if not impossible (Redmine still uses Engines).

To my best knowledge, Redmine 2.0 (Rails 3.2.3) doesn't support plugins written for 1.x line (Rails 2.3.14). There's no reason to keep old Engines module around when you have native Rails support for mountable engines. Also, new Engines are far from the only change in Rails 3, plugins would have to be modified anyway.

And honestly, I hope that ChiliProject is willing to do some serious code and tests refactoring, not "just" support Rails 3.

Anyway, this isn't the right place to discuss Rails 3 compatibility unlike #601. I don't think that Engines extraction could break anything, but I understand it's not worth an effort.

Do you have any further comments to this pull request?

**2012-05-22 09:22 pm - Felix Schäfer**

Jan Vlnas wrote:

> Do you have any further comments to this pull request?

I tried to give your pull request a ride but was unable to sort through what was needed to run the tests for the engines, could you give me a quick walk-through?

**2012-05-22 09:43 pm - Jan Vlnas**

Felix Schäfer wrote:

> I tried to give your pull request a ride but was unable to sort through what was needed to run the tests for the engines, could you give me a quick walk-through?

Sure. Engines have some minimal dependencies (nothing extra compared to ChiliProject), take a look at "Gemfile in a standalone repo":<https://github.com/jnv/engines/blob/master/Gemfile> @test-unit@ here is for 1.9.3 compatibility, same case as #1013, @sqlite3@ is needed by a test application.

Otherwise the process is described in @README@ â€” the default Rake task for Engines generates documentation, you have to run @rake test@.

**2012-05-22 09:52 pm - Jan Vinas**

Of course, you have to run @rake test@ from @vendor/plugins/engines@ in ChiliProject's tree. Just for the completeness.

**2012-06-18 11:19 am - Jan Vinas**

Bump.

Any chance of getting this into 3.3.0 along with #944?

**2012-06-22 08:44 am - Felix Schäfer**

- Status changed from Needs more information to Ready for review

- Assignee changed from Jan Vinas to Felix Schäfer

Haven't come around to re-testing it, sorry (and it didn't pop up on my radar in that time :-/ ).

**2012-06-23 06:58 pm - Felix Schäfer**

- Target version set to 3.3.0

OK, I was able to reproduce this and confirm that your patches indeed work, I'm not sure how to best handle this thoughâ€¦

Holger, should we:

- \* just merge the patches because we've confirmed locally that they work,
- \* just merge the patches because we've confirmed locally and on Jan's Travis account that they work and trust him to report any more bugs that might occur there,
- \* split the engines out into a new repo in the chili org and work with it as a subtree in the main chili repo?

Jan, regarding the subtree part: if I understood correctly how they work, we could just leave the code as it is in the main repo but have it additionally mirrored in a separate repository, correct? (I haven't been able to try it out because it seems it hasn't made it into stable git, has it?)

**2012-06-23 08:20 pm - Jan Vinas**

git-subtree was merged into Git some three months ago. It's available just in recent versions of Git (1.7.11). You can still install git-subtree manually â€” just put the git-subtree / git-subtree.sh in your \$PATH, see <https://github.com/apenwarr/git-subtree/blob/master/INSTALL>

It's not exactly mirroring, you still have to sync/extract changes from/to a separate repo manually (using @subtree merge@ and @subtree push@) â€” this is similar to submodules which are also bound to a particular commit. You will probably have to remove vendor/plugins/engines path first and then merge it back as subtree. Otherwise it will work exactly like a regular directory, no need to run additional commands for end users. Also, you don't need to have git-subtree installed if you don't want to manipulate subtrees.

**2012-06-23 09:25 pm - Felix Schäfer**

- Status changed from Ready for review to Closed

OK, I've talked with Holger on IRC and we'll include submitted patches we won't put in a separate repo or test it on a CI though as we're not planning on changing code in there.

Everything merged in commit:29af3ec

(Oh, and I have git 1.7.11 and no subtreeâ€¦!)

**2012-06-23 09:49 pm - Jan Vinas**

Great, thank you! I actually hope ChiliProject will migrate to Rails 3 before the Engines plugin manages to break again; I see there's no point in maintaining a separate repository.

Subtree is in contrib dir: <https://github.com/git/git/tree/v1.7.11/contrib/subtree> (@/usr/share/doc/git/contrib@ on Debian-based systems with @git@ package) so yeah, some manual assembly is still requiredâ€¦!