

## Redmine - Defect #15827

### Redmine is using a buggy Rails SQLite adapter

2014-01-08 23:28 - Timur Alperovich

<b>Status:</b>	Closed	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	Database	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Affected version:</b>	2.4.0
<b>Resolution:</b>	Fixed		
<b>Description</b> <p>Redmine is hitting at least two bugs in the Rails SQLite adapter (there is one more, but Rails folks stopped accepting patches against 3.2.xx).</p> <p>The issues are: <a href="https://github.com/rails/rails/issues/13595">https://github.com/rails/rails/issues/13595</a> <a href="https://github.com/rails/rails/issues/13631">https://github.com/rails/rails/issues/13631</a></p> <p>I would encourage the bugs that are tracking these errors (such as <a href="#">#13210</a>) to be re-opened and examine the root of the problem, as opposed to suggesting workarounds (changing the database engine to MySQL or another DB). This affects <b>all</b> Redmine versions released that use Rails 3.2.xx.</p>			
<b>Related issues:</b> <p>Related to Redmine - Defect #13210: 500 with sqlite exception (database locked) <b>Closed</b></p>			

#### History

##### #1 - 2014-01-08 23:33 - Timur Alperovich

Since Rails itself will not see these bugfixes in the 3.2 version (since they stopped accepting them), Redmine should either explicitly state that SQLite3 is **not** a supported backend altogether or provide a patched Rails gem to use it. As is, closing every bug that refers to these problems with recommendation to upgrade to MySQL does not reflect the actual problem.

##### #2 - 2014-01-10 00:05 - Jan Niggemann (redmine.org team member)

Do I get this right:

- We provide a software that can use different 3rd party backends.
- One of those backends is broken.
- You expect *us* to fix / work around it?

Sorry to say, but fixing other projects' bugs is well beyond the scope and possibility of the very small redmine development team.

As is, closing every bug that refers to these problems with recommendation to upgrade to MySQL does not reflect the actual problem.

While that is true in the sense that we are not able to help all users affected by bugs in this 3rd party backend, there's nothing we can do about that, except encourage you to develop patches, send them to the rails devs and fight to get them accepted...

##### #3 - 2014-01-10 00:34 - Timur Alperovich

Jan Niggemann (redmine.org team member) wrote:

Do I get this right:

- We provide a software that can use different 3rd party backends.
- One of those backends is broken.
- You expect *us* to fix / work around it?

Sorry to say, but fixing other projects' bugs is well beyond the scope and possibility of the very small redmine development team.

Please see the note below. I'd like to clarify the expectation: if a project is tightly coupled with a buggy project, it's a problem. Rails 3.2.xx has bugs which will not be fixed and some of them are severe. The bugs may affect specific database backends that Rails uses, but the bug is not in SQLite3 or the SQLite3 gem. It is in ActiveRecord -- an important component of a tightly coupled framework (Rails) that Redmine uses.

Given how tightly Redmine is coupled with Rails, is it unreasonable to expect that the team would put in some effort in understanding the Rails bugs it's hitting? Especially, if the bugs will never be fixed in the Rails version the project uses?

As is, closing every bug that refers to these problems with recommendation to upgrade to MySQL does not reflect the actual problem.

While that is true in the sense that we are not able to help all users affected by bugs in this 3rd party backend, there's nothing we can do about that, except encourage you to develop patches, send them to the rails devs and fight to get them accepted...

Do you mean the third party back-end being Rails? Because that is where the bug is (not in SQLite3). Specifically, the bug is in ActiveRecord.

I suppose maybe "Rails SQLite adapter" somehow implied "the bug is in SQLite3 gem or C implementation". I'm sorry, maybe I did not make it clear -- the bug is in Rails (not in SQLite3 gem or its C library). Is there a way to use Redmine without Rails? As far as I can tell, Rails is not one of "N possible backends", where one could stop using Rails and work around it. Rails itself supports multiple backends and it turns out that this particular Rails bug only affects SQLite3. If the Rails bug affected MySQL I sincerely doubt the response to the issue would be the same (switch to a different database).

Thus I made the suggestion that SQLite3 should not be listed as supported by Redmine -- the Rails version Redmine uses has severe bugs in its use of SQLite3 (ActiveRecord may lock the database from reads by any other processes until its terminated, not rollback transactions properly, and, less severe, attempt to rollback transactions twice). Or at the very least, inform the users that these are outstanding issues that Rails stopped accepting bugfixes for and Redmine will not patch bugs in Rails.

Lastly, if you check the issues referenced in the bug, they actually have pull requests to fix them. Those pull requests will not make into Rails 3.2 as it is closed for bug fixes (as stated in the bug) -- that is why I wanted to inform the consumer of Rails that there are bugs in ActiveRecord. Otherwise, this issue would simply be: "upgrade the Rails version to xxx".

#### **#4 - 2014-01-15 11:02 - Toshi MARUYAMA**

- Description updated

- Category set to Database

#### **#5 - 2014-01-15 11:03 - Toshi MARUYAMA**

- Related to Defect #13210: 500 with sqlite exception (database locked) added

#### **#6 - 2017-04-16 03:06 - Go MAEDA**

- Status changed from New to Closed

- Resolution set to Fixed

Timur Alperovich wrote:

Redmine is hitting at least two bugs in the Rails SQLite adapter (there is one more, but Rails folks stopped accepting patches against 3.2.xx).

The issues are:

<https://github.com/rails/rails/issues/13595>

<https://github.com/rails/rails/issues/13631>

Both of these issues have been fixed in Rails 4.

<https://github.com/rails/rails/issues/13595#issuecomment-31610226>

<https://github.com/rails/rails/issues/13631#issuecomment-41931366>

Current versions of Redmine (3.2/3.3) uses Rails 4.2, so I believe that Redmine is not affected by these issues now.