

Redmine - Defect #12501

Wiki history partially doesn't work after upgrade

2012-12-04 13:28 - Stanislav German-Evtushenko

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Wiki	Estimated time:	0.00 hour
Target version:		Affected version:	2.1.4
Resolution:	Wont fix		
Description			
Situation:			
<ul style="list-style-type: none">I have upgraded Redmine from 1.2-stable to 2.1-stable.Wiki history is stored in gzip.			
Problem:			
<ul style="list-style-type: none">When I try to open old revisions of some pages (not for all) I get the message: Internal error An error occurred on the page you were trying to access. If you continue to experience problems please contact your Redmine administrator for assistance. If you are the Redmine administrator, check your log files for details about the error.			
<ul style="list-style-type: none">In logs:			
<pre>Started GET "/projects/project1/wiki/Start/diff/60" for 10.10.10.10 at 2012-12-04 16:23:26 +0400 Processing by WikiController#diff as HTML Parameters: {"project_id"=>"project1", "id"=>"Start", "version"=>"60"} Current user: admin (id=1) Completed 500 Internal Server Error in 10ms Zlib::DataError (invalid bit length repeat): app/models/wiki_content.rb:97:in `inflate' app/models/wiki_content.rb:97:in `text' app/models/wiki_page.rb:190:in `initialize' app/models/wiki_page.rb:120:in `new' app/models/wiki_page.rb:120:in `diff' app/controllers/wiki_controller.rb:202:in `diff'</pre>			
Info page:			
Default administrator account changed True			
Attachments directory writable True			
Plugin assets directory writable True			
RMagick available (optional) True			
Environment:			
Redmine version 2.1.4.stable.10899			
Ruby version 1.9.3 (x86_64-linux)			
Rails version 3.2.8			
Environment production			
Database adapter SQLite			
Redmine plugins:			
redmine_blogs 0.4.0			

Gem list:

```
actionmailer (3.2.8)
actionpack (3.2.8)
activemodel (3.2.8)
activerecord (3.2.8)
activeresource (3.2.8)
activesupport (3.2.8)
acts-as-taggable-on (2.3.3)
arel (3.0.2)
builder (3.0.0)
bundler (1.2.3)
coderay (1.0.8)
daemon_controller (1.1.0)
erubis (2.7.0)
fastthread (1.0.7)
hike (1.2.1)
i18n (0.6.1)
journey (1.0.4)
jquery-rails (2.0.3)
json (1.7.5)
mail (2.4.4)
mime-types (1.19)
multi_json (1.3.7)
net-ldap (0.3.1)
passenger (3.0.18)
pg (0.14.1)
polyglot (0.3.3)
rack (1.4.1)
rack-cache (1.2)
rack-openid (1.3.1)
rack-ssl (1.3.2)
rack-test (0.6.2)
rails (3.2.8)
railties (3.2.8)
rake (10.0.2)
rdoc (3.12)
rmagick (2.13.1)
ruby-openid (2.1.8)
rubygems-bundler (1.1.0)
rvm (1.11.3.5)
sprockets (2.1.3)
sqlite3 (1.3.6)
thor (0.16.0)
tilt (1.3.3)
treetop (1.4.12)
tzinfo (0.3.35)
```

History

#1 - 2012-12-04 14:10 - Stanislav German-Evtushenko

I have checked the database and found out that old gzip records and new gzip records of "wiki_content_versions" are different. New one looks like a text with X' at the beginning:

```
X'HEXHEXHEXHEX'
```

but old one is more like a binary

```
'x?u?OJ?@.????a??t?z?.....etc'
```

How can I fix that?

#2 - 2012-12-04 16:07 - Etienne Massip

PostgreSQL v9?

#3 - 2012-12-04 17:54 - Stanislav German-Evtushenko

SQLite

#4 - 2012-12-04 19:58 - Stanislav German-Evtushenko

I have just tried to proceed as clean migration as possible. I made "sqlite3 .dump" on the old system and created new production db with that dump on the new system. Then I made "rake db:migrate". And now I get the following message for old gzipped history:

```
Started GET "/projects/project1/wiki/Start?version=57" for 10.10.10.10 at 2012-12-04 22:47:20 +0400
Processing by WikiController#show as HTML
  Parameters: {"version"=>"57", "project_id"=>"project1", "id"=>"Start"}
  Current user: anonymous
  Rendered wiki/_content.html.erb (1.1ms)
  Rendered wiki/show.html.erb within layouts/base (7.6ms)
Completed 500 Internal Server Error in 21ms
```

```
ActionView::Template::Error (invalid code -- missing end-of-block):
  1: <div class="wiki wiki-page">
  2:   <%= textilizable content, :text, :attachments => content.page.attachments,
  3:     :edit_section_links => (@sections_editable && {:controller => 'wiki', :action => 'edit', :proje
ct_id => @page.project, :id => @page.title}) %>
  4: </div>
app/models/wiki_content.rb:97:in `inflate'
app/models/wiki_content.rb:97:in `text'
app/helpers/application_helper.rb:522:in `textilizable'
app/views/wiki/_content.html.erb:2:in `_app_views_wiki__content_html_erb__46270150915704852_59849880'
app/views/wiki/show.html.erb:37:in `_app_views_wiki_show_html_erb__3888669060056873954_57312980'
app/controllers/wiki_controller.rb:92:in `show'
```

#5 - 2012-12-04 20:00 - Stanislav German-Evtushenko

Any ideas?

#6 - 2012-12-05 08:06 - Stanislav German-Evtushenko

Please help! I'm not sure if I have to go back to Redmine 1.2 so far and drop any changes had been made since upgrade (two days of changes for now) or I can still work with the new Redmine 2.1 and will be able to move history from the old database in an appropriate way later.

#7 - 2012-12-05 10:09 - Stanislav German-Evtushenko

I have just checked Redmine 1.4-stable. It uses same gzip format as Redmine 1.2-stable does:

```
INSERT INTO "wiki_content_versions" VALUES (1162,30,30,4,'x]SMk<DC>@^L<BD>/<EC>...etc','gzip','This change is ai
med to test gzip in SQLite database','2012-12-05 09:59:35',40);
```

#8 - 2012-12-05 15:52 - Stanislav German-Evtushenko

Can this change be the cause of the problem?

- RAILS 2:
 - http://apidock.com/rails/v2.1.0/ActiveRecord/ConnectionAdapters/SQLiteColumn/binary_to_string/class
 - http://apidock.com/rails/v2.3.8/ActiveRecord/ConnectionAdapters/SQLiteColumn/binary_to_string/class
- RAILS 3:
 - http://apidock.com/rails/v3.2.8/ActiveRecord/ConnectionAdapters/SQLiteColumn/binary_to_string/class

#9 - 2012-12-05 18:29 - Stanislav German-Evtushenko

I've done some more investigation. Please have a look.

- How old DB stores blob data:

```
$ sqlite3 production-test.db "update wiki_content_versions set data='test' where id=2294"
$ sqlite3 production-test.db "select data from wiki_content_versions where id=2294"
test
$ sqlite3 production-test.db "select QUOTE(data) from wiki_content_versions where id=2294"
'test'
```

- How new DB stores blob data:

```
$ echo -n 'test' | hexdump -ve '1/1 "%.2x"'
74657374
$ sqlite3 production-test.db "update wiki_content_versions set data=x'74657374' where id=2294"
$ sqlite3 production-test.db "select data from wiki_content_versions where id=2294"
test
$ sqlite3 production-test.db "select QUOTE(data) from wiki_content_versions where id=2294"
X'74657374'
```

- Old database (Redmine 1.2) contains blobs in binary representation and new database (Redmine 2.1) contains blobs in X'HEXHEXHEX' representation.
 - So if you try to get data using 'select data ...' query both databases give same results but if you try to get data using 'select QOUTE(data) ...' you get binary in the first case and X'HEXHEXHEX' in the second one.
- Problem is that old database style (with binary representation) doesn't work with the new Redmine version.

#10 - 2012-12-05 18:31 - Stanislav German-Evtushenko

Does it look like a Rails bug?

#11 - 2012-12-05 18:46 - Etienne Massip

Did you try to activate debug and trace SQL queries on your 2.x instance?

#12 - 2012-12-05 19:48 - Stanislav German-Evtushenko

No idea how to debug. Could you give a hint?

#13 - 2012-12-05 22:49 - Etienne Massip

Use source/trunk/config/additional_environment.rb.example.

#14 - 2012-12-06 07:18 - Stanislav German-Evtushenko

Don't see anything that can help:

```
Started GET "/projects/it-tasks/wiki/Pagel?version=11" for 10.10.10.10 at 2012-12-06 10:11:21 +0400
Processing by WikiController#show as HTML
Parameters: {"version"=>"11", "project_id"=>"it-tasks", "id"=>"Pagel"}
(0.2ms) SELECT MAX("settings"."updated_on") AS max_id FROM "settings"
User Load (0.1ms) SELECT "users".* FROM "users" WHERE "users"."type" IN ('User', 'AnonymousUser') AND "users"."id" = ? AND (users.status = 1) LIMIT 1 [{"id", 3}]
Current user: user1 (id=3)
Project Load (0.1ms) SELECT "projects".* FROM "projects" WHERE "projects"."identifiier" = 'it-tasks' LIMIT 1
Wiki Load (0.1ms) SELECT "wikis".* FROM "wikis" WHERE "wikis"."project_id" = 1 LIMIT 1
EnabledModule Load (0.1ms) SELECT name FROM "enabled_modules" WHERE "enabled_modules"."project_id" = 1
SQL (0.3ms) SELECT "members"."id" AS t0_r0, "members"."user_id" AS t0_r1, "members"."project_id" AS t0_r2, "members"."created_on" AS t0_r3, "members"."mail_notification" AS t0_r4, "projects"."id" AS t1_r0, "projects"."name" AS t1_r1, "projects"."description" AS t1_r2, "projects"."homepage" AS t1_r3, "projects"."is_public" AS t1_r4, "projects"."parent_id" AS t1_r5, "projects"."created_on" AS t1_r6, "projects"."updated_on" AS t1_r7, "projects"."identifiier" AS t1_r8, "projects"."status" AS t1_r9, "projects"."lft" AS t1_r10, "projects"."rgt" AS t1_r11, "roles"."id" AS t2_r0, "roles"."name" AS t2_r1, "roles"."position" AS t2_r2, "roles"."assignable" AS t2_r3, "roles"."builtin" AS t2_r4, "roles"."permissions" AS t2_r5, "roles"."issues_visibility" AS t2_r6 FROM "members" LEFT OUTER JOIN "projects" ON "projects"."id" = "members"."project_id" LEFT OUTER JOIN "member_roles" ON "member_roles"."member_id" = "members"."id" LEFT OUTER JOIN "roles" ON "roles"."id" = "member_roles"."role_id" WHERE "members"."user_id" = 3 AND (projects.status<>9) ORDER BY projects.name
WikiPage Load (0.3ms) SELECT "wiki_pages".* FROM "wiki_pages" WHERE "wiki_pages"."wiki_id" = 1 AND (LOWER(title) = LOWER('Pagel')) ORDER BY title LIMIT 1
WikiContent Load (0.1ms) SELECT "wiki_contents".* FROM "wiki_contents" WHERE "wiki_contents"."page_id" = 100 LIMIT 1
WikiContent::Version Load (0.2ms) SELECT "wiki_content_versions".* FROM "wiki_content_versions" WHERE "wiki_content_versions"."wiki_content_id" = 100 AND "wiki_content_versions"."version" = 11 LIMIT 1
Wiki Load (0.1ms) SELECT "wikis".* FROM "wikis" WHERE "wikis"."id" = 1 LIMIT 1
Project Load (2.1ms) SELECT "projects".* FROM "projects" WHERE "projects"."id" = 1 LIMIT 1
CACHE (0.0ms) SELECT name FROM "enabled_modules" WHERE "enabled_modules"."project_id" = 1
WikiPage Load (0.1ms) SELECT "wiki_pages".* FROM "wiki_pages" WHERE "wiki_pages"."id" = 100 LIMIT 1
CACHE (0.0ms) SELECT "wiki_contents".* FROM "wiki_contents" WHERE "wiki_contents"."page_id" = 100 LIMIT 1
(0.1ms) SELECT "users".id FROM "users" INNER JOIN "watchers" ON "users"."id" = "watchers"."user_id" WHERE "users"."type" IN ('User', 'AnonymousUser') AND "watchers"."watchable_id" = 100 AND "watchers"."watchable_type" = 'WikiPage'
WikiPage Load (0.1ms) SELECT "wiki_pages".* FROM "wiki_pages" WHERE "wiki_pages"."id" = 269 LIMIT 1
WikiPage Load (0.1ms) SELECT "wiki_pages".* FROM "wiki_pages" WHERE "wiki_pages"."id" = 67 LIMIT 1
WikiPage Load (0.1ms) SELECT "wiki_pages".* FROM "wiki_pages" WHERE "wiki_pages"."id" = 1 LIMIT 1
CACHE (0.0ms) SELECT "wikis".* FROM "wikis" WHERE "wikis"."id" = 1 LIMIT 1
CACHE (0.0ms) SELECT "projects".* FROM "projects" WHERE "projects"."id" = 1 LIMIT 1
CACHE (0.0ms) SELECT "wikis".* FROM "wikis" WHERE "wikis"."id" = 1 LIMIT 1
CACHE (0.0ms) SELECT "projects".* FROM "projects" WHERE "projects"."id" = 1 LIMIT 1
CACHE (0.0ms) SELECT "wikis".* FROM "wikis" WHERE "wikis"."id" = 1 LIMIT 1
CACHE (0.0ms) SELECT "projects".* FROM "projects" WHERE "projects"."id" = 1 LIMIT 1
Rendered wiki/_content.html.erb (1.1ms)
Rendered wiki/show.html.erb within layouts/base (13.9ms)
Completed 500 Internal Server Error in 38ms
```

```
ActionView::Template::Error (invalid distance too far back):
 1: <div class="wiki wiki-page">
 2:   <%= textilizable content, :text, :attachments => content.page.attachments,
 3:     :edit_section_links => (@sections_editable && {:controller => 'wiki', :action => 'edit', :proje
ct_id => @page.project, :id => @page.title}) %>
 4: </div>
app/models/wiki_content.rb:97:in `inflate'
app/models/wiki_content.rb:97:in `text'
app/helpers/application_helper.rb:522:in `textilizable'
app/views/wiki/_content.html.erb:2:in `_app_views_wiki__content_html_erb__1100882616125802453_63646400'
app/views/wiki/show.html.erb:37:in `_app_views_wiki_show_html_erb__2952486958244708562_38946520'
app/controllers/wiki_controller.rb:92:in `show'
```

#15 - 2012-12-06 09:21 - Stanislav German-Evtushenko

I have written a script and converted all all the records to the X'HEXHEXHEX' format but that didn't help :(Each time I get "ActionView::Template::Error (invalid distance too far back)"

```
#!/bin/bash

db=production-old.db
db_out=production-new.db

id_list=$(echo "select id from wiki_content_versions where compression='gzip';" | sqlite3 $db)
for id in $id_list; do
  data=$(echo "select data from wiki_content_versions where id=$id;" | sqlite3 $db)
  data_hex=$(echo -n "$data" | hexdump -ve '1/1 "%.2x"')
  echo "update wiki_content_versions set data=x'$data_hex' where id=$id;" | sqlite3 $db_out
done
```

#16 - 2012-12-06 11:20 - Stanislav German-Evtushenko

Another idea. Can the moving from x32 to x64 system be the cause?

#17 - 2012-12-06 13:02 - Stanislav German-Evtushenko

Here is another test.

I have created two pages (one in Redmine 2.1 and another in Redmine 1.2) and made the same changes for them (current page text is "h1. Testpage2").

Databases contents:

- Redmine 2.1

```
00000000 78 da cb 30 d4 53 08 49 2d 2e 29 48 4c 4f 35 02 |x..0.S.I-.)HLO5.|
00000010 0a                                     |.|
00000011
```

- Redmine 1.2

```
00000000 78 da cb 30 d4 53 08 49 2d 2e 29 48 4c 4f 35 02 |x..0.S.I-.)HLO5.|
00000010 25 30 30 1c b9 04 57 0a                       |%00...W.|
00000018
```

Does anybody has an idea how to fix it so far?

#18 - 2012-12-06 15:25 - Stanislav German-Evtushenko

Stanislav German-Evtushenko wrote:

Can this change be the cause of the problem?

- RAILS 2:
 - http://apidock.com/rails/v2.1.0/ActiveRecord/ConnectionAdapters/SQLiteColumn/binary_to_string/class
 - http://apidock.com/rails/v2.3.8/ActiveRecord/ConnectionAdapters/SQLiteColumn/binary_to_string/class
- RAILS 3:
 - http://apidock.com/rails/v3.2.8/ActiveRecord/ConnectionAdapters/SQLiteColumn/binary_to_string/class

I was right. Problem is exactly there and this is bug of the old rails 2.x version.

I have manually replaced "25 30 30"

```
00000000 78 da cb 30 d4 53 08 49 2d 2e 29 48 4c 4f 35 02 |x..0.S.I-.)HLO5.|
00000010 25 30 30 1c b9 04 57 0a                       |%00...W.|
```

00000018

by "00"

```
00000000 78 da cb 30 d4 53 08 49 2d 2e 29 48 4c 4f 35 02 |x..0.S.I-.)HLO5.|
00000010 00 1c b9 04 57 0a |...W.|
00000016
```

and successfully decompressed the data! And now I have a lot of work with converting data. I have to convert all of "253030" to "00" and all of "253235" to "25". Can anybody suggest a quick solution?

#19 - 2012-12-06 16:10 - Stanislav German-Evtushenko

I did it! Hope it can save huge amount time for somebody who gets into the same situation.

```
#!/bin/bash

# production-old.db - Redmine 1.2 database
# production-new.db - Redmine 2.1 database (after db:migration)

db=production-old.db
db_out=production-new.db

id_list=$(echo "select id from wiki_content_versions where compression='gzip';" | sqlite3 $db)
for id in $id_list; do
    data=$(echo "select data from wiki_content_versions where id=$id;" | sqlite3 $db)
    data_hex=$(echo -n "$data" | sed 's/\x25\x30\x30/\x00/g; s/\x25\x32\x35/\x25/g' | hexdump -ve '1/1 "%."
2x"')
    echo "update wiki_content_versions set data=x'$data_hex' where id=$id;" | sqlite3 $db_out
done
```

#20 - 2012-12-06 16:31 - Stanislav German-Evtushenko

Would be nice to have this information on the page: [RedmineUpgrade#SQLite-database](#)

#21 - 2012-12-06 17:58 - Stanislav German-Evtushenko

Link to the related changes in Rails on GitHub: <https://github.com/rails/rails/compare/v3.2.1...v3.2.2#L66L5>

Pull request with a fix description: <https://github.com/rails/rails/pull/4809>

#22 - 2012-12-08 10:34 - Jean-Philippe Lang

- Status changed from New to Closed

- Resolution set to Wont fix

Thanks Stanislav for your investigation, I'm closing it as "Won't Fix" since its a Rails issue. I've added a note to [RedmineUpgrade](#).