

## Redmine - Defect #12505

### Git: fetch\_changesets returns a fatal error

2012-12-05 00:53 - Paul Wilson

<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>	SCM	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Affected version:</b>	2.0.4
<b>Resolution:</b>	Wont fix		
<b>Description</b>			
<p>For some time we have been updating changesets per project repository via a post-commit git hook call to script/runner as described in <a href="http://www.redmine.org/issues/8857#note-30">[[http://www.redmine.org/issues/8857#note-30]]</a>.</p> <p>With our recent upgrade from v1.2.1 to v2.0.4 (I have changed script/runner to script/rail runner) calls to this script are failing and changesets are not being updated.</p> <p>Running the script in a terminal window get me this output:</p> <pre>(eval):117: warning: multiple values for a block parameter (0 for 1) fatal: bad object &lt;some repository hash here&gt;</pre> <p>I've tried following the (substantial) dialog since I opened the original issue about git being too long in fetching repositories but I don't find a reference to this problem.</p> <p>Can someone point me in the right direction of a solution?</p>			
<b>Related issues:</b>			
Related to Redmine - Defect #11841: git log error: git exited with non-zero s...			<b>New</b>

## History

### #1 - 2012-12-05 01:23 - Paul Wilson

My current system info:

O/S is SLES 11 service pack 2

rake about:

```
env RAILS_ENV="production" rake about
About your application's environment
Ruby version          1.8.7 (x86_64-linux)
RubyGems version     1.3.6
Rack version         1.4
Rails version        3.2.6
Active Record version 3.2.6
Action Pack version  3.2.6
Active Resource version 3.2.6
Action Mailer version 3.2.6
Active Support version 3.2.6
Middleware           Rack::Cache, ActionDispatch::Static, Rack::Lock, #<ActiveSupport::Cache::Strategy::LocalCache::Middleware:0x7f6426fee4f8>, Rack::Runtime, Rack::MethodOverride, ActionDispatch::RequestId, Rails::Rack::Logger, ActionDispatch::ShowExceptions, ActionDispatch::DebugExceptions, ActionDispatch::RemoteIp, ActionDispatch::Callbacks, ActiveRecord::ConnectionAdapters::ConnectionManagement, ActiveRecord::QueryCache, ActionDispatch::Cookies, ActionDispatch::Session::CookieStore, ActionDispatch::Flash, ActionDispatch::ParamsParser, ActionDispatch::Head, Rack::ConditionalGet, Rack::ETag, ActionDispatch::BestStandardsSupport, OpenIdAuthenticaton
Application root     /srv/www/redmine-2.1-stable
Environment          production
Database adapter     mysql
Database schema version 20120422150750
```

### #2 - 2012-12-05 14:37 - Etienne Massip

- Status changed from New to Closed

- Resolution set to Invalid

Please have a look to [HowTo\\_setup\\_automatic\\_refresh\\_of\\_repositories\\_in\\_Redmine\\_on\\_commit](#).

I close this one as a support request and not a bug.

### #3 - 2012-12-05 17:26 - Paul Wilson

Call this whatever you wish, defect or support, but please, don't just dismiss it without any consideration.

Your referenced wiki page has among its instructions:

Notice: •Instead of calling an url, you can use **ruby /path\_to\_redmine/redmine/script/rails runner "Repository.fetch\_changesets" -e production > /dev/null 2>&1 &**.

•Note also that project identifier can either be the number of the project, e.g. id=1 or the identifier name that you gave the project, e.g., id=mobileapp

Here is the verbose output of just such a command on our system:

```
host:$/srv/www/redmine/script/rails runner "Repository.fetch_changesets" -e production
```

```
(eval):117: warning: multiple values for a block parameter (0 for 1)
    from /var/lib/rubygems/1.8/gems/actionpack-3.2.6/lib/action_dispatch/routing/mapper.rb:992
fatal: bad object 8c65df6fc974579cb16fa854f420c6936bd222e7
fatal: bad object fb8c07fb9180526fa5988ac21965fe1266953c95
fatal: bad object 365e7a540f99cf4da9e55dda0b7abbe4fe4db765
```

If these 3 fatal errors are expected behavior, then I suppose it is a support request but if not, then it may well be a defect or some other problem. Or perhaps repository updates by runner "Repository.fetch\_changesets" are no longer supported?

I'm not stuck on the syntax. I can change:

```
env PROJECT_IDENTIFIER=<project-id> RAILS_ENV=production /usr/bin/ruby /<path_to_my_redmine>/script/rails runner \
"Project.find_by_identifier(ENV["PROJECT_IDENTIFIER"]).try(:repository).try(:fetch_changesets)" &
```

To:

```
/usr/bin/ruby /srv/www/redmine/script/rails runner "Repository.fetch_changesets" -e production
```

but where do I specify the project id with this method? I don't find a reference to specifying the project\_id in the script/rails runner section of the how-to wiki. I'm not having any success using curl calls to "http://<my\_redmine\_server\_url>/sys/fetch\_changesets?key=<API KEY>." They aren't yielding any results for me. Changesets aren't being updated.

Also, on the referenced wiki how-to page, can I assume that '<your service key>' and '<API KEY>' are both references to the API key and that, in the Subversion section of Step 2, the wget example should be: wget "http://<redmine url>/sys/fetch\_changesets?key=<API KEY>" rather than wget "http://your/redmine/path/..."?

### #4 - 2012-12-05 17:26 - Paul Wilson

- Status changed from Closed to Reopened

### #5 - 2012-12-05 18:16 - Etienne Massip

Paul Wilson wrote:

Also, on the referenced wiki how-to page, can I assume that '<your service key>' and '<API KEY>' are both references to the API key and that, in the Subversion section of Step 2, the wget example should be: wget "http://<redmine url>/sys/fetch\_changesets?key=<API KEY>" rather than wget "http://your/redmine/path/..."?

Sure. This is a community-maintained page but I'm not sure that public edit access has been restored yet so I updated it.

### #6 - 2012-12-05 18:47 - Etienne Massip

BTW, wget method works for me on trunk, could you please check your production.log file contents?

### #7 - 2012-12-06 03:04 - Paul Wilson

Here's the production.log output from **/srv/www/redmine/script/rails runner "Repository.fetch\_changesets" -e production**

```
Connecting to database specified by database.yml
Creating scope :active. Overwriting existing method User.active.
Creating scope :open. Overwriting existing method Version.open.
DEPRECATION WARNING: The InstanceMethods module inside ActiveSupport::Concern will be no longer included automatically. Please define
instance methods directly in CollectiveIdea::Acts::NestedSet::Model instead. (called from include at
/srv/www/redmine-2.1-stable/lib/plugins/awesome_nested_set/lib/awesome_nested_set/awesome_nested_set.rb:58)
DEPRECATION WARNING: The InstanceMethods module inside ActiveSupport::Concern will be no longer included automatically. Please define
instance methods directly in CollectiveIdea::Acts::NestedSet::Model instead. (called from include at
/srv/www/redmine-2.1-stable/lib/plugins/awesome_nested_set/lib/awesome_nested_set/awesome_nested_set.rb:58)
Creating scope :open. Overwriting existing method Issue.open.
OpenIdAuthentication.store is nil. Using in-memory store.
git log error: git exited with non-zero status: 128
git log error: git exited with non-zero status: 128
git log error: git exited with non-zero status: 128
```

And with my previously working script/rails runner method:

```
host:$ PROJECT_IDENTIFIER="poppen"
RAILS_ENV="production"
env PROJECT_IDENTIFIER=$PROJECT_IDENTIFIER RAILS_ENV=$RAILS_ENV \
/srv/www/redmine/script/rails runner \
"Project.find_by_identifier(ENV['PROJECT_IDENTIFIER']).try(:repository).try(:fetch_changesets)" &
```

production.log output:

```
Connecting to database specified by database.yml
Creating scope :active. Overwriting existing method User.active.
Creating scope :open. Overwriting existing method Version.open.
DEPRECATION WARNING: The InstanceMethods module inside ActiveSupport::Concern will be no longer included automatically. Please define
instance methods directly in CollectiveIdea::Acts::NestedSet::Model instead. (called from include at
/srv/www/redmine-2.1-stable/lib/plugins/awesome_nested_set/lib/awesome_nested_set/awesome_nested_set.rb:58)
DEPRECATION WARNING: The InstanceMethods module inside ActiveSupport::Concern will be no longer included automatically. Please define
instance methods directly in CollectiveIdea::Acts::NestedSet::Model instead. (called from include at
/srv/www/redmine-2.1-stable/lib/plugins/awesome_nested_set/lib/awesome_nested_set/awesome_nested_set.rb:58)
Creating scope :open. Overwriting existing method Issue.open.
OpenIdAuthentication.store is nil. Using in-memory store.

Started GET "/" for 127.0.0.1 at Thu Dec 06 08:54:25 +0800 2012
Processing by WelcomeController#index as HTML
Redirected to http://sylvester.thenetcircle.lab:8001/login?back\_url=http%3A%2F%2Fsylvester.thenetcircle.lab%3A8001%2F
Filter chain halted as :check_if_login_required rendered or redirected
Completed 302 Found in 20ms (ActiveRecord: 3.2ms)
git log error: git exited with non-zero status: 128
```

To the best of my knowledge, the only change to this system was the redmine upgrade from 1.2.1 to 2.0.4. We have also the redmine\_backlogs plugin v0.9.26 so we are running tagged version 2.0.4 of redmine rather than 2.1-stable. Our git version is 1.8.0.

#### #8 - 2012-12-06 09:01 - Paul Wilson

I set up a testing environment today and restored my pre-upgrade redmine backup. Here's my initial testing environment (ruby version is 1.8.7):  
host:\$ gem list

- LOCAL GEMS \*\*\*

```
actionmailer (2.3.11)
actionpack (2.3.11)
activerecord (2.3.11)
activeresource (2.3.11)
activesupport (2.3.11)
daemon_controller (1.1.0)
fastthread (1.0.7)
i18n (0.4.2)
mysql (2.8.1)
passenger (3.0.18)
rack (1.1.0)
rails (2.3.11)
rake (10.0.2, 0.8.3)
rubygems-update (1.3.7)
```

With v1.2.1 installed, all fetch\_changeset methods work without error and changesets are updated. I can call the update with 'script/runner' or with

either curl or wget. All fire off the appropriate 'git' '--git-dir' ... commands for each repository.

Then I upgraded the redmine version to the desired 2.0.4.

I updated rubygems to 1.8.24 and installed the bundler gem.

From my redmine root directory executed 'bundle install --without test development sqlite postgresql rmagick' (my test system does not have rmagick installed)

I generated the session store secret, ran the db migration, and cleared the cache and existing sessions per the UPGRADING instruction document.

After an apache restart, I executed each of the aforementioned methods. All fail similarly and terminate with:

```
git log error: git exited with non-zero status: 128
```

```
git log error: git exited with non-zero status: 128
```

```
git log error: git exited with non-zero status: 128
```

#### #9 - 2012-12-06 13:33 - Michael D

I can confirm the problem after updating to 2.1.4.stable.10927

Problems seem to be git hashes which do not exist yet (or anymore). This was not a problem on the old server with Redmine 1.3.x

When calling fetch\_changesets on the command line, I get this output:

```
/usr/bin/ruby /...../redmine/script/rails runner "Repository.fetch_changesets" -e production
(eval):108: warning: multiple values for a block parameter (0 for 1)
      from /usr/lib/ruby/gems/1.8/gems/actionpack-3.2.8/lib/action_dispatch/routing/mapper.rb:992
fatal: bad object 45ee07f8801e0a7d477955972168d95343794938
```

When calling via URL this shows up in the production.log:

```
git log error: git exited with non-zero status: 128
```

Environment:

```
env RAILS_ENV="production" rake about
About your application's environment
Ruby version      1.8.7 (i386-linux)
RubyGems version  1.8.24
Rack version      1.4
Rails version     3.2.8
Active Record version 3.2.8
Action Pack version 3.2.8
Active Resource version 3.2.8
Action Mailer version 3.2.8
Active Support version 3.2.8
Middleware        Rack::Cache, ActionDispatch::Static, Rack::Lock, #<ActiveSupport::Cache::Strategy::LocalCache::Middleware:0xb6f14bd4>, Rack::Runtime, Rack::MethodOverride, ActionDispatch::RequestId, Rails::Rack::Logger, ActionDispatch::ShowExceptions, ActionDispatch::DebugExceptions, ActionDispatch::RemoteIp, ActionDispatch::Callbacks, ActiveRecord::ConnectionAdapters::ConnectionManagement, ActiveRecord::QueryCache, ActionDispatch::Cookies, ActionDispatch::Session::CookieStore, ActionDispatch::Flash, ActionDispatch::ParamsParser, ActionDispatch::Head, Rack::ConditionalGet, Rack::ETag, ActionDispatch::BestStandardsSupport, OpenIdAuthentication
Application root  /data/www/redmine
Environment       production
Database adapter  mysql
Database schema version 20120731164049
```

#### #10 - 2012-12-07 06:05 - Paul Wilson

I believe Michael D's assessment of the problem is correct and I have found a solution, albeit an ugly one.

I dropped and re-added each of the repositories from redmine, then updated the changesets. I initially tried to update all changesets in one op but that failed with duplicate insert errors. I was however, able to fetch changesets individually by project without errors. Now I have all changesets current and can update them by project without throwing any errors.

#### #11 - 2012-12-10 08:46 - Michael D

The solution described above (to drop and all repositories and then to adding them and updating the changesets repository per repository) worked for me too.

#### #12 - 2013-01-05 17:30 - Jan Kustrát

I've hit the same issue after an upgrade from 1.2.0 to 2.2.0.

## Quick summary

- the error is caused by Redmine trying to ask git for commits which are **no longer in the repository**
- Redmine gets these commit hashes from the extra\_info column in the repositories table and **fails to filter out for branches which are gone**

## Details

It looks like this is caused by Redmine asking git for non-existing revisions. According to strace, git is launched like this:

```
git --git-dir /var/lib/gitolite/repositories/trojita.git/ log --no-color --encoding=UTF-8 --raw --date=iso --pretty=fuller --parents --stdin --reverse
```

...and quite a few of the commit IDs is fed to its standard input. This one seems to be the offending one:

```
^8d364e155731b81071da332d8d163561c02ec3e5
```

Indeed, when I do the same manually, git complains as well:

```
bash-4.1$ echo "^8d364e155731b81071da332d8d163561c02ec3e5" | git --git-dir /var/lib/gitolite/repositories/trojita.git/ log --no-color --encoding=UTF-8 --raw --date=iso --pretty=fuller --parents --stdin --reverse
fatal: bad object 8d364e155731b81071da332d8d163561c02ec3e5
```

After some searching through the Ruby code, I've found out that the hashes come from the extra\_info field in the "repositories" table. In my case, the relevant row looks like this:

```
sqlite> select * from repositories where url like '%trojita%';
5|1|/var/lib/gitolite/repositories/trojita.git/|/var/lib/gitolite/repositories/trojita.git/|Repository::Git|
|---
extra_report_last_commit: "0"
db_consistent:
  ordering: 1
branches:
  condstore:
    last_scmid: 131c724675b0b829ec49855395660fa675e932b5
  esearch:
    last_scmid: a23c8c77f017a2a5bcdfa25538df2df49bae22df
  searching:
    last_scmid: 2de47d7bb53e021e448be34637b8a074db6a4df4
  master:
    last_scmid: 96a5bfbcf30600c79d62361143d817714f432c3d
  tl-composer-rework:
    last_scmid: c7c551ee309d19ee74d14f12a525f82eb987e5a1
  qresync:
    last_scmid: 5f2ecbd4fe19f01a655370823b248b752ba9206f
  pwn:
    last_scmid: 8d364e155731b81071da332d8d163561c02ec3e5
  mildred:
    last_scmid: 00b5811be1a1a28eaf22bb4609373213b59957a8
  imap-id:
    last_scmid: a5ee6fd739e338f0d9663392804ddedd30fdacf8
  qt5:
    last_scmid: d27f4f26eb286cb94e0392dfb775c3ab23740658
  unselect-rework:
    last_scmid: 7395ba33dafa7d26facb6facdbd3022b64157075
  fetch-header-fields:
    last_scmid: 0993257daef6c3c341ea192d2087a6bfbed7c793
  xtuple:
    last_scmid: cfb3ebc58ee1be2232a9ebcedd5cb77ff9fe0ec5
  fetching-headers:
    last_scmid: 6ef0d85d4c9213db32e34e4a48708961aaca79ff
  lineedit-icon:
    last_scmid: 02a12d93f913889ba490dc4d817eafa11eded3f5
  searching-filtering:
    last_scmid: db5cccae10bbe38681b39ed832ca01a652c26247
  qt5-vector-strict-iterators:
    last_scmid: 133d37244edac8b148bc0ace9caab87444e7fef2
  thebootroo:
    last_scmid: 7c0f845d029954ec5042655a5a5b3912b59eef6b
  qml:
    last_scmid: 7e732fbaffd2e0bf969a3354f4b01e59c00d34b5
  ssl:
    last_scmid: 6ccf91f8452b3ee474ed90ea1fafb868d138d67e
  maemo:
    last_scmid: 6e81454a6aa4987d96a0b5d49b2e2d620b32a23f
  harmattan:
    last_scmid: 995eb3c9ba861bce067ff1add1c123dda65aca31
|t
```

The offending stuff comes from the "pwn" branch which was deleted a long, long time ago. The branch is no longer in the repository:

```
bash-4.1$ git --git-dir /var/lib/gitolite/repositories/trojita.git/ branch --no-color --verbose --no-abbrev
fetching-headers          6ef0d85d4c9213db32e34e4a48708961aaca79ff Removing and commenting out code which
is "not required" for RFC2822 header parsing
lineedit-icon            02a12d93f913889ba490dc4d817eafalleded3f5 wip: broken implementation of "lineedit
with icon". Does not even compile.
maemo                    6e81454a6aa4987d96a0b5d49b2e2d620b32a23f added 1.5x text size multiplier into th
e embedded web view to compensate for the n900's ridiculously high PPI screen.
* master                 17ff40d9e159e25db67294aa4af2977daaa12cdc docs++
qt5-vector-strict-iterators 133d37244edac8b148bc0ace9caab87444e7fef2 Better name for the .pro file
thebootroo              7c0f845d029954ec5042655a5a5b3912b59eef6b got it compiling on desktop
tl-composer-rework      c7c551ee309d19ee74d14f12a525f82eb987e5a1 Don't use hardcoded numbers for the com
pletion count
unselect-rework         7395ba33dafe7d26facb6facdbd3022b64157075 wip
```

...and the commit has been apparently removed by a periodic `git gc run --git` is by default configured to remove unreachable commits after a certain time passes; in this case, the throwaway branch was deleted months ago:

```
/var/lib/gitolite/.gitolite/logs/gitolite-2011-06.log:2011-06-24.17:08:00      jkt      147.231.26.118  git-re
ceive-pack 'trojita'      +      8d364e155731b8  0000000000000000  trojita refs/heads/pwn  refs/.*
```

## Suggested fix

Make sure that the commits to exclude are only picked up from branches which are still in the repository.

### #13 - 2013-01-05 18:01 - Jan Kunderát

- *File prevent-git-log-wrong-revisions-bug-12505.patch added*

The attached patch makes sure that only those branches which are still in the repository are considered when preparing a list of commits to exclude. It fixes the issue for me.

The proper fix shall probably make sure that the `extra_info` column is cleaned up after receiving the list of branches from the SCM. However, this patch is the first Ruby line I've ever written, so I'll leave the real fix to someone else.

### #14 - 2013-01-06 02:14 - Toshi MARUYAMA

- *Subject changed from fetch\_changesets returns a fatal error to Git: fetch\_changesets returns a fatal error*

### #15 - 2013-01-11 21:53 - Dave Ingram

I was getting this same error on a clean install of Redmine 2.2.0 on Ubuntu 12.04. The initial page load brought in all commits up to that point, but subsequent commits pushed to the repository were no longer updating in Redmine.

I can't tell if I exactly followed the comments in <http://www.redmine.org/issues/12505#note-10>, but what I did was go to the project administration page, then to repositories, deleted the existing repository and then re-added the repository. I then ran:

```
/var/redmine/script/rails runner "Repository.fetch_changesets" -e production
```

This pulled in all of the latest changes and has continued to work since.

### #16 - 2013-02-09 15:11 - Nicklas Overgaard

I was facing the same problem on Redmine 2.2.2, came from 1.2.SOMETHING.

The patch from comment 13 solved the issue, so perhaps this could just be merged into the next maintenance release of the 2.2 series?

### #17 - 2013-02-09 18:31 - Toshi MARUYAMA

- *File git.diff added*

- *Status changed from Reopened to Closed*

- *Resolution changed from Invalid to Wont fix*

I have updated note-13 patch.

But I cannot accept this change.

Because revisions on **shared** repository should not be deleted.

See Mercurial FAQ [I committed a change containing nuclear launch codes, how do I delete it permanently?](#)

In this case, you need to clear "extra\_info" column manually.

### #18 - 2014-11-17 12:50 - Anonymous

Toshi MARUYAMA wrote:

I have updated note-13 patch.  
But I cannot accept this change.  
Because revisions on **shared** repository should not be deleted.  
See Mercurial FAQ [I committed a change containing nuclear launch codes, how do I delete it permanently?](#)

In this case, you need to clear "extra\_info" column manually.

It doesn't matter if a revision **should** not be deleted: it happens in real-life (it happened to us recently on redmine 2.5.2).

The workaround that consists of deleting & re-importing the repository has its flaws: it may be slow, and most importantly, some commits may have been manually re-associated to issues (in case of error in the commit message association), we would lose that. It may also do additional actions via the commit logs: close the issue, it may have been reopened for some reasons and thus re-closing it is not a good idea; and there is also time logging via commit messages... (and maybe other features I'm not aware of).

The other workaround requiring to patch the db manually is even less user-friendly

So I believe that if the patch to fix this in redmine is simple (which it seems), then I see no reason to refuse to merge it in redmine.

Could you please reopen & fix this issue?

Thanks.

#### #19 - 2014-11-17 13:08 - Toshi MARUYAMA

Thomas Riccardi wrote:

It doesn't matter if a revision **should** not be deleted: it happens in real-life (it happened to us recently on redmine 2.5.2).

Because you use "git push -f" and "git gc".

The workaround that consists of deleting & re-importing the repository has its flaws: it may be slow, and most importantly, some commits may have been manually re-associated to issues (in case of error in the commit message association), we would lose that. It may also do additional actions via the commit logs: close the issue, it may have been reopened for some reasons and thus re-closing it is not a good idea; and there is also time logging via commit messages... (and maybe other features I'm not aware of).

This is partially fixed by [#4823](#).

The other workaround requiring to patch the db manually is even less user-friendly

This is covered in #1273.

#### #20 - 2014-11-17 13:16 - Anonymous

Toshi MARUYAMA wrote:

Thomas Riccardi wrote:

It doesn't matter if a revision **should** not be deleted: it happens in real-life (it happened to us recently on redmine 2.5.2).

Because you use "git push -f" and "git gc".

Yes, "git push -f" happens in real-life.

And "git gc" may be started automatically by some commands.

The workaround that consists of deleting & re-importing the repository has its flaws: it may be slow, and most importantly, some commits may have been manually re-associated to issues (in case of error in the commit message association), we would lose that. It may also do additional actions via the commit logs: close the issue, it may have been reopened for some reasons and thus re-closing it is not a good idea; and there is also time logging via commit messages... (and maybe other features I'm not aware of).

This is partially fixed by [#4823](#).

Ok, good to know.

The other workaround requiring to patch the db manually is even less user-friendly

This is covered in #1273.

Ok.  
This still requires admin intervention in such case, whereas with the proposed patch here nothing would be required.

Are there other reasons (than "this should not happen") to **not** include this patch?

**#21 - 2014-11-17 13:32 - Toshi MARUYAMA**

Thomas Riccardi wrote:

Are there other reasons (than "this should not happen") to **not** include this patch?

Because you should not use "git push -f".  
<https://twitter.com/rhodecode/status/528290436532940800>

**#22 - 2014-11-17 13:39 - Toshi MARUYAMA**

Thomas Riccardi wrote:

Yes, "git push -f" happens in real-life.  
And "git gc" may be started automatically by some commands.

<http://git-scm.com/docs/git-gc>

Some git commands may automatically run git gc; see the --auto flag below for details. If you know what you're doing and all you want is to disable this behavior permanently without further considerations, just do:

```
$ git config --global gc.auto 0
```

**#23 - 2014-11-17 14:17 - Anonymous**

Toshi MARUYAMA wrote:

Thomas Riccardi wrote:

Are there other reasons (than "this should not happen") to **not** include this patch?

Because you should not use "git push -f".  
<https://twitter.com/rhodecode/status/528290436532940800>

I know, and I don't, but I'm not the only one to use git at my work, and then we need an admin to fix the issue, it takes time, and it could easily be avoided with the patch.

However, I would understand the rejection of this patch if guaranteeing that redmine always works in such broken conditions would be a lot of work or hacks. But it seems that only this small patch is needed, so that's why I think this should be merged.

Toshi MARUYAMA wrote:

Thomas Riccardi wrote:

Yes, "git push -f" happens in real-life.  
And "git gc" may be started automatically by some commands.

<http://git-scm.com/docs/git-gc>

Some git commands may automatically run git gc; see the --auto flag below for details. If you know what you're doing and all you want is to disable this behavior permanently without further considerations, just do:

[...]

I know this, but I don't want to disable auto git gc: it's useful!

**#24 - 2014-11-17 14:31 - Toshi MARUYAMA**

Thomas Riccardi wrote:

But it seems that only this small patch is needed, so that's why I think this should be merged.



If you want, please cover all cases in tests.  
I don't want to run "git gc" on tests and I don't want to maintain it.

**#25 - 2014-11-17 14:47 - Toshi MARUYAMA**

Thomas Riccardi wrote:

Toshi MARUYAMA wrote:

Because you should not use "git push -f".  
<https://twitter.com/rhodecode/status/528290436532940800>

I know, and I don't, but I'm not the only one to use git at my work, and then we need an admin to fix the issue, it takes time, and it could easily be avoided with the patch.

Please ask admin to use "receive.denyNonFastForwards = true".  
<http://git-scm.com/book/en/v2/Customizing-Git-Git-Configuration>

**#26 - 2018-08-23 12:55 - Alexandr Kirilov**

Might be related <https://www.redmine.org/issues/29416>

**#27 - 2021-03-11 14:00 - Erik E**

From what I understand I ran into similar issues here:  
[How to reinit "Latest revisions" \(changesets\) for repository](#)

**Files**

---

prevent-git-log-wrong-revisions-bug-12505.patch	477 Bytes	2013-01-05	Jan Kandrát
git.diff	4.01 KB	2013-02-09	Toshi MARUYAMA