

## Redmine - Defect #1419

### Error fetching git changesets

2008-06-10 17:35 - Brandon Keepers

<b>Status:</b>	Closed	<b>Start date:</b>	2008-06-10
<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>	0.8	<b>Affected version:</b>	
<b>Resolution:</b>	Fixed		
<b>Description</b>			
<p>I've been using redmine for a while now, but a few weeks ago the changesets stopped importing from git. When I run <code>Repository.fetch_changesets</code>, I get this error:</p>			
<pre>ActiveRecord::StatementInvalid: Mysql::Error: Column 'changeset_id' cannot be null: INSERT INTO changes (`changeset_id`, `action`, `revision`, `branch`, `from_path`, `from_revision`, `path`) VALUES (NULL, 'M', NULL, NULL, NULL, NULL, 'config/aaf.rb')   from /usr/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'   from /usr/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/mysql_adapter.rb:243:in `execute'   from /usr/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/mysql_adapter.rb:253:in `insert'   from /usr/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:1811:in `create_without_callbacks'   from /usr/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/callbacks.rb:254:in `create_without_timestamps'   from /usr/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/timestamp.rb:39:in `create'   from /usr/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:1789:in `create_or_update_without_callbacks'   from /usr/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/callbacks.rb:242:in `create_or_update'   from /usr/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:1545:in `save_without_validation'   from /usr/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/validations.rb:752:in `save_without_transactions'   from /usr/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/transactions.rb:129:in `save'   from /usr/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in `transaction'   from /usr/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/transactions.rb:95:in `transaction'   from /usr/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/transactions.rb:121:in `transaction'   from /usr/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/transactions.rb:129:in `save'   from /usr/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:451:in `create'   from ./script/../config/../config/../app/models/repository/git.rb:59:in `fetch_changesets'   from ./script/../config/../config/../app/models/repository/git.rb:58:in `each'   from ./script/../config/../config/../app/models/repository/git.rb:58:in `fetch_changesets'   from ./script/../config/../config/../app/models/repository/git.rb:50:in `reverse_each'   from ./script/../config/../config/../app/models/repository/git.rb:50:in `fetch_changesets'   from /usr/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in `transaction'   from /usr/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/transactions.rb:95:in `transaction'   from /usr/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/transactions.rb:121:in `transaction'   from ./script/../config/../config/../app/models/repository/git.rb:49:in `fetch_changesets'   from /usr/lib/ruby/gems/1.8/gems/activesupport-1.4.2/lib/active_support/core_ext/symbol.rb:10:in `__send__'</pre>			

```
from /usr/lib/ruby/gems/1.8/gems/activesupport-1.4.2/lib/active_support/core_ext/symbol.rb:10:
in `to_proc'
from ./script/./config/./config/./app/models/repository.rb:87:in `each'
from ./script/./config/./config/./app/models/repository.rb:87:in `fetch_changesets'
```

## Associated revisions

---

### Revision 1953 - 2008-10-25 12:23 - Jean-Philippe Lang

Check that git changeset is not in the database before creating it (#1419).

## History

---

### #1 - 2008-06-16 07:35 - Jean-Philippe Lang

- Status changed from New to Closed
- Resolution set to Invalid

Please read [SubmittingBugs](#) and provide the required information.

### #2 - 2008-06-16 13:56 - Brandon Keepers

- MySQL 4.1.20
- Ruby 1.8.6
- Rails 2.0.2
- Redmine [r1524](#)
- Git 1.5.3.4

### #3 - 2008-08-26 16:32 - Luc Heinrich

- Status changed from Closed to Reopened

I get the same problem here, it suddenly started to happen today, without notice. It was working perfectly an hour ago and now poof, I get the same exception.

- MySQL 4.1.22
- Ruby 1.8.7 (2008-08-11 patchlevel 72) [powerpc-darwin8]
- Rails 2.0.2
- Redmine [r1557](#)
- Git 1.6.0

### #4 - 2008-08-26 17:46 - Luc Heinrich

Still occurring after upgrading to:

- Rails 2.1
- Redmine [r1769](#)

### #5 - 2008-08-28 02:52 - Brandon Keepers

I was able to get around this by deleting all the existing Changesets and then re-importing them:

```
script/runner 'Changeset.destroy_all; Repository.fetch_changesets'
```

### #6 - 2008-09-06 09:55 - Luc Heinrich

Yes, deleting changesets and re-importing them work, that's what I did too. But the bug can reappear after a while. It did for me yesterday...

### #7 - 2008-09-19 09:12 - C G

Git does not return the commits in chronological order.

So this bug is initiated by the sort order of the *has\_many :changesets* association in the *repository* model.

This ordering affects the result of *repository#latest\_changeset* which will be given to *git\_adapter#revisions* as the *identifier\_from* parameter. Thus the changesets fetched from git might contain changesets, that have been created in the redmine database before.

The fix we applied is:

```
in repository.rb:
```

```
+ has_many :changesets, :dependent => :destroy, :order => "#{Changeset.table_name}.committed_on DESC, #{Change
set.table_name}.id DESC"
- has_many :changesets, :dependent => :destroy, :order => "#{Changeset.table_name}.id DESC"
```

And it seems to work fine so far. (No need to delete all changesets and reimport them.)

#### #8 - 2008-09-19 11:45 - C G

Another solution to the above problem would be to override `repository#latest_changeset` within the `git.rb`, so only git will be influenced by that fix.

```
# in git.rb:
def latest_changeset
  changesets.find(:first, :order => "#{Changeset.table_name}.id DESC")
end
```

#### #9 - 2008-09-24 09:15 - C G

A quick correction: in [#note-7](#) the '+' and '-' signs of the diff are permuted. It should be

```
- has_many :changesets, :dependent => :destroy, :order => "#{Changeset.table_name}.committed_on DESC, #{Change
set.table_name}.id DESC"
+ has_many :changesets, :dependent => :destroy, :order => "#{Changeset.table_name}.id DESC"
```

#### #10 - 2008-10-25 12:18 - Jean-Philippe Lang

- *Status changed from Reopened to Closed*
- *Target version set to 0.8*
- *Resolution changed from Invalid to Fixed*

[r1953](#) should fix this problem. It now checks that the changeset is not in the database before inserting it.  
I prefer not to rely on the order in which git returns the changesets.