

ステータス:	New	開始日:	2022/05/09
優先度:	通常	期日:	
担当者:		進捗率:	0%
カテゴリ:	SCM_3	予定工数:	0.00時間
対象バージョン:		作業時間:	0.00時間
Redmineorg_URL:	https://www.redmine.org/issues/18754	status_id:	1
category_id:	3	tracker_id:	1
version_id:	0	plus1:	0
issue_org_id:	18754	affected_version:	
author_id:	118295	closed_on:	
assigned_to_id:	0	affected_version_id:	
comments:	8		

説明

If svn xml log contains logentry without data the revision reading is interrupted and no error message is written to redmine logs.

Revisions are read from xml returned from command:

```
svn log --xml ...
```

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
mkrol
2015-01-02T06:24:51.507749Z
test commit
```

```
...
```

@logentry@ for revision 2177 is empty - there is no @author@, @msg@ and @date@ tags.

But the code assumes that @revision@, @date@ and @msg@ exist in @logentry@.

This is the code from @lib/redmine/scm/adapters/subversion_adapter.rb@, lines 175-196:

```
begin
  doc = parse_xml(output)
  each_xml_element(doc['log'], 'logentry') do |logentry|
    (...)

    revisions << Revision.new({:identifier => logentry['revision'],
                              :author => (logentry['author'] ? logentry['author']['__content__'] :
                              :time => Time.parse(logentry['date']['__content__'].to_s).localtime,
                              :message => logentry['msg']['__content__'],
                              :paths => paths
                              })
  end
rescue
end
```

If @@ does not exist the @logentry['msg']@ returns @nil@. The call @logentry['msg']['content']@ becomes @nil['content']@ and exception is thrown (then it is ignored and it is ignored outside loop...).

The existing of @logentry['author']@ is checked before reading @logentry['author']['content']@. It should be done before reading @date@ and @msg@ also. For example:

```
revisions << Revision.new({:identifier => logentry['revision'],
                             :author => (logentry['author'] ? logentry['author']['__content__'] :
                             :time => (logentry['date'] ? Time.parse(logentry['date']['__content__'] :
                             :message => (logentry['msg'] ? logentry['msg']['__content__'] : ""
                             :paths => paths
                             })
```

Why there are empty revisions in svn log? This is when revision has only files which are not visible for user performing @svn log@ command. Probably it's a rare case.

journals

My environment:

Environment:

Redmine version	2.6.0.stable.13735
Ruby version	1.9.3-p194 (2012-04-20) [x86_64-linux]
Rails version	3.2.21
Environment	production
Database adapter	Mysql2

SCM:

Subversion	1.6.17
Git	1.7.10.4
Filesystem	

Redmine plugins:

redmine_checklists	3.0.2
redmine_close_button	0.0.8

We face this issue often because the repository we use contains protected data along with the actual project data. This happens actually very often, provided you have some permission management inside svn. I think this is not awkward or a special case at all and should be fixed.

IMHO this ticket duplicates: #3663, #7086

I created a patch to address this problem, instead of bailing out for the whole batch any empty log entries are skipped and a message is logged (loglevel INFO)

Stephan Wiehr wrote:

I created a patch to address this problem, instead of bailing out for the whole batch any empty log entries are skipped and a message is logged (loglevel INFO)

I think logging is reasonable, but discrete revision causes another problem.

I think re-raising exception with logging is better.

Toshi MARUYAMA wrote:

Stephan Wiehr wrote:

I created a patch to address this problem, instead of bailing out for the whole batch any empty log entries are skipped and a message is logged (loglevel INFO)

I think logging is reasonable, but discrete revision causes another problem.

I think re-raising exception with logging is better.

The current code just stops processing further logentries and then consumes the exception in an unconditional begin ... rescue end block.

The provided patch is meant not only to log but also to enable continuation of the logentry processing.

Current code:

```
begin
  doc = parse_xml(output)
  each_xml_element(doc['log'], 'logentry') do |logentry|
    [code where exception occurs]
  end
rescue
end
```

Code with patch:

```
begin
  doc = parse_xml(output)
  each_xml_element(doc['log'], 'logentry') do |logentry|
    begin
      [code where exception occurs]
    rescue
      logger.info "Skipped svn revision ###{logentry['revision']} due to parse error" if logentry && logentry['revision']
    end
  end
end
rescue
end
```

I don't see how re-raising the exception would help here.

related_issues

relates,Needs feedback,3663,Problem with subversion logentry without date nor author.
relates,New,7086,Few fixes for subversion

履歴

#1 - 2022/05/10 17:09 - Admin Redmine

- カテゴリ を SCM_3 にセット