

Redmine - Defect #18754

empty svn logentry crashes revision batch reading

2015-01-05 14:02 - Michał Król

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	SCM	Estimated time:	0.00 hour
Target version:		Affected version:	
Resolution:			

Description

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"?>
<log>
<logentry
  revision="2178">
<author>mkrol</author>
<date>2015-01-02T06:24:51.507749Z</date>
<msg>test commit</msg>
</logentry>
<logentry
  revision="2177">
</logentry>
...
```

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 <msg> 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__']).to_s).localtime : nil),
  :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.

Related issues:

Related to Redmine - Defect # 3663: Problem with subversion logentry without ...

Needs feedback 2009-07-24

Related to Redmine - Patch # 7086: Few fixes for subversion

New 2010-12-09

History

#1 - 2015-01-05 14:09 - Michał Król

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
```

#2 - 2015-01-05 14:32 - Toshi MARUYAMA

- Category set to SCM

#3 - 2018-01-03 11:44 - Jaap de Haan

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

#4 - 2018-01-03 12:10 - Stephan Wiehr

- *File subversion_adapter.rb.patch added*

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)

#5 - 2018-01-31 16:47 - Toshi MARUYAMA

- *Related to Defect #3663: Problem with subversion logentry without date nor author. added*

#6 - 2018-01-31 16:48 - Toshi MARUYAMA

- *Related to Patch #7086: Few fixes for subversion added*

#7 - 2018-01-31 16:55 - Toshi MARUYAMA

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.

#8 - 2018-04-19 16:00 - Stephan Wiehr

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
rescue
end
```

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

Files

subversion_adapter.rb.patch	2.54 KB	2018-01-03	Stephan Wiehr
-----------------------------	---------	------------	---------------