

説明

In Trac API 0.12, the representation of timestamps was changed from seconds since the epoch to microseconds since the epoch:
<http://trac.edgewall.org/wiki/TracDev/ApiChanges/0.12#Timestampstorageindatabase>

See this patch that changed it in Trac:
<http://trac.edgewall.org/attachment/ticket/6466/6466-microsecond-times-r8770.patch>

If you try to run the existing `migrate_from_trac.rake` script, the year will be thousands of years in the future, and the importer will populate the date in redmine is 0000000000. This causes many things to break.
The problem still exists with the trunk version of the `migrate_from_trac.rake` script.

In #882 , the user encountered the problem, but went down the wrong path with the fix, because he didn't understand why timestamps were being imported incorrectly.

I am submitting a better fix, which detects the database version of Trac. If the database version is > 22, then every time a timestamp from Trac is encountered, we divide by 1 million.

This patch is against trunk.
I have used this patch successfully and it fixes all my problems importing from Trac.

I think many people are hitting these issues when doing Trac -> Redmine migrations. Here are some example tickets:

- #5764
- #8690
- #12477

journals

I have a copy of migrate_from_trac.rake in a Github fork, where I am preserving the patches:

https://github.com/rodrigc/redmine/blob/master/lib/tasks/migrate_from_trac.rake

Hopefully my patches can be incorporated into Redmine, so that I don't have to keep my own

private copy. This will benefit others who are trying to migrate to Redmine from Trac.

I used these patches to migrate about 3000 Trac tickets to Redmine. See:

"FreeNAS project migration from Trac to Redmine":/boards/1/topics/39532

Craig, could you please have a look at #14844? Would be great if this could be fixed in this patch. Shouldn't be too hard but I have no Ruby experience else I would do it on my own.

Marc,

I'll see what I can do, but can't promise anything. My main reasons for fixing the migrate_from_trac.rake script to do the following:

"FreeNAS project migration from Trac to Redmine":/boards/1/topics/39532

However, since I have finished that, I can't devote a lot more time to Remdmine, but I will try if I have time.

I would encourage you to take a whack at learning Ruby and trying to fix this script. I didn't know Ruby before fixing migrate_from_trac.rake but I learned enough to get going. The code in Redmine is very clean, so it is easy to pick up.

The code in Redmine which accesses the database is part of the ActiveRecord class in Rails. ActiveRecord has an API for doing transactions:

<http://api.rubyonrails.org/classes/ActiveRecord/Transactions.html>

Give it a shot and see if you can get something working, and I can review it.

related_issues

- relates,New,5764,migrate_from_trac does not support trac 0.12
- relates,Needs feedback,8690,User can't be displayed
- relates,Needs feedback,12477,Internal server error after migrate from Trac 0.12.3 ro Redmine 2
- relates,Closed,14590,migrate_from_trac.rake does not import Trac users, uses too short password
- relates,Closed,14592,migrate_from_trac.rake does not properly parse First Name and Last Name
- relates,New,14593,migrate_from_trac.rake, print out validation errors when doing save
- relates,New,14844,migrate_from_trac is not run in database transaction
- relates,Reopened,6868,migrate_from_trac and trac 0.12
- relates,New,10738,trac import issue
- relates,Resolved,20943,migrate_from_trac.rake dont work
- duplicates,Closed,14845,Error in user controller
- duplicates,Closed,17898,Exception in Migration from Trac to Redmine (incl. solution)

履歴

- #1 - 2022/05/10 17:12 - Admin Redmine
- カテゴリ を Importers_15 にセット