

Redmine - Defect #882

migrate_from_trac doesn't import timestamps of wiki and tickets

2008-03-17 23:15 - Andreas Neuhaus

Status:	Closed	Start date:	2008-03-17
Priority:	Normal	Due date:	
Assignee:		% Done:	90%
Category:	Importers	Estimated time:	0.00 hour
Target version:	0.7	Affected version:	
Resolution:	Fixed		

Description

After importing a Trac project with Redmine [r1270](#), I noticed that all imported wiki page history was timestamped with the current time. Also the tickets doesn't seem to have a start time set.

After a quick look into lib/tasks/import_from_trac.rake it looks like timestamps of wiki page edits aren't imported, although I can't explain why ticket timestamps fail to import.

Associated revisions

Revision 1302 - 2008-03-29 11:54 - Jean-Philippe Lang

Fixed: migrate_from_trac doesn't import timestamps of wiki and tickets (patch #882 by Andreas Neuhaus slightly edited).

History

#1 - 2008-03-26 13:59 - Andreas Neuhaus

- Status changed from New to Resolved
- Target version set to 0.7
- % Done changed from 0 to 90

I solved this problem in my case by overriding Time.now during the import. This ist more like a workaround and maybe it'd be more appropriate to disable the ActiveRecord timestamps at all during import and set created_on and updated_on attributes manually (but since I'm not very familiar with the Redmine models and the acts_as_versioned plugin, I had problems doing so and did this hack to be able to import my project into Redmine without losing the changelog timestamps)

I also added a comment and set the author to nil for the automatic page edit that is done for each completely imported page (where the wiki syntax of Trac is converted to Redmine)

```
Index: lib/tasks/migrate_from_trac.rake
=====
--- lib/tasks/migrate_from_trac.rake      (revision 1291)
+++ lib/tasks/migrate_from_trac.rake      (working copy)
@@ -69,6 +69,21 @@
     'developer' => developer_role
   }

+   class ::Time
+     class << self
+       alias :real_now :now
+       def now
+         real_now - @fake_diff.to_i
+       end
+       def fake (time)
+         @fake_diff = real_now - time
+         res = yield
+         @fake_diff = 0
+         res
+       end
+     end
+   end

  class TracComponent < ActiveRecord::Base
    set_table_name :component
  end
@@ -345,14 +360,14 @@
```

```

        i.tracker = TRACKER_MAPPING[ticket.ticket_type] || DEFAULT_TRACKER
        i.custom_values << CustomValue.new(:custom_field => custom_field_map['resolution'], :value =>
ticket.resolution) unless ticket.resolution.blank?
        i.id = ticket.id unless Issue.exists?(ticket.id)
-       next unless i.save
+       next unless Time.fake(ticket.time) { i.save }
        TICKET_MAP[ticket.id] = i.id
        migrated_tickets += 1

        # Owner
        unless ticket.owner.blank?
            i.assigned_to = find_or_create_user(ticket.owner, true)
-           i.save
+           Time.fake(ticket.time) { i.save }
        end

        # Comments and status/resolution changes
@@ -380,7 +395,7 @@
                                :old_value => resolution_change.oldvalue,
                                :value => resolution_change.newvalue)
        end
-       n.save unless n.details.empty? && n.notes.blank?
+       Time.fake(time) { n.save } unless n.details.empty? && n.notes.blank?
        end

        # Attachments
@@ -390,7 +405,7 @@
        a.file = attachment
        a.author = find_or_create_user(attachment.author)
        a.container = i
-       migrated_ticket_attachments += 1 if a.save
+       migrated_ticket_attachments += 1 if Time.fake(attachment.time) { a.save }
        end

        # Custom fields
@@ -426,7 +441,7 @@
        p.content.text = page.text
        p.content.author = find_or_create_user(page.author) unless page.author.blank? || page.author == '
trac'
        p.content.comments = page.comment
-       p.new_record? ? p.save : p.content.save
+       Time.fake(page.time) { p.new_record? ? p.save : p.content.save }

        next if p.content.new_record?
        migrated_wiki_edits += 1
@@ -439,13 +454,15 @@
        a.file = attachment
        a.author = find_or_create_user(attachment.author)
        a.container = p
-       migrated_wiki_attachments += 1 if a.save
+       migrated_wiki_attachments += 1 if Time.fake(attachment.time) { a.save }
        end
    end

    wiki.reload
    wiki.pages.each do |page|
        page.content.text = convert_wiki_text(page.content.text)
+       page.content.author = nil
+       page.content.comments = 'Migration to Redmine'
        page.content.save
    end
end

```

#2 - 2008-03-29 11:55 - Jean-Philippe Lang

- Status changed from Resolved to Closed

- Resolution set to Fixed

Committed in [r1302](#) with slight changes (eg. use ticket changetime instead of time since Time.fake is used to properly set updated_on properties).