migrate_from_mantis script fails with missing table name

See #10501 and #10504 for environment. This bug is with the patches supplied in those reports applied.

Migrating users......
rake aborted!

Mysql::Error: Table 'bugtracker.mantis_project_category_table' doesn't exist: SELECT * FROM `mantis_project_category_table` WHERE (`mantis_project_category_table`.project_id = 1)
/home/alan/dev/redmine/vendor/rails/activerecord/lib/active_record/connection_adapters/abstract_adapter.rb:227:in `log'
/home/alan/dev/redmine/vendor/rails/activerecord/lib/active_record/connection_adapters/mysql_adapter.rb:324:in `execute'
/home/alan/dev/redmine/vendor/rails/activerecord/lib/active_record/connection_adapters/mysql_adapter.rb:639:in `select'
/home/alan/dev/redmine/vendor/rails/activerecord/lib/active_record/connection_adapters/abstract/database_statements.rb:7:in `select_all_without_query_cache'
/home/alan/dev/redmine/vendor/rails/activerecord/lib/active_record/connection_adapters/abstract/query_cache.rb:62:in `select_all'
/home/alan/dev/redmine/vendor/rails/activerecord/lib/active_record/base.rb:665:in `find_by_sql'
/home/alan/dev/redmine/vendor/rails/activerecord/lib/active_record/base.rb:1582:in `find_every'
/home/alan/dev/redmine/vendor/rails/activerecord/lib/active_record/base.rb:619:in `find'
/home/alan/dev/redmine/vendor/rails/activerecord/lib/active_record/associations/association_collection.rb:60:in `find'
/home/alan/dev/redmine/vendor/rails/activerecord/lib/active_record/associations/association_collection.rb:423:in `find_target'
/home/alan/dev/redmine/vendor/rails/activerecord/lib/active_record/associations/association_collection.rb:365:in `load_target'
/home/alan/dev/redmine/vendor/rails/activerecord/lib/active_record/associations/association_proxy.rb:213:in `method_missing'
/home/alan/dev/redmine/lib/tasks/migrate_from_mantis.rake:392:in `method_missing'
/home/alan/dev/redmine/lib/tasks/migrate_from_mantis.rake:392:in `method_missing'
/var/lib/gems/1.8/gems/rake-0.9.2.2/lib/rake/task.rb:151:in `invoke'
/var/lib/gems/1.8/gems/rake-0.9.2.2/lib/rake/application.rb:133:in `standard_exception_handling'
/var/lib/gems/1.8/gems/rake-0.9.2.2/lib/rake/application.rb:66:in `run'
/var/lib/gems/1.8/gems/rake-0.9.2.2/lib/rake/application.rb:133:in `standard_exception_handling'
/var/lib/gems/1.8/gems/rake-0.9.2.2/lib/rake/application.rb:63:in `run'
Related issues:
Related to Redmine - Defect # 10504: migrate_from_mantis script fails on firs...  Resolved
Related to Redmine - Defect # 10501: migrate_from_mantis script fails on firs...  Closed

History

#1 - 2012-03-25 10:44 - Alan Chandler
- File migrate.patch added
- Status changed from New to Resolved
- Assignee set to Alan Chandler

As well as the issues report above and in #10501 and #10504 I found quite a few more issues - namely

- Normal Priority Issues get converted to High Priority in migration
- Projects marked as Private in Mantis where made public in Redmine
- Project Hierarchy was not respected
- The project identifier cuts out Upper Case characters - leaving some names pretty meaningless
- The issues updated_on field was set at the migration time rather than the time pulled over from Mantis
- The issue Created on field was left as an integer (in SQLite) rather than properly converted to a time.

I have fixed all of these and can now perform a complete migration with a good deal of success.

The Diff Patch between the version I have now and the 1.3.2 distribution is attached. (it superceeds the previous patches provided)

#2 - 2014-04-19 10:17 - Etienne Massip
- Target version set to Candidate for next minor release

#3 - 2014-05-18 05:42 - Toshi MARUYAMA
- Related to Defect #10504: migrate_from_mantis script fails on first project when trying to update version dates added

#4 - 2020-02-29 13:44 - Go MAEDA
- Related to Defect #10501: migrate_from_mantis script fails on first project using sqlite3 added

Files
migrate.patch  11 KB  2012-03-25  Alan Chandler