

ステータス:	Closed	開始日:	2022/05/09
優先度:	通常	期日:	
担当者:		進捗率:	100%
カテゴリ:	Database_21	予定工数:	0.00時間
対象バージョン:	2.3.0_60	作業時間:	0.00時間
Redmineorg_URL:	https://www.redmine.org/issues/12713	status_id:	5
category_id:	21	tracker_id:	2
version_id:	60	plus1:	0
issue_org_id:	12713	affected_version:	
author_id:	1	closed_on:	
assigned_to_id:	1	affected_version_id:	
comments:	35		
説明			
journals			
r11116 passes the full test suite with Microsoft SQL Server 2012 using these gems: • activerecord-sqlserver-adapter (3.2.10) * tiny_tds (0.5.1) After checking out redmine from svn trunk, migration task for new database was failed with error like this. Index name 'changesets_repos_rev' on table 'changesets' does not exist /work/redmine/db/migrate/091_change_changesets_revision_to_string.rb:3:in `up' I wonder this is the correct way or not, but this workaround works fine for me. I hope I can give you any help. <pre>--- db/migrate/091_change_changesets_revision_to_string.rb (revision 11125) +++ db/migrate/091_change_changesets_revision_to_string.rb (working copy) @@ -1,6 +1,8 @@ class ChangeChangesetsRevisionToString < ActiveRecord::Migration def self.up - remove_index :changesets, :name => :changesets_repos_rev + if index_exists? :changesets, :changesets_repos_rev + remove_index :changesets, :name => :changesets_repos_rev + end change_column :changesets, :revision, :string, :null => false add_index :changesets, [:repository_id, :revision], :unique => true, :name => :changesets_repos_rev end end</pre>			
I attached a corrected version of the patch. You've got a typo in changesets. Daniel Felix wrote: I attached a corrected version of the patch. You've got a typo in changesets.			

I'm so sorry and thank you for making the revisions.

Akiko Takano wrote:

After checking out redmine from svn trunk, migration task for new database was failed with error like this.

It works fine for me. Are you able to reproduce when running db:migrate on a new empty database? The index is created in a previous migration (@034_create_changesets.rb@) so I'd like to understand why you get this error before committing this patch.

Jean-Philippe Lang wrote:

It works fine for me. Are you able to reproduce when running db:migrate on a new empty database? The index is created in a previous migration (@034_create_changesets.rb@) so I'd like to understand why you get this error before committing this patch.

I can reproduce the error when I use sqlite3.
Here is the part of error message.

```
.....  
== ChangeChangesetsRevisionToString: migrating =====  
-- remove_index(:changesets, {:name=>:changesets_repos_rev})  
rake aborted!  
An error has occurred, this and all later migrations canceled:  
  
Index name 'changesets_repos_rev' on table 'changesets' does not exist  
/home/.../.rbenv/versions/1.9.2-p320/lib/ruby/gems/1.9.1/gems/activerecord-3.2.10/lib/active_record/connection_adapters/abstract/schema_statements.rb:587:in `index_name_for_remove'
```

Sorry I'm not sure this is because of my environment or not.

However when I use MySQL, migration to clean database is done without any problem. (Nothing to change from r11097)

Thanks, I am able to reproduce with sqlite3. The index actually exists before migration 91 but is named @altered_changesets_repos_rev@. I guess it was silently renamed by a prior migration that affects the revision column, and it does not occur with all backends. The problem with the patch is that we end up with 2 identical unique indexes, which is suboptimal. I'll try to find a better way to workaround this issue.

I've committed a fix in r11137. Note that the proposed patch was wrong (the index name was given as the column name and index_exists? was always returning false).

Jean-Philippe Lang wrote:

I've committed a fix in r11137. Note that the proposed patch was wrong (the index name was given as the column name and index_exists? was always returning false).

Thank you so much for your time. Since r11137, migration task works fine.

I'm also looking forward to support SQL Server.

I'm really pleased to see this feature as we run Redmine on SQL Server 2005 at the moment. Here are some of my notes - I hope they're helpful.

h2. Things I fix each time I install (updated for 2.2.0 on Ruby 1.8.7)

- In Ruby 1.8.7 /lib /ruby /gems /1.8 /specifications /tilt-1.3.3.gemspec, mail-2.4.4.gemspec and jquery-rails-2.0.3.gemspec change "s.date = %q{2011-05-21 00:00:00.000000000Z}" to "s.date = %q{2011-05-21}" (probably nothing to do with

Redmine but included for completeness)

- Fix line 34 of redmine-2.2.0 /app /controllers /versions_controller.rb. Change: @trackers = @project.trackers.find(:all, :order => 'position') to @trackers = @project.trackers.find(:all, :order => "#{Tracker.table_name}.position") as described in <http://www.redmine.org/issues/11002>
- Fix redmine-2.2.0 /app /models /workflow_rule.rb. In the SQL query at the bottom change the fieldname " rule " to " [rule] " in two locations.

h2. Known issues due to use of SQL server

- Gantt charts don ' t work
- Can ' t sort or group by a custom field
- Dates for versions don ' t work
- Calendars don't show any events

h2. Known issue which may or may not be SQL server

- Log time feature is very slow on Ruby 1.8.7, and crashes on Ruby 1.9.x with "SystemStackError (stack level too deep): actionpack (3.2.9) lib/action_dispatch/middleware/reloader.rb:70"

Hi Rick,

maybe you can try the trunk on a testserver?

Most of your reported bugs seem to be fixed in the current trunk. :-)

Best regards,

Daniel

Rick Mason wrote:

- In Ruby1.8.7 /lib /ruby /gems /1.8 /specifications /tilt-1.3.3.gemspec, mail-2.4.4.gemspec and jquery-rails-2.0.3.gemspec change "s.date = %q{2011-05-21 00:00:00.000000000Z}" to "s.date = %q{2011-05-21}" (probably nothing to do with Redmine but included for completeness)

I think this can be solved by upgrading your rubygems

- Fix line 34 of redmine-2.2.0 /app /controllers /versions_controller.rb. Change: @trackers = @project.trackers.find(:all, :order => 'position') to @trackers = @project.trackers.find(:all, :order => "#{Tracker.table_name}.position") as described in <http://www.redmine.org/issues/11002>
- Fix redmine-2.2.0 /app /models /workflow_rule.rb. In the SQL query at the bottom change the fieldname " rule " to " [rule] " in two locations.

Fixed

- Gantt charts don ' t work
- Can ' t sort or group by a custom field
- Dates for versions don ' t work
- Calendars don't show any events

Fixed

- Log time feature is very slow on Ruby 1.8.7, and crashes on Ruby 1.9.x with "SystemStackError (stack level too deep): actionpack (3.2.9) lib/action_dispatch/middleware/reloader.rb:70"

Logging time works fine for me with SQLServer and other databases, this should be reported in a specific issue.

Any reason why the gem declarations haven't been added to Gemfile?

Installing lots of unused gems when running a simple `@bundle install@` is a problem IMHO. For example, `tiny_tds` won't compile unless you have `freetds` installed, resulting in an error when bundling. And having to specify all the databases you're not using with the `@--without@` option is far from user-friendly.

I'm thinking of modifying the Gemfile so that only the gems required for the database adapters used in the database configuration file (`database.yml`) are actually declared. The install process would then be:

`configure database.yml`

`run bundle install`

Please, have a look at the attached patch. Feedback is welcome.

having to specify all the databases you're not using with the `--without` option is far from user-friendly.

I agree, your patch is a good idea and will simplify the installation for novice users.

Jean-Philippe Lang wrote:

Installing lots of unused gems when running a simple `@bundle install@` is a problem IMHO. For example, `tiny_tds` won't compile unless you have `freetds` installed, resulting in an error when bundling. And having to specify all the databases you're not using with the `@--without@` option is far from user-friendly.

Fully agree.

I'm thinking of modifying the Gemfile so that only the gems required for the database adapters used in the database configuration file (`database.yml`) are actually declared. The install process would then be:

`configure database.yml`

`run bundle install`

Looks fine.

Please, have a look at the attached patch. Feedback is welcome.

Nice patch. What about:

- not aborting on unknown adapter, just warn? People should then be able to use their own adapter in `Gemfile.local`
- handling and aborting if no adapter has been found

FWIW I migrated a live PostgreSQL DB to SQL Server and everything is working like a charm right now.

Etienne Massip wrote:

FWIW I migrated a live PostgreSQL DB to SQL Server and everything is working like a charm right now.

Thanks for the feedback. Tests are now running with SQL Server 2012 on the "integration server":
<http://www.redmine.org/builds/index.html>.

For linux users: with `freetds 0.82` that was packaged for the server distro, a few tests was breaking the build with lots of `@ActiveRecord::StatementInvalid: TinyTds::Error: Attempt to initiate a new Adaptive Server operation with results pending"@ errors`. I had to install `freetds 0.91` to fix that.

Might be worth it supporting SQL Server on JRuby too?

Index: Gemfile

```
=====
--- Gemfile (revision 11250)
+++ Gemfile (working copy)
@@ -53,9 +53,10 @@
     when /sqlite3/
       gem "sqlite3", :platforms => [:mri, :mingw]
       gem "activerecord-jdbcsqlite3-adapter", :platforms => :jruby
-   when /sqlserver/
+   when /sqlserver/, /jdbcmssql/
       gem "tiny_tds", "> 0.5.1", :platforms => [:mri, :mingw]
       gem "activerecord-sqlserver-adapter", :platforms => [:mri, :mingw]
+   gem "activerecord-jdbcmsql-adapter", :platforms => :jruby
     else
       warn("Unknown database adapter `#{adapter}`, use Gemfile.local to load your own database gems")
     end
```

I haven't tried yet but I plan to do so (some day). ----- I was not able to make it work with JRuby: * using jdbc-jtds: I'm getting this error on startup: ActiveRecord::JDBCError: The driver encountered an unknown error: cannot link Java class net.sourceforge.jtds.jdbc.Driver, probable missing dependency: net/sourceforge/jtds/jdbc/Driver : Unsupported major.minor version 51.0@ * using jdbc-mssql-azure: while it's incompatible according to the "activerecord-jdbc-adapter documentation":<https://github.com/jruby/activerecord-jdbc-adapter#inside-rails>, the application starts. But still, a migration raises an error that I don't get when using mri+freetds Good luck!

----- I was able to have it running with JRuby 1.7.2 except I had to use @gem "activerecord-jdbcmsql-adapter", :git => 'git://github.com/jruby/activerecord-jdbc-adapter.git', :branch => 'jdbc-jtds-1.2.x', :platform => :jruby@ because of "this issue":<https://github.com/jruby/activerecord-jdbc-adapter/issues/290>. Then I had some issue with badly displayed accentuated characters but since it affects also translated labels it should be caused by some Puma incompatibility. Edit: I haven't hit your issue :| ----- Jean-Philippe Lang wrote: > * using jdbc-mssql-azure: while it's incompatible according to the "activerecord-jdbc-adapter documentation":<https://github.com/jruby/activerecord-jdbc-adapter#inside-rails>, the application starts. But still, a migration raises an error that I don't get when using mri+freetds There is a note here about Azure: <https://github.com/jruby/activerecord-jdbc-adapter/tree/master/activerecord-jdbcmsql-adapter>.

----- Etienne Massip wrote: > I was able to have it running with JRuby 1.7.2 except I had to use @gem "activerecord-jdbcmsql-adapter", :git => 'git://github.com/jruby/activerecord-jdbc-adapter.git', :branch => 'jdbc-jtds-1.2.x', :platform => :jruby@ because of "this issue":<https://github.com/jruby/activerecord-jdbc-adapter/issues/290>. Then we'll wait for the fixed gem. ----- tiny_tds 0.5.1 fails connecting with integrated security on Windows systems. v0.6.0.rc1 fixes this issue. Can you update the gem file?

----- Marcel Nadjé wrote: > v0.6.0.rc1 fixes this issue. 0.5.1 works fine for me on Windows. Do you have any reference of this issue? ----- Jean-Philippe Lang wrote: > 0.5.1 works fine for me on Windows. Do you have any reference of this issue? "Any plan on supporting Windows Authentication?":<https://groups.google.com/forum/?fromgroups#topic/rails-sqlserver-adapter/tJS1mfA1EEI>

----- I'd like not to bundle RC gems in Redmine so I think you'll have to edit your Gemfile if you want to use this one. ----- Sorry => AR-JDBC 1.2.6 "incoming":<https://github.com/jruby/activerecord-jdbc-adapter/commit/0c0fc102424b2fb0490f14e95401cb35f0b3601d> and should make SQLServer JDBC adapter usable. ----- AR-JDBC 1.2.6 "has been released":<https://rubygems.org/gems/activerecord-jdbcmsql-adapter>. ----- Still doesn't work for me with JRuby1.7.2 and:

```
gem "activerecord-jdbc-adapter", "1.2.6"
gem "jdbc-jtds", "> 1.2.7", :platforms => :jruby
gem "activerecord-jdbcmsql-adapter", :platforms => :jruby
```

I get this on db:migrate:

```
== SetLanguageLengthToFive: migrating =====
-- change_column(:users, :language, :string, {:limit=>5, :default=>""})
rake aborted!
An error has occurred, this and all later migrations canceled:
```

```
ActiveRecord::JDBCError: Table sysobjects does not exist: select def.name from sysobjects def, syscolumns col, sysobjects tab where col.cdefault = def.id and col.lname = 'language' and tab.name = 'users' and col.id = tab.id
```

I'm closing it since it's no longer an issue about Redmine support for SQLServer. I'd prefer

that we track this in a separate ticket.

@gem "activerecord-jdbc-mssql-adapter", "~> 1.2.6" :platforms => :jruby@ is enough since @jdbc-jtds@ and @activerecord-jdbc-adapter@ are installed as its dependencies and it solved my precedent issue as expected.

I still have encoding issue translation and every server I try, it must be an environment issue unrelated to DB since every string fetched from DB is fine but I've no clue which, will try to have a look when I get some time.

I'm running it in Windows BTW.

Jean-Philippe Lang wrote:

I get this on db:migrate:

[...]

I think this is a misconfiguration issue and that every account should have read access to these sys* tables.

Etienne Massip wrote:

@gem "activerecord-jdbc-mssql-adapter", "~> 1.2.6" :platforms => :jruby@ is enough since @jdbc-jtds@ and @activerecord-jdbc-adapter@ are installed as its dependencies and it solved my precedent issue as expected.

jdbc-jtds after 1.2.7 is not Java 6 compatible, that's why I used this version.

I still have encoding issue translation and every server I try, it must be an environment issue unrelated to DB since every string fetched from DB is fine but I've no clue which, will try to have a look when I get some time.

Try to set JRUBY_OPTION to -J-Dfile.encoding = UTF8

I think this is a misconfiguration issue and that every account should have read access to these sys* tables.

rake db:migrate runs just fine with ruby1.9+tiny_tds and the same database account

Jean-Philippe Lang wrote:

Try to set JRUBY_OPTION to -J-Dfile.encoding = UTF8

We have a winner!

I think this is a misconfiguration issue and that every account should have read access to these sys* tables.

rake db:migrate runs just fine with ruby1.9+tiny_tds and the same database account

Right, I migrated a blank DB and got the same error as yours.

FWIW I dug into these errors, eventually fixed all of them and went through the whole migration process but these are nasty bugs in AR-JDBC MSSQL adapter and there will be some time until Redmine will be able to work with JRuby and MSSQL.

related_issues

relates,Closed,12693,MSSQL column quotations

relates,Closed,12694,Dates in issue journal

relates,Closed,12692,MS SQL DB migration

relates,Closed,11002,Duplicate field in query breaks Roadmap view

relates,Closed,5822,Problem with Gantt and Calendar with SQL Server 2005

relates,Closed,13091,Task: Document MSSQL Support

履歴

#1 - 2022/05/10 17:14 - Admin Redmine

- カテゴリ を Database_21 にセット
- 対象バージョン を 2.3.0_60 にセット