

## Redmine - Defect #10930

Feature # 12713 (Closed): Microsoft SQLServer support

### rake redmine:load\_default\_data error in 2.0 with SQLServer

2012-05-15 18:04 - Jason Trahan

<b>Status:</b>	Closed	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Etienne Massip	<b>% Done:</b>	0%
<b>Category:</b>	Database	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	2.3.0	<b>Affected version:</b>	2.0.0
<b>Resolution:</b>	Fixed		
<b>Description</b>			
<pre>C:\Ruby\Redmine&gt;rake redmine:load_default_data --trace - Invoke redmine:load_default_data (first_time) - Invoke environment (first_time) - Execute environment - Execute redmine:load_default_data  Select language: ar, bg, bs, ca, cs, da, de, el, en, en-GB, es, et, eu, fa, fi, fr, gl, he, hr, hu, id, it, ja, ko, lt, lv, mk, mn, nl, no, pl, pt, pt-BR, ro, r u, sk, sl, sq, sr, sr-YU, sv, th, tr, uk, vi, zh, zh-TW [en]===== rake aborted! can't convert ActiveRecord::StatementInvalid into String C:/Ruby/Redmine/lib/tasks/load_default_data.rake:31:in `+' C:/Ruby/Redmine/lib/tasks/load_default_data.rake:31:in `rescue in block (2 levels) in &lt;top (required)&gt;' C:/Ruby/Redmine/lib/tasks/load_default_data.rake:25:in `block (2 levels) in &lt;top (required)&gt;' C:/Ruby/193/lib/ruby/1.9.1/rake/task.rb:205:in `call' C:/Ruby/193/lib/ruby/1.9.1/rake/task.rb:205:in `block in execute' C:/Ruby/193/lib/ruby/1.9.1/rake/task.rb:200:in `each' C:/Ruby/193/lib/ruby/1.9.1/rake/task.rb:200:in `execute' C:/Ruby/193/lib/ruby/1.9.1/rake/task.rb:158:in `block in invoke_with_call_chain' C:/Ruby/193/lib/ruby/1.9.1/monitor.rb:211:in `mon_synchronize' C:/Ruby/193/lib/ruby/1.9.1/rake/task.rb:151:in `invoke_with_call_chain' C:/Ruby/193/lib/ruby/1.9.1/rake/task.rb:144:in `invoke' C:/Ruby/193/lib/ruby/1.9.1/rake/application.rb:116:in `invoke_task' C:/Ruby/193/lib/ruby/1.9.1/rake/application.rb:94:in `block (2 levels) in top_level' C:/Ruby/193/lib/ruby/1.9.1/rake/application.rb:94:in `each' C:/Ruby/193/lib/ruby/1.9.1/rake/application.rb:94:in `block in top_level' C:/Ruby/193/lib/ruby/1.9.1/rake/application.rb:133:in `standard_exception_handling' C:/Ruby/193/lib/ruby/1.9.1/rake/application.rb:88:in `top_level' C:/Ruby/193/lib/ruby/1.9.1/rake/application.rb:66:in `block in run' C:/Ruby/193/lib/ruby/1.9.1/rake/application.rb:133:in `standard_exception_handling' C:/Ruby/193/lib/ruby/1.9.1/rake/application.rb:63:in `run' C:/Ruby/193/bin/rake:32:in `&lt;main&gt;' Tasks: TOP =&gt; redmine:load_default_data</pre>			

## History

#1 - 2012-05-15 18:15 - Jason Trahan

- Status changed from New to Resolved

Actually this looks like a problem with <https://github.com/rails-sqlserver/activerecord-sqlserver-adapter>. I will open a ticket with them.

**#2 - 2012-05-15 18:29 - Etienne Massip**

- Category set to Database
- Status changed from Resolved to Closed
- Resolution set to Invalid

BTW, SQL Server backend is not supported for the time being.

**#3 - 2012-05-15 18:54 - Jason Trahan**

Yes I am aware. It has been working good up until 2.0 but I think the adapter has a bug in it so I will open a case with them.

**#4 - 2012-05-15 19:06 - Etienne Massip**

Jason Trahan wrote:

| *Yes I am aware. It has been working good up until 2.0 but I think the adapter has a bug in it so I will open a case with them.*

I'm curious about the amount of work you had to go before having Redmine fully working with SQL server, if you don't mind?

**#5 - 2012-05-15 19:10 - Jason Trahan**

Virtually none. I had to manually change 1 thing in the database due to a bug in the adapter not converting an int to an nvarchar, but for redmine itself i didn't have to make any changes.

However something with the adapter is generating this sql statement which is invalid because the order by has the same column twice sorting by both directions which is invalid.

```
EXEC sp_executesql N'SELECT TOP (1) [enumerations].* FROM [enumerations] WHERE (type = "DocumentCategory") ORDER BY
enumerations.position ASC, position DESC'
```

**#6 - 2012-05-15 19:25 - Etienne Massip**

Should be avoidable by tweaking source:/tags/2.0.0/app/models/enumeration.rb#L21 and source:/tags/2.0.0/app/models/enumeration.rb#L27.

**#7 - 2012-05-15 19:29 - Jason Trahan**

Commenting out those 2 lines does seem to help with the data load. I will put them back in and see how normal operation is.

**#8 - 2012-05-15 19:33 - Etienne Massip**

Actually I'm not even sure that the act\_as\_tree is useful??

Right now I don't get the purpose of r2947.

**#9 - 2012-05-15 19:38 - Jason Trahan**

Actually that `act_as_tree` seems to be the culprit. If I comment out that line instead of the `default_scope` it works.

**#10 - 2012-05-15 19:41 - Jason Trahan**

correction. once I truncated the tables and tried again it still errored out. Weird thing is that the `default_scope` has existed since 2009 and has worked so I am thinking some other change was made that is causing the adapter to generate both order bys.

**#11 - 2012-05-15 19:44 - Jason Trahan**

if I remove `default_scope` then the sql statement comes out to this below which is correct.

```
EXEC sp_executesql N'SELECT TOP (1) [enumerations].* FROM [enumerations] WHERE (type = "DocumentCategory") ORDER BY position DESC'
```

**#12 - 2012-05-15 20:01 - Jason Trahan**

- Status changed from *Closed* to *Reopened*
- Assignee set to *Etienne Massip*

OK I was able to get the data loaded and I put the `default_scope` back in. So far it is working like the previous versions so I'm not sure what during the load defaults is causing the issue. I guess I can leave it in your hands.

**#13 - 2012-05-17 11:58 - Jason Trahan**

Opened a ticket here too to see if they say if this is an adapter issue or not.

<https://github.com/rails-sqlserver/activerecord-sqlserver-adapter/issues/203>

**#14 - 2012-05-17 13:09 - Jason Trahan**

Based on this comment. <https://github.com/rails-sqlserver/activerecord-sqlserver-adapter/issues/203#issuecomment-5761927>

I modified `subversion.rb` and changed

```
def latest_changesets(path, rev, limit=10)
  revisions = scm.revisions(path, rev, nil, :limit => limit)
  revisions ? changesets.find_all_by_revision(revisions.collect(&:identifier), :order => "committed_on DESC", :include => :user) : []
end
```

to

```
def latest_changesets(path, rev, limit=10)
  revisions = scm.revisions(path, rev, nil, :limit => limit)
  revisions ? changesets.find_all_by_revision(revisions.collect(&:identifier), :order => "#{Changeset.table_name}.committed_on DESC",
:include => :user) : []
end
```

I imagine this will need to be done in various places to make it more compliant with mssql. I'm still not sure why this became a problem when I upgraded to 2.0.0 and the latest sql adapter because in 1.4.x it was working without any changes.

**#15 - 2012-05-23 21:06 - Rick Mason**

Etienne Massip wrote:

*I'm curious about the amount of work you had to go before having Redmine fully working with SQL server, if you don't mind?*

I inherited a Redmine 1.0.4 database from another developer. He said he'd had issues getting dates to work, though that was with ODBC and the description of TinyTDS sounds like it may have solved that. He never got the Gantt view working, but it does now which backs that up.

Aside from this sorting issue, the only other one I've encountered is that any attempt to sort or group by a custom field will fail. Filtering is fine. The problem is that the SQL includes a LIMIT clause which MSSQL doesn't recognise. Maybe that's the driver too? Our workaround is to point Excel at the database to get those queries.

**#16 - 2012-05-23 21:09 - Jason Trahan**

A simple fix to that is to open a ticket with the guy who writes the adapter and ask him to convert the limit statements to top statements. He does other conversions now with the adapter.

**#17 - 2012-05-23 21:13 - Rick Mason**

OK, I'll get some evidence together tomorrow and open a ticket. Thanks Jason.

**#18 - 2013-01-12 08:34 - Jean-Philippe Lang**

- Subject changed from *rake redmine:load\_default\_data error in 2.0* to *rake redmine:load\_default\_data error in 2.0 with SQLServer*
- Status changed from *Reopened* to *Closed*
- Target version set to *2.3.0*
- Resolution changed from *Invalid* to *Fixed*

This was fixed in r11100 as part of SQLServer support for 2.3.0 (#12713).

**#19 - 2013-03-04 08:06 - Etienne Massip**

- Priority changed from *Urgent* to *Normal*
- Parent task set to *#12713*