Unable to autoload constant Version.table_name if gems uses Version class

2020-03-29 08:59 - Alexander Meindl

Status: New
Priority: Normal
Assignee:
Category: Gems support
Target version:

Description

Introduced with r18780 Version.table_name is not available, if a gem package (like attr_encrypted) is used with a class named "Version".

At the moment this problem does not exist with Gem packages from Redmine itself, but gems from plugins.

Existing tests are broken, e.g.

```
bin/rails test/unit/project_test.rb -n test_move_a_root_project_to_a_project
```

# Running:

```
E
```

Error:

```
ProjectTest#test_move_a_root_project_to_a_project:
NoMethodError: undefined method `table_name' for AttrEncrypted::Version::Module
  app/models/issue.rb:1739:in `update_versions'
  app/models/issue.rb:1385:in `update_versions_from_hierarchy_change'
  app/models/project.rb:935:in `update_versions_from_hierarchy_change'
  app/models/project.rb:424:in `set_parent!'
  test/unit/project_test.rb:316:in `test_move_a_root_project_to_a_project'
```

Similar problem was #23269. A better description of the problem with examples are here:


I added a patch for a more solid solution to prevent the problem. It would be great to have this change in Redmine.

I added another version of patch, which uses Rails 5 where.not and Arel::Table. With this solution, no more table name specification is required.

Both solutions work, but second one is more cleaner - but more refactoring.

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>full_qualified_version_class.patch</td>
<td>858 Bytes</td>
<td>2020-03-29</td>
<td>Alexander Meindl</td>
</tr>
<tr>
<td>full_qualified_version_class_version2.patch</td>
<td>1.46 KB</td>
<td>2020-03-29</td>
<td>Alexander Meindl</td>
</tr>
</tbody>
</table>