

## Redmine - Defect #31116

### Database migrations don't run correctly for plugins when specifying the `VERSION` env variable

2019-03-29 00:06 - Anonymous

<b>Status:</b>	Closed	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Go MAEDA	<b>% Done:</b>	0%
<b>Category:</b>	Plugin API	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	5.1.0	<b>Affected version:</b>	4.0.2
<b>Resolution:</b>	Fixed		
<b>Description</b>			
tl;dr: When comparing versions, the current_version is set to the core's latest migration instead of the plugin's latest.			
Steps to reproduce:			
1. Install any plugin with migrations, for example:			
<pre>cd redmine/plugins git clone https://github.com/ixti/redmine_tags.git  bundle install rake redmine:plugins:migrate NAME=redmine_tags</pre>			
2. Undo the last migration of that plugin:			
<pre>rake redmine:plugins:migrate NAME=redmine_tags VERSION=20151209153801</pre>			
3. Migrate to the latest version again, but specifying the version this time:			
<pre>rake redmine:plugins:migrate NAME=redmine_tags VERSION=20180122193833</pre>			
Expected behaviour: the 20180122193833 migration is executed.			
Actual behaviour: the 20180122193833 is not executed.			
This is because redmine core's latest migration is 20180923091603 which is greather that the plugin's 20180122193833.			
The attached fix defines a `migrate` method for the `Redmine::Plugin::MigrationContext` class that makes sure to pass along the current plugin's latest migration. Using the default migrate method defined in activerecord executes a `current_version` that looks for the most current migration in the list of all migrations (core and plugins).			
Note: I have not run any tests with this patch.			

#### Associated revisions

##### Revision 21933 - 2022-10-30 07:26 - Go MAEDA

Database migrations don't run correctly for plugins when specifying the `VERSION` env variable (#31116).

Contributed by crypto gopher.

#### History

##### #1 - 2019-03-30 11:28 - Anonymous

Test run available here: <https://gitlab.com/sdwolfz/redmine/-/jobs/187237412>

Although... I am not sure how to add a test that checks for this exact error.

##### #2 - 2019-03-31 07:06 - Go MAEDA

- Category set to Plugin API

- Target version set to Candidate for next major release

### #3 - 2019-04-17 17:49 - Martijn Vernooij

- File redmine-4.0.2.plugin-upgrade.patch added

I noticed the same issue (with 4.0.2 though) and 'fixed' it another way.

### #4 - 2022-09-19 02:30 - crypto gopher

- File MigrationContext-current\_version.patch added

The problem stems from fact, that Redmine::Plugin::MigrationContext does not override current\_version() method. So whenever ActiveRecord::MigrationContext.migrate() is executed, it calls ActiveRecord::MigrationContext.current\_version() which does not differentiate between Redmine's migrations and plugin's migrations - yielding erroneous result.

The most straightforward change is to override ActiveRecord::MigrationContext.migrate() by using Redmine::Plugin::Migrator.current\_version() which already provides required functionality:

```
class MigrationContext < ActiveRecord::MigrationContext
  ...

  def current_version
    Migrator.current_version
  end
end
```

### #5 - 2022-09-21 05:29 - Go MAEDA

- Status changed from New to Confirmed

### #6 - 2022-09-21 06:52 - Go MAEDA

crypto gopher wrote:

The most straightforward change is to override ActiveRecord::MigrationContext.migrate() by using Redmine::Plugin::Migrator.current\_version() which already provides required functionality:

Thank you for posting the patch.

I wrote a test for your patch as follows. Do you think the following code tests the right way?

```
diff --git a/test/unit/lib/redmine/plugin_test.rb b/test/unit/lib/redmine/plugin_test.rb
index a5a1b2aa3..24c06240c 100644
--- a/test/unit/lib/redmine/plugin_test.rb
+++ b/test/unit/lib/redmine/plugin_test.rb
@@ -217,4 +217,17 @@ class Redmine::PluginTest < ActiveSupport::TestCase
    assert Redmine::Plugin.migrate('foo_plugin')
  end
+
+  def test_migration_context_should_override_current_version
+    plugin = @klass.register :foo_plugin do
+      name 'Foo plugin'
+      version '0.0.1'
+    end
+    migration_dir = File.join(@klass.directory, 'db', 'migrate')
+
+    Redmine::Plugin::Migrator.current_plugin = plugin
+    context = Redmine::Plugin::MigrationContext.new(migration_dir, ::ActiveRecord::Base.connection.schema_migration)
+    # current_version should be zero because Foo plugin has no migration
+    assert_equal 0, context.current_version
+  end
end
```

### #7 - 2022-10-13 06:13 - Go MAEDA

- Target version changed from Candidate for next major release to 5.1.0

Setting the target version to 5.1.0.

## #8 - 2022-10-30 07:26 - Go MAEDA

- Status changed from Confirmed to Closed
- Assignee set to Go MAEDA
- Resolution set to Fixed

Committed the patch. Thank you for your contribution.

### Files

0001-Fixes-plugin-migration-with-specific-version.patch	1.75 KB	2019-03-28	Anonymous
redmine-4.0.2.plugin-upgrade.patch	1.47 KB	2019-04-17	Martijn Vernooij
MigrationContext-current_version.patch	399 Bytes	2022-09-19	crypto gopher