

## Redmine - Defect #35199

### HookTest resets all listeners - core tests run inconsistently with plugins

2021-05-03 00:57 - Martin Cizek

|  |                                  |
|--|----------------------------------|
| <b>Status:</b> New   | <b>Start date:</b>               |
| <b>Priority:</b> Normal  | <b>Due date:</b>                 |
| <b>Assignee:</b>   | <b>% Done:</b> 0%                |
| <b>Category:</b> Plugin API  | <b>Estimated time:</b> 0.00 hour |
| <b>Target version:</b>   | <b>Affected version:</b>         |
| <b>Resolution:</b>   |                                  |
| <b>Description</b>   |                                  |
| <p>When a test runner is invoked and the tests include either <code>test/integration/lib/redmine/hook_test.rb</code> or <code>test/unit/lib/redmine/hook_test.rb</code>, there is a side effect that all hooks classes / listeners are cleared. It is probably related to hook listeners being stored in class variables.</p> <p>This may affect other tests, but most importantly, it makes it difficult to verify redmine core integrity with plugins in place. While plugins' patches are still applied, the listeners are not guaranteed and the results differ between e.g. <code>rake test</code> and <code>rake test:functionals</code>.</p> <p>More info about the context below.</p> <p>Although this might not be necessarily a defect - maybe core tests are not supposed to run with plugins - the consequences of plugin developers not running the core tests are terrible and affect the reputation of whole Redmine ecosystem.</p> <p>What we've observed:</p> <ul style="list-style-type: none"><li>- We have run Redmine core tests with plugins we use.</li><li>- It's even difficult to find a plugin that doesn't break anything in the core.</li><li>- A logical objection would be "plugins modify core functionality, so it's normal that core tests are not passing". But this turned out NOT to be the case. In fact:</li></ul> <ul style="list-style-type: none"><li>- Around 50% of the errors and failures are that the implementation is not robust enough to withstand testing.</li><li>- And the other 50% are user-reproducible bugs, which would have been discovered if core tests had been run.</li><li>- In the end, we have not observed any error or failure caused by intentional plugin behavior.</li></ul> <p>Based on your feedback, I can provide specific examples and will be happy to update <a href="#">[[plugin_tutorial]]</a> to encourage core testing.</p> |                                  |

#### History

#1 - 2021-05-10 21:26 - Martin Cizek

- File `plugin_test_support.sh` added

We currently use the functions in the attached script to work around the issue.

At the moment, the only way around is to temporarily rename the tests with plugin-breaking side effects. In Rails 6.2, there is ``DEFAULT_TEST_EXCLUDE`` env that can be used instead of renaming files. But avoiding the side effect completely would still be the very very best. :)

#### Files

|                        |           |            |              |
|------------------------|-----------|------------|--------------|
| plugin_test_support.sh | 849 Bytes | 2021-05-10 | Martin Cizek |
|------------------------|-----------|------------|--------------|