

Redmine - Defect #13654

Can't set parent issue when issue relations among child issues are present

2013-04-01 06:57 - daehong kim

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Jean-Philippe Lang	% Done:	0%
Category:	Issues	Estimated time:	0.00 hour
Target version:	3.3.0	Affected version:	2.3.0
Resolution:	Fixed		
Description			
<p>#13586 solved the relation between subtasks. so subtasks has no relations can make new relation now. but try to move tasks have relation under a task, "Failed to save 1 issue(s) on 2 selected: #xxx." is occurred.(only first issue is moved) and try to move #xxx under parent issue, "Parent task is invalid" is occurred.</p> <p>after remove the relation, it's OK.</p> <p>I tested these things under r11692</p> <p>Thanks.</p> <p>About your application's environment Ruby version 1.9.2 (x86_64-linux) RubyGems version 1.8.24 Rack version 1.4 Rails version 3.2.13 Active Record version 3.2.13 Action Pack version 3.2.13 Active Resource version 3.2.13 Action Mailer version 3.2.13 Active Support version 3.2.13 Middleware Rack::Cache, ActionDispatch::Static, Rack::Lock, #<ActiveSupport::Cache::Strategy::LocalCache::Middleware:0x00000002da4290>, Rack::Runtime, Rack::MethodOverride, ActionDispatch::RequestId, Rails::Rack::Logger, ActionDispatch::ShowExceptions, ActionDispatch::DebugExceptions, ActionDispatch::RemoteIp, ActionDispatch::Callbacks, ActiveRecord::ConnectionAdapters::ConnectionManagement, ActiveRecord::QueryCache, ActionDispatch::Cookies, ActionDispatch::Session::CookieStore, ActionDispatch::Flash, ActionDispatch::ParamsParser, ActionDispatch::Head, Rack::ConditionalGet, Rack::ETag, ActionDispatch::BestStandardsSupport, OpenIdAuthentication Application root /data/redmine Environment production Database adapter mysql2 Database schema version 20130217094251</p>			
Related issues:			
Related to Redmine - Defect #13586: Circular loop testing prevents precedes/f...		Closed	
Related to Redmine - Defect #14015: Ruby hangs when adding a subtask		Closed	
Related to Redmine - Defect #17580: After copying a task, setting the parent ...		Closed	
Related to Redmine - Defect #8628: "Related to" reference may yield circular ...		Closed	2011-06-16
Has duplicate Redmine - Defect #15599: tickets that are related to each other...		Closed	
Has duplicate Redmine - Defect #17845: Assigning parent task fails when a rel...		Closed	
Has duplicate Redmine - Defect #18403: Can't change parent task if an issue h...		Closed	
Has duplicate Redmine - Defect #19406: Cannot set multiple "tasks with preced...		Closed	
Has duplicate Redmine - Defect #20750: linking a copied issue to a superordin...		Closed	
Has duplicate Redmine - Defect #15320: Changing Child's Parent Ticket Field T...		Closed	
Has duplicate Redmine - Defect #19627: Cannot set task.Parent = Grandparent		Closed	
Has duplicate Redmine - Defect #21758: Cannot specify parent task if has link...		Closed	
Has duplicate Redmine - Defect #21558: The issue parent has in most caes to b...		Closed	

Has duplicate Redmine - Defect #22774: "Parent task is invalid" message when ...	Closed
Has duplicate Redmine - Defect #25137: Assigning a parent in issue properties...	Closed

Associated revisions

Revision 15056 - 2016-01-10 16:12 - Jean-Philippe Lang

Can't set parent issue when issue relations among child issues are present (#13654).

History

#1 - 2013-04-04 11:22 - Toshi MARUYAMA

- Description updated

#2 - 2013-04-12 13:15 - Stanislav German-Evtushenko

Same to me

#3 - 2013-04-13 09:42 - Toshi MARUYAMA

- Subject changed from *can't make connected issues into subtasks* to *can't make related issues into subtasks*

#4 - 2013-04-19 22:22 - Mauro Bender

Same here. I would like to have this feature, it seems really useful for me.

#5 - 2013-05-06 18:05 - Ralf S.

Comment:

- +1 Same for us, this was working well >2.3.0
- we updated from V2.3.0 to V2.3.1 to solve this problem (and [#13586](#))
- moving subtasks to another main task is a common usecase for us

Workaround Details:

- in V2.3.0 it was possible to workaround by 1. clearing the parent-task and then setting the new parent task
- but V2.3.1 seems to make the bug more serious, the workaround doesn't work any more (always "Parent task is invalid")

#6 - 2013-05-10 13:36 - Mischa The Evil

- Subject changed from *can't make related issues into subtasks* to *Can't set parent issue when issue relations among child issues are present*

The problem seems to come up when issue relations are in place *while* trying to establish a parent-child construction by changing the parent issue field value of the child-to-be issue. If the parent-child construction is set first, issue relations among child issues can be configured flawlessly. Note: this only applies to issue relations among child-issues; relations between the parent and its child(s) is not supported (and has never been supported).

#7 - 2013-06-10 13:15 - Jonathan Tee

with patch from [#14015](#) it is possible to set parent issue again, but icons are not displayed correctly, see <http://www.redmine.org/issues/14015#note-7>

#8 - 2013-06-19 16:41 - Jonathan Tee

forget my notes about missing icons. Now i've learned that icons are only displayed if the parent issue is above the issue. So if i sort on 'parent id' my parent issue is the first and all my sub issues are displayed with icons :-)

#9 - 2013-09-02 07:37 - Ilyas Makashev

same here. For examlpe, I have tasks 1, 2, 3. 2 is child of 1, 2 is related to 3. When I try to move 3 to 1 as child, it says "Parent task is invalid". After removing relationship between 2 and 3, it works.

#10 - 2013-09-03 06:35 - Ilyas Makashev

the problem seems to appear in line 579 of issue.rb. As I understand it's not a bug, but feature.

There is a function `all_dependent_issues`, which constructs array with all child's of issue we move and related to them issues. If the target parent issue appears in that array by some way (I think through related issues), error is thrown.

If we do not add related issues to `all_dependent_issues`, everything works just fine (to do that comment lines 923-936 in issue.rb).

I can missing something, but why we add related issues to `all_dependent_issues`? I don't see any logic of that.

#11 - 2013-09-17 21:06 - Felix Schäfer

We're ([Planio](#)) also seeing this problem, it's very strange because you can remove the relationship between the future siblings, make 2 issues siblings and re-add the relationship between the 2 siblings afterward.

Is there any way we can help with this or speed up things in any way?

#12 - 2013-09-18 21:53 - Bruno Medeiros

Same problem here. A blocks B, and if I try to set A and B parent to C, I got an error: "Parent task is invalid"
As Felix said, the only workaround is to remove the issues relations and re-add them, which could be very painful.

#13 - 2013-09-19 14:56 - Etienne Massip

- Status changed from New to Confirmed
- Target version set to Candidate for next minor release

Confirmed, happens when setting a parent

1. to an issue which has a relation to an issue which has a parent
2. because of 1. and the fact that the parent issue is set to each one by one, to issues which already have mutual relations

#14 - 2013-10-21 14:41 - Felix Schäfer

I don't understand the `Issue#all_dependent_issues` method or where it's used well enough yet, so this has to be taken to a grain of salt, but removing the line source/branches/2.3-stable/app/models/issue.rb@12213#L903 solves this problem. Be aware though: This is untested for everything else, so I can't say what other repercussions this has on other functions of Redmine.

#15 - 2013-10-25 07:21 - Toru Haraguchi

+1
Here same issue arises which did not while we were using 2.2.x.
If `all_dependent_issues(except)` is universal as intended, `except` argument may be not provided correctly at usage. And probably, `all_dependent_issues(except)` should be called multiple times with different `except` s.

#16 - 2013-11-29 06:29 - Ilyas Makashev

`Issue#all_dependent_issues` seems to be too complicated and unreasonable (and also it slows down the performance when creating or updating issues). This algorithm defines dependent issues term and says that it consists of parent, child and related issues of particular issue. In fact we shouldn't consider related issues as dependent issues.

Let's say we want B to make as child to A. The only checking we should do is to ensure that B doesn't already belong to the children tree of A and B doesn't have A in its own children tree. In fact this point is very different to issue relationships.

So I am definitely sure that we should remove from `Issue#all_dependent_issues` algorithm the lines deal with issue relationships. These lines are 926-940 in <http://www.redmine.org/projects/redmine/repository/revisions/12213/entry/branches/2.3-stable/app/models/issue.rb>. As I tested it seems to break nothing.

Also we may need to rewrite a bit this function because there will be no use of some logic and constants anymore.

I you need any help I can prepare patch.

#17 - 2014-01-02 10:36 - Adrien Crivelli

Also confirm the behavior, removing relationship, adding parent, and re-creating relationship is a possible workaround for the time being. But that may send lots of undesired email notifications.

#18 - 2014-01-23 11:00 - Pierre Pretorius

I'm running Redmine 2.3.1.stable. We experience this issue as well.

This workaround doesn't work

- 1) Remove any existing parent then save.
- 2) Add the new parent.

This workaround works

- 1) Remove any existing parent and related tasks.
- 2) Add the new parent and related tasks removed in step 1.

Additionally last week our server went into an infinite loop. Ruby processes maxed at 100% cpu and doesn't respond to new requests. This is probably related issue [#14015](#) or [#13586](#) which has been fixed in 2.3.2.

#19 - 2014-03-14 08:35 - Maxime Vez

With Redmine 2.4, I also experience this bug. Workaround : first remove the related tasks, put them under the same parent, then add again the relation between the brother tasks.

#20 - 2014-04-25 13:42 - Toshi MARUYAMA

- Has duplicate Defect #15599: tickets that are related to each other block a declaration of parent task for them (error: invalid task) added

#21 - 2014-04-25 15:06 - Diego Tiemann

Same here, in Redmine Version 2.5. I would like to have this feature, it seems really useful for me.

#22 - 2014-06-02 06:05 - Toshi MARUYAMA

- Related to Defect #15320: Changing Child's Parent Ticket Field To Parent's Parent fails. added

#23 - 2014-07-03 05:41 - Yas UCHIDA

With Redmine 2.5.1, I also experience this bug. I would like to have this feature, it seems really useful for me.

#24 - 2014-07-07 22:15 - William Roush

Pierre Pretorius wrote:

- 1) Remove any existing parent and related tasks.

Ouch, ever since we got history on adding/removing related tasks, having to do things like this to fix relationship issues gets *very* noisy. :(

#25 - 2014-09-26 11:10 - Etienne Rossignon

Those defects [#13654](#) and [#15320](#) are really annoying and prevents us to rearrange a hierarchie of issues. (we are using 2.4.1)

As stated above I am not sure that the piece of code that prevent creation of cycles should care about "linked issues" but only about parent-child relationships.

#26 - 2014-10-01 12:10 - Ismael Barros²

This issue is still reproducible in 2.5.2.stable; it's a pretty big show-stopper.

#27 - 2014-10-10 09:52 - Toshi MARUYAMA

- Has duplicate Defect #17845: Assigning parent task fails when a related issue is yet a sub issue of the new parent task added

#28 - 2014-10-13 11:35 - Laurent Girard

+1

#29 - 2014-10-14 22:37 - Deoren Moor

+1

I got bit by this again today. My workaround for similar situations when Redmine won't let me relate ticket A to ticket B is to relate B to A. However since I was trying to change parent tickets I had to nuke all relations, and then "reparent" the ticket (as others have already said). I did not bother trying to add the relations back when I was done.

#30 - 2014-10-21 12:40 - Robert Korulczyk

+1

Confirmed in Redmine 2.5.2.stable.13466. Really annoying.

#31 - 2014-10-27 13:19 - Dzianis Frydliand

+1

Confirmed in Redmine 2.5.2

Please fix it!

#32 - 2014-10-29 19:57 - Mikhail Barg

+1!

#33 - 2014-11-19 11:00 - Roger Griffiths

Etienne Rossignon wrote:

Those defects [#13654](#) and [#15320](#) are really annoying and prevents us to rearrange a hierarchie of issues. (we are using 2.4.1)

As stated above I am not sure that the piece of code that prevent creation of cycles should care about "linked issues" but only about parent-child relationships.

I agree that this should be resolved but some types of "linked issue" relationships do matter where cycles are concerned:-

- Duplicates/Duplicated By
- Blocks/Blocked By
- Precedes/Follows
- Copied To/Copied From

Only the "Related To" linkage (which is symmetrical) should notionally ignore cycles and arguably the cycles should be managed on a per relationship type basis.

Parent-Child relationships are a traditional WBS structure feature and the Blocks/Blocked By and Precedes/Follows relationships are traditional scheduling relationships (the two should not interfere with each other).

#34 - 2014-11-25 20:55 - @ go2null

Roger Griffiths wrote:

I agree that this should be resolved but some types of "linked issue" relationships do matter where cycles are concerned:-

- Duplicates/Duplicated By
- Blocks/Blocked By
- Precedes/Follows
- Copied To/Copied From

Only the "Related To" linkage (which is symmetrical) should notionally ignore cycles and arguably the cycles should be managed on a per relationship type basis.

Parent-Child relationships are a traditional WBS structure feature and the Blocks/Blocked By and Precedes/Follows relationships are traditional scheduling relationships (the two should not interfere with each other).

Any reason why Duplicates/Duplicated By and Copied To/Copied From should be considered as forming a hierarchy?

#35 - 2015-01-20 05:07 - Mischa The Evil

@ go2null wrote:

Roger Griffiths wrote:

...

Any reason why Duplicates/Duplicated By and Copied To/Copied From should be considered as forming a hierarchy?

They are both tied to specific issue actions:

- Duplicates/Duplicated By relation: enforces the auto-closure of duplicates when the duplicated issue is being closed.
- Copied To/Copied From: this relation is *always added automatically* when copying an issue (until [r13668, 3.0.0](#); nevertheless, since issue copy is implemented as "old issue [Copy-action] issue new form + pre-filled properties (which can be manually changed) > new issue" issue copied to will logically always *follow out of* issue copied from (not taking manual relation changes into account here)).

#36 - 2015-01-21 06:59 - Mischa The Evil

- Has duplicate Defect #18403: Can't change parent task if an issue has some relations added

#37 - 2015-02-19 15:20 - Alexander Morozov

+1
Fix please!

#38 - 2015-02-20 21:39 - Omer Arslan

+1 - will be very useful

#39 - 2015-03-28 04:58 - Go MAEDA

- Has duplicate Defect #19406: Cannot set multiple "tasks with precedes/follows relationship" to *all* become child-tasks of some parent added

#40 - 2015-06-23 19:50 - Dirk Hoffmann

It is frustrating to see that this is still on priority "Normal" after two years of discussions and many duplicate reports.

It even seems that it is a regression (around 2.3.0/2.3.1), so it should receive some significant attention from developers.

#41 - 2015-06-24 07:02 - Anonymous

+1 - This bug is really annoying since two years now...

#42 - 2015-06-24 10:54 - Rafael Gómez García

Florian S. wrote:

+1 - This bug is really annoying since two years now...

#43 - 2015-07-28 14:25 - Jaromír Rys

+1

#44 - 2015-09-16 16:40 - Fernando Hartmann

+1 This bug really disturbs many of my users.

#45 - 2015-10-08 12:10 - Martin Hartung

+1 (because the reason is really unlogic and not recognizable)

#46 - 2015-10-14 04:51 - Toshi MARUYAMA

- Related to Defect #20750: linking a copied issue to a superordinate ticket does not work added

#47 - 2015-10-14 04:51 - Toshi MARUYAMA

- Related to deleted (Defect #20750: linking a copied issue to a superordinate ticket does not work)

#48 - 2015-10-14 04:52 - Toshi MARUYAMA

- Has duplicate Defect #20750: linking a copied issue to a superordinate ticket does not work added

#49 - 2015-10-26 16:22 - Thomas Steinert

+1 with Redmine 2.6.1

#50 - 2015-10-28 18:48 - @ go2null

Mischa The Evil wrote:

@ go2null wrote:

Roger Griffiths wrote:

...

Any reason why Duplicates/Duplicated By and Copied To/Copied From should be considered as forming a hierarchy?

They are both tied to specific issue actions:

- Duplicates/Duplicated By relation: enforces the auto-closure of duplicates when the duplicated issue is being closed.
- Copied To/Copied From: this relation is *always added automatically* when copying an issue (until [r13668, 3.0.0](#); nevertheless, since issue copy is implemented as "old issue [Copy-action] issue new form + pre-filled properties (which can be manually changed) > new issue" issue copied to will logically always *follow out of* issue copied from (not taking manual relation changes into account here)).

Duplicates/Duplicated By: OK.

Copied To/Copied From: I've had users copy an issue to make the copy the parent (as the *copy from* has children already), only to find out that they cannot.

#51 - 2015-11-03 19:47 - Roman Pickl

+1

#52 - 2015-11-05 11:52 - Jean Louis

+1

#53 - 2015-11-10 13:20 - Nils Grimm

+1, confirmed in redmine 3.1.1.

#54 - 2015-11-19 06:12 - Toshi MARUYAMA

- Related to Defect #17580: After copying a task, setting the parent as the original task's parent triggers an error added

#55 - 2015-11-26 17:41 - Kirill Marchuk

Hi all!

I believe that "all_dependent_issues" is not-so-well defined method.

I see that it includes children as dependent, while (as far as I understand) quite the opposite is true: parent task depends on it's children, and not vice versa. This basically follows out of assumption, that you cannot close (or solve) Parent task, until you have all the Children closed (or solved).

Am I wrong ? Can anyone comment on this ?

I would suggest to re-define (and re-implement) this method to return only **really** dependent issues - i.e., issues that cannot be closed or solved until you close or solve the issue in question (at least, concerning "parenthood", but normally it should also make distinction between dependency-neutral relations like "related" and dependency-critical relations like "blocked").

Any objections ? I would be happy to hear what the experts think about this.

Thanks a lot.

#56 - 2015-11-27 06:44 - Kirill Marchuk

I propose to define a method "all_blocking_and_preceding_issues (issue)", that would return a collection of issues, by building a queue of:

- issue in question;
- issues, that have either "blocking" or "preceding" relation with any issue in the queue;
- issues, that are children of any issue in the queue.

(repeat building the queue until we cannot add any issue to the queue)

and the only check, that we have to have while adding a parent P to issue S, should be: make sure that P is not in 'all_blocking_and_preceding_issues (S)' (because that would create "closed dependency loop". All the other dependency checks are not relevant here.

Would be great to hear from the maintainers - is this acceptable approach ? I want to implement patch on this, but would be great to have a support from the experts.

#57 - 2015-11-29 14:12 - Kirill Marchuk

There's unit test in "test/unit/issue_test.rb" called "test_setting_parent_to_a_dependent_issue_should_not_validate"

In this test, issue3 precedes issue1 and issue1 precedes issue2; and then issue2 is attempted to be set as a parent of issue3. The expected behaviour in a test is to FAIL this attempt.

As far I understand, there should be nothing wrong with issue3 preceding issue2 AND being a child of issue2; apparently, the test authors thought different about that.

May I ask what's the logic behind this test ? I see that it was added here (http://www.redmine.org/projects/redmine/repository/revisions/11641/diff/trunk/test/unit/issue_test.rb), although I don't exactly understand a point of test_setting_parent_to_a_dependent_issue_should_not_validate; test_setting_parent_should_not_allow_circular_dependency seems sufficient to mitigate the issue [#8794](#)

Please correct me if I am wrong on this

#58 - 2015-12-05 16:13 - Kirill Marchuk

I want to suggest a patch for review; but the failing "test_setting_parent_to_a_dependent_issue_should_not_validate" test stops me from that.

the very point of my patch (as it should help solve the problem we're discussing here) is to ALLOW (incl. transient) non-blocking relationships between to-be-parent and to-be-child issues. The only case when setting P as parent to issue S should fail, is when P already BLOCKS or PRECEDES S (directly or transiently)

May I include removal or refactoring of test "test_setting_parent_to_a_dependent_issue_should_not_validate" in a patch ? Or what else should I do ?

Any suggestions are welcome.

#59 - 2015-12-06 15:05 - Jean-Philippe Lang

Kirill Marchuk wrote:

May I include removal or refactoring of test "test_setting_parent_to_a_dependent_issue_should_not_validate" in a patch ?

Yes, that would be fine.

#60 - 2015-12-09 18:22 - Kirill Marchuk

- File defect-13654.diff added

Sorry for the delay.

OK, so here's the proposed patch. I've tested the case, described in the Defect, and the problem no longer occurs. Although I definitely did not check all the possible cases.

All the "issue_test.rb" tests are successful with this patch; although I had to change one of them (see above). Also I've added one-line comments for 2 of the test-cases - I think it will be easier for the contributors to see, why is this test here.

Looking forward for your feedback!

#61 - 2015-12-10 06:02 - Kirill Marchuk

Yesterday, when I've proposed a patch, "rake test" was still running (it's all too slow on my machine), so I attached the patch before all the test were run.

At the end of it, it had 1 failure, something like "IssueNestedCollectionSet"

It was too late to investigate, and I will try to try it on master today.

Just FYI

#62 - 2015-12-11 18:33 - Kirill Marchuk

Looks like the same happens with "master" on my machine:

```
1) Failure:
IssueNestedSetConcurrencyTest#test_concurrency [test/unit/issue_nested_set_concurrency_test.rb:45]:
Expected "Mysql2::Error: Deadlock found when trying to get lock; try restarting transaction: SELECT `issues`.`id` FROM `issues` WHERE (root_id IN (SELECT root_id FROM issues WHERE id IN (1031,1028))) ORDER BY `issues`.`id` ASC FOR UPDATE" to be nil.
```

```
2) Failure:
IssueNestedSetConcurrencyTest#test_concurrent_subtasks_creation [test/unit/issue_nested_set_concurrency_test.rb:61]:
Expected "Mysql2::Error: Deadlock found when trying to get lock; try restarting transaction: SELECT `issues`.`id` FROM `issues` WHERE (root_id IN (SELECT root_id FROM issues WHERE id IN (1037))) ORDER BY `issues`.`id` ASC FOR UPDATE" to be nil.
```

2 runs, 3 assertions, 2 failures, 0 errors, 0 skips

I have no ideas what to do with it, but I have a humble hope that this has nothing to do with my changes (as long as it's on master), so I would ask to review my patch and somehow push this process further..

I am pretty sure there're problems with coding style and other things, so I am ready to process this feedback

#63 - 2015-12-17 04:13 - Kirill Marchuk

Hi all

Is there any timeline regarding patch review, or any other feedback ?

Thanks

#64 - 2015-12-30 11:28 - Kirill Marchuk

Happy New Year everyone !

I hope in 2016 the patch will be reviewed.

#65 - 2016-01-05 09:16 - Go MAEDA

Kirill Marchuk's patch works fine for me.

#66 - 2016-01-09 12:19 - Toshi MARUYAMA

Kirill Marchuk wrote:

Looks like the same happens with "master" on my machine:

1) Failure:

IssueNestedSetConcurrencyTest#test_concurrency [test/unit/issue_nested_set_concurrency_test.rb:45]:

Expected "Mysql2::Error: Deadlock found when trying to get lock; try restarting transaction: SELECT `issues`.`id` FROM `issues` WHERE (root_id IN (SELECT root_id FROM issues WHERE id IN (1031,1028))) ORDER BY `issues`.`id` ASC FOR UPDATE" to be nil.

2) Failure:

IssueNestedSetConcurrencyTest#test_concurrent_subtasks_creation [test/unit/issue_nested_set_concurrency_test.rb:61]:

Expected "Mysql2::Error: Deadlock found when trying to get lock; try restarting transaction: SELECT `issues`.`id` FROM `issues` WHERE (root_id IN (SELECT root_id FROM issues WHERE id IN (1037))) ORDER BY `issues`.`id` ASC FOR UPDATE" to be nil.

2 runs, 3 assertions, 2 failures, 0 errors, 0 skips

Due to [#19344](#).

#67 - 2016-01-09 12:21 - Toshi MARUYAMA

- Target version changed from Candidate for next minor release to 3.3.0

#68 - 2016-01-10 16:18 - Jean-Philippe Lang

- Status changed from Confirmed to Closed

- Assignee set to Jean-Philippe Lang

- Resolution set to Fixed

This is fixed by [r15056](#). Tests were added for the use cases reported here.

What differs from the proposed patch is that blocks/blocked by relations are now totally independant of parent/subtasks relations as Redmine does not block the closing of parent issues in any way.

And it was important to keep #test_setting_parent_to_a_dependent_issue_should_not_validate as is for the reasons now explained in comments. It was failing with the proposed change.

Kirill, thanks for your help in fixing this issue.

#69 - 2016-01-13 06:04 - Go MAEDA

- Related to Defect #8628: "Related to" reference may yield circular dependency error message added

#70 - 2016-01-17 02:16 - Go MAEDA

- Related to deleted (Defect #15320: Changing Child's Parent Ticket Field To Parent's Parent fails.)

#71 - 2016-01-17 02:16 - Go MAEDA

- Has duplicate Defect #15320: Changing Child's Parent Ticket Field To Parent's Parent fails. added

#72 - 2016-01-17 03:42 - Go MAEDA

- Has duplicate Defect #19627: Cannot set task.Parent = Grandparent added

#73 - 2016-01-19 07:31 - Go MAEDA

- Has duplicate Defect #21758: Cannot specify parent task if has linked related issue added

#74 - 2016-02-12 04:19 - Toshi MARUYAMA

- Has duplicate Defect #21558: The issue parent has in most cases to be deleted before assigning a new parent. added

#75 - 2016-05-04 14:37 - Thomas Koch

The Problem still exists with Redmine 3.2.0.stable.

#76 - 2016-05-04 14:43 - Matteo Steccolini

Thomas Koch wrote:

The Problem still exists with Redmine 3.2.0.stable.

The target revision appears to be 3.3.0.

#77 - 2016-12-11 08:57 - Go MAEDA

- Has duplicate Defect #22774: "Parent task is invalid" message when setting parent to root issue after setting to a child of the root. added

#78 - 2017-03-04 04:47 - Go MAEDA

- Has duplicate Defect #25137: Assigning a parent in issue properties does not work (Error "Parent task is invalid") although the Parent exists added

Files

defect-13654.diff	5.38 KB	2015-12-09	Kirill Marchuk
-------------------	---------	------------	----------------