

ChiliProject - Bug # 497: When deleting a subtask priority of the parent task does not become changeable again

Status:	Needs more information	Priority:	Normal
Author:	Jan Schulz-Hofen	Category:	Issue tracking
Created:	2011-06-30	Assignee:	
Updated:	2011-08-07	Due date:	
Remote issue URL:	http://www.redmine.org/issues/8716		
Affected version:	master		
Description:	Not sure if it's really also a Chili bug, but I'd say it's likely one, so I just wanted to post it here as well		
	How to reproduce this:		
	# Create a ticket		
	# Create a subtask to this ticket		
	# Set the subtask's priority to "high" or any other value		
	# Try to update the parent task ; priority cannot be changed (this is correct behavior as the parent's priority is calculated)		
	# Now, delete the subtask		
	# Try to update the parent task again ; priority still cannot be changed (this is a bug)		

History

2011-07-01 04:42 pm - Eric Davis

- Category set to Issue tracking

2011-07-16 09:48 am - Madhusudan C.S.

I would like to patch this issue myself. (Although I am new to chiliproject/redmine codebase, I would like to contribute.) I have spent considerable amount of time looking into this issue in the Redmine/Chiliproject code base and I have come to the conclusion that this is actually a bug in Awesome Nested Set. I may be wrong here. So please guide me on how to go about fixing this bug. Here is the sample rails console session as to why I think this is a bug in Awesome Nested Set.

I am just pasting the migration from my irb-history and then the actual console session:

<pre>

```
class CreateCategories < ActiveRecord::Migration
  def self.up
    create_table :categories do |t|
      t.string :name
      t.integer :parent_id
      t.integer :lft
      t.integer :rgt
    end
  end
  def self.down
    drop_table :categories
  end
end
```

and then CreateCategories.up

<pre>

```

ruby-1.8.7-head >> class Category < ActiveRecord::Base
ruby-1.8.7-head?..   acts_as_nested_set
ruby-1.8.7-head?.. end
â†’ [
  [0] "before_move",
  [1] "after_move"
]
ruby-1.8.7-head >> p = Category.new()
â†’ #<Category:0x7f6874fde128> {
  :id => nil,
  :name => nil,
  :parent_id => nil,
  :lft => nil,
  :rgt => nil
}
ruby-1.8.7-head >> p.save
SQL (0.4ms)  SELECT max("categories".rgt) AS max_rgt FROM "categories"
Category Create (12.0ms)  INSERT INTO "categories" ("name", "lft", "parent_id", "rgt") VALUES(NULL, 13, NULL, 14)
â†’ true
ruby-1.8.7-head >> c = Category.new()
â†’ #<Category:0x7f6874fbf2a0> {
  :id => nil,
  :name => nil,
  :parent_id => nil,
  :lft => nil,
  :rgt => nil
}
ruby-1.8.7-head >> c.save
SQL (0.4ms)  SELECT max("categories".rgt) AS max_rgt FROM "categories"
Category Create (0.3ms)  INSERT INTO "categories" ("name", "lft", "parent_id", "rgt") VALUES(NULL, 15, NULL, 16)
â†’ true
ruby-1.8.7-head >> p.leaf?
â†’ true
ruby-1.8.7-head >> c.leaf?
â†’ true
ruby-1.8.7-head >> c.move_to_child_of(p)
Category Load (0.4ms)  SELECT "lft", "rgt", "parent_id" FROM "categories" WHERE ("categories"."id" = 9)
Category Load (0.3ms)  SELECT "lft", "rgt", "parent_id" FROM "categories" WHERE ("categories"."id" = 10)
Category Update (7.5ms)  UPDATE "categories" SET "lft" = CASE WHEN "lft" BETWEEN 14 AND 14 THEN "lft" + 16 - 14 WHEN "lft" BETWEEN 15
AND 16 THEN "lft" + 14 - 15 ELSE "lft" END, "rgt" = CASE WHEN "rgt" BETWEEN 14 AND 14 THEN "rgt" + 16 - 14 WHEN "rgt" BETWEEN 15 AND 16
THEN "rgt" + 14 - 15 ELSE "rgt" END, "parent_id" = CASE WHEN id = 10 THEN 9 ELSE "parent_id" END
Category Load (0.3ms)  SELECT "lft", "rgt", "parent_id" FROM "categories" WHERE ("categories"."id" = 9)
Category Load (0.2ms)  SELECT "lft", "rgt", "parent_id" FROM "categories" WHERE ("categories"."id" = 10)
â†’ []
ruby-1.8.7-head >> p.save
â†’ true
ruby-1.8.7-head >> c.save
â†’ true
ruby-1.8.7-head >> p.leaf?
â†’ false
ruby-1.8.7-head >> c.leaf?
â†’ true
ruby-1.8.7-head >> p.reload

```

```

Category Load (0.4ms)  SELECT * FROM "categories" WHERE ("categories"."id" = 9)
â†’ #<Category:0x7f6874fde128> {
  :id => 9,
  :name => nil,
  :parent_id => nil,
  :lft => 13,
  :rgt => 16
}
ruby-1.8.7-head >> c.reload
Category Load (0.5ms)  SELECT * FROM "categories" WHERE ("categories"."id" = 10)
â†’ #<Category:0x7f6874fbf2a0> {
  :id => 10,
  :name => nil,
  :parent_id => 9,
  :lft => 14,
  :rgt => 15
}
ruby-1.8.7-head >> p.leaf?
â†’ false
ruby-1.8.7-head >> c.leaf?
â†’ true
ruby-1.8.7-head >> c.destroy
Category Destroy (0.4ms)  DELETE FROM "categories" WHERE "id" = 10
â†’ #<Category:0x7f6874fbf2a0> {
  :id => 10,
  :name => nil,
  :parent_id => 9,
  :lft => 14,
  :rgt => 15
}
ruby-1.8.7-head >> c.save
â†’ true
ruby-1.8.7-head >> c.destroyed?
â†’ true
ruby-1.8.7-head >> p.leaf?
â†’ false
ruby-1.8.7-head >> p.save
â†’ true
ruby-1.8.7-head >> p.leaf?
â†’ false
ruby-1.8.7-head >> p.reload
Category Load (0.5ms)  SELECT * FROM "categories" WHERE ("categories"."id" = 9)
â†’ #<Category:0x7f6874fde128> {
  :id => 9,
  :name => nil,
  :parent_id => nil,
  :lft => 13,
  :rgt => 16
}
ruby-1.8.7-head >> p.leaf?
â†’ false
ruby-1.8.7-head >> p.children
Category Load (0.3ms)  SELECT * FROM "categories" WHERE ("categories"."parent_id" = 9) ORDER BY "lft"

```

â†' []
</pre>

the parent (p) of the child (c) returns false when the child is destroyed. And p.leaf? is the method that is used to control the priority enabled/disabled state in the Issue views. So I think this bug needs to be fixed in Awesome Nested Set. Please advise me as to how to go about this.

2011-07-22 03:26 pm - Eric Davis

- *Status changed from Open to Needs more information*

Madhusudan C.S. wrote:

> So I think this bug needs to be fixed in Awesome Nested Set. Please advise me as to how to go about this.

awesome_nested_set is a third party library so you should report the bug there. Once it's fixed we can upgrade our version of it in ChiliProject.

https://github.com/collectiveidea/awesome_nested_set

2011-08-07 03:48 am - Andy Dufilie

I've created a pull request "here": <https://github.com/edavis10/redmine/pull/38> to fix this issue (also posted on the redmine tracker).