


```
select position
from (select id, position from enumerations) as parent
where parent_id = parent.id), 1) WHERE "enumerations"."id" IN (SELECT "enumerations"."id" FROM "enumerations" WHERE "enumerations"."type" IN ('TimeEntryActivity') AND "enumerations"."parent_id" IS NOT NULL ORDER BY "enumerations"."position" ASC)
```

#5 - 2020-08-28 10:17 - shigeo teraoka

Thank you for your analysis.

If SQL is executed 66 times, it will take time to update.
Is it possible to improve it?

#6 - 2020-09-03 08:01 - Mizuki ISHIKAWA

- File 33289.patch added

I am attaching a patch to improve this situation.

Probably [Enumeration#update_position](#) is the code to change the position of the project activity according to the position of the parent activity. Simply adding a project activity should not require repositioning other project activities. This patch changes the position of other project activities only if you sort the activities in Administration > Enumerations.

After applying this patch, the following SQL should not be executed even if you change the settings on the time tracking tab. It should speed up the settings.

```
TimeEntryActivity Update All (800.9 ms)
UPDATE "enumerations" SET position = coalesce((
  select position
  from (select id, position from enumerations) as parent
  where parent_id = parent.id), 1) WHERE "enumerations"."id" IN (SELECT "enumerations"."id" FROM "enumerations" WHERE "enumerations"."type" IN ('TimeEntryActivity') AND "enumerations"."parent_id" IS NOT NULL ORDER BY "enumerations"."position" ASC)
```

#7 - 2020-09-03 10:56 - Go MAEDA

- Subject changed from *I want to improve the update response of the checkbox on the time tracking tab of the setting to Updating time tracking activities in project setting may take too long time*

- Category changed from *Time tracking to Performance*

#8 - 2020-09-16 11:17 - Stoyan Zhekov

I was able to reproduce the problem and confirm, that the patch, provided above, is improving the performance.

First a problem recap and how to reproduce it:

The way TimeEntryActivity works: you create them in 'Administration|Enumerations|Activities' and change them per project in 'Project|Settings|Time Tracing'.

The important thing: **only when something is changed in per project settings, new DB record is created.** So even with a lot of activities and projects, **there will be no visible problem, if there are no changes.**

To create the test setup, a script, provided by Mr. Mizuki Ishikawa (the patch author) was used:

```
70.times do |n|
  TimeEntryActivity.create!(name: "TimeEntryActivity#{n}")
end
100.times do |n|
  project = Project.create!(name: "Project#{n}", identifier: "project-#{n}")
  project.set_parent!(nil)
  project.update_or_create_time_entry_activities(
    TimeEntryActivity.where(project_id: nil).map do |t|
      [
        t.id.to_s, {
          "parent_id" => t.id.to_s,
          "active" => "0",
          "custom_field_values" => {"7" => "0"}
        }
      ]
    end.to_h
  )
end
```

A small **warning**: Apply the patch **before** running the script and then remove it, if needed, for the tests. Without the patch it is taking **very long time** to create the DB records. Which **already is a prove for patch effectiveness**.

The way to introduce a lot of changes is from 'Project|Settings|Time tracking' to first click 'Reset' and then manually change **all the activities**. Then click 'Save' button and check how long it takes to save the changes.

I tested with **100** and with **200** projects. Results from my tests (local PostgreSQL DB, running in Docker container, Redmine-4.1.1):

- no patch, 70 activities, 100 projects: ~35sec
- no patch, 70 activities, 200 projects: ~50sec
- with patch, 70 activities, 200 projects: ~2-3sec

As you can see increasing the number of projects increase the time of saving. So it is possible, as Mr. Shigeo Teraoka reported, with 800 projects this time to be more then 5 min.

And as you can see applying the patch will reduce that time significantly.

#9 - 2020-09-17 01:27 - Mizuki ISHIKAWA

Stoyan Zhekov wrote:

I was able to reproduce the problem and confirm, that the patch, provided above, is improving the performance.

Thank you for checking the patch.

#10 - 2020-09-18 12:33 - Go MAEDA

- *Target version set to 4.2.0*

Setting the target version to 4.2.0.

#11 - 2020-09-22 09:52 - Go MAEDA

- *Status changed from New to Closed*

- *Assignee set to Go MAEDA*

- *Resolution set to Fixed*

Committed the patch. Thank you all for your contribution.

Files

33289.patch	2.5 KB	2020-09-03	Mizuki ISHIKAWA
-------------	--------	------------	-----------------