Redmine - Defect #36897

The minutes part of a time entry is displayed as 60 instead of being carried over

2022-04-05 16:04 - Denis Yurashku

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Go MAEDA	**% Done:**	0%
Category:	Time tracking	**Estimated time:**	0.00 hour
Target version:	6.0.0		
Resolution:	Fixed	**Affected version:**	

Description

Date is formatted wrong at "<PROJECT> – Time entries" ("7:60" instead of "8:00", for example). See screenshot.

This patch fixes it.

Related issues:

Related to Redmine - Feature #23996: Introduce a setting to change the displa...	**Closed**
Related to Redmine - Defect #40914: Fix precision issues in TimeEntry#hours c...	**Closed**
Related to Redmine - Defect #42172: `format_hours` method produces incorrect ...	**Closed**

Associated revisions

Revision 22946 - 2024-08-08 06:24 - Go MAEDA

Fix issue where minutes part of a time entry is displayed as 60 instead of being carried over (#36897).

Patch by Go MAEDA (user:maeda).

History

#1 - 2022-04-05 16:30 - Denis Yurashku

I've encountered this in 4.0.6, but AFAIK this bugged part of code still not fixed in 5.0.0.

#2 - 2022-04-06 15:21 - Holger Just

- Related to Feature #23996: Introduce a setting to change the display format of timespans to HH:MM added

#3 - 2022-04-06 15:23 - Holger Just

- Target version set to Candidate for next minor release

This whole thing can probably be simplified to just

```
def format_hours(hours)
  return "" if hours.blank?

  if Setting.timespan_format == 'minutes'
    "%d:%02d" % (hours * 60).divmod(60)
  else
    l_number("%.2f", hours.to_f)
  end
end
```

#4 - 2022-04-06 16:39 - Denis Yurashku

Probably yes.

But as far as I understood from how hours are displayed in Time entries, they need to be `.round`-ed (not just dropping the fractional part).
I.e. if you've spent 7:59:30 (7.991666666666667 hrs), for human it's 8:00 (we don't count seconds in Time entries). :)
Patch gives "8:00" too.

I think like this.

#5 - 2022-04-06 17:35 - Holger Just

Hmmm, with my version, trailing "seconds" indeed would just be cut off instead of being rounded. Thus, semantically 7:59:01 and 7:59:59 would both be shown as "7:59" which is different from the current behavior.

If we round the minutes, we would again have to deal with the case of 7.99 hours being rounded up to "7:60" however.

Maybe this version could work then:

```
h, m = (hours * 60).divmod(60).map!(&:round)
if m == 60
  h += 1
  m = 0
end
"%d:%02d" % [h, m]
```

The if block could also be shortened to the equivalent one-liner:

```
h, m = h+1, 0 if m == 60
```

In any case, we can be sure that after the first line, m is an Integer between 0 and 60 (inclusive). If it was rounded up to 60, we set to to 0 and increase the hour by one in the if block.

This should then be equivalent to the proposed version in the original patch, but with a little less math involved and a little bit more straight-forward.

At Planio, we run this code for some time now which is a mixture of both approaches:

```
h = hours.floor
m = ((hours - h) * 60).round
if m == 60
  h += 1
  m = 0
end
"%d:%02d" % [h, m]
```

Here, the first two lines (which are the same as in Redmine core) return the exact same result as h, m = (hours * 60).divmod(60).map!(&:round).

#6 - 2022-04-20 14:42 - Denis Yurashku

Holger Just wrote:

> At Planio, we run this code for some time now which is a mixture of both approaches:
>
> ```
> h = hours.floor
> m = ((hours - h) * 60).round
> if m == 60
>   h += 1
>   m = 0
> end
> "%d:%02d" % [h, m]
> ```
>
> Here, the first two lines (which are the same as in Redmine core) return the exact same result as h, m = (hours * 60).divmod(60).map!(&:round).

Yes, I think we can go *this* way. :)

#7 - 2022-04-28 10:14 - Denis Yurashku

- File Format_hours_edit.diff added

So, the adjusted patch.

#8 - 2022-06-14 11:33 - Denis Yurashku

Any movements on this?

Is it "Confirmed"?

#9 - 2022-07-17 07:44 - Go MAEDA

Denis Yurashku wrote:

> Any movements on this?
>
> Is it "Confirmed"?

I still cannot reproduce the issue. Could you tell me detailed steps to output "7:60"? Without this, I cannot write test code.

#10 - 2022-07-18 15:57 - Holger Just

You can reproduce the issue by creating a time entry with decimal logged hours slightly below 8 hours, e.g. 7.991666666666667 hours. Here, the fractional minutes will be rounded up to 60 minutes because 0.991666666666667 * 60 = 59.50000000000002. This value is always allowed as the time entry form also allows entering float values (which are then parsed as floats in lib/redmine/core_ext/string/conversions.rb. External time tracking apps might round durations by the second and then send those exact values which are accepted by Redmine.

Programmatically, you can also create this situation from "full" minutes by logging exactly 75 time entries of 44 minutes each (i.e. 0.7333333333333333 hours). The sum of those time entries then comes at 54.99999999999999. Here, the minutes part is again rounded up to 60 by the existing code.

Depending on the database and the rounding of the floats involved, this situation may also be created with other times as shown in the screenshot as we always round the raw values to two decimal places before summing them in TimeEntryQuery#total_for_hours which may introduce additional error margins.

#11 - 2022-07-19 08:42 - Go MAEDA

- File issue-page.png added

- File spent-time-page.png added

- Tracker changed from Patch to Defect

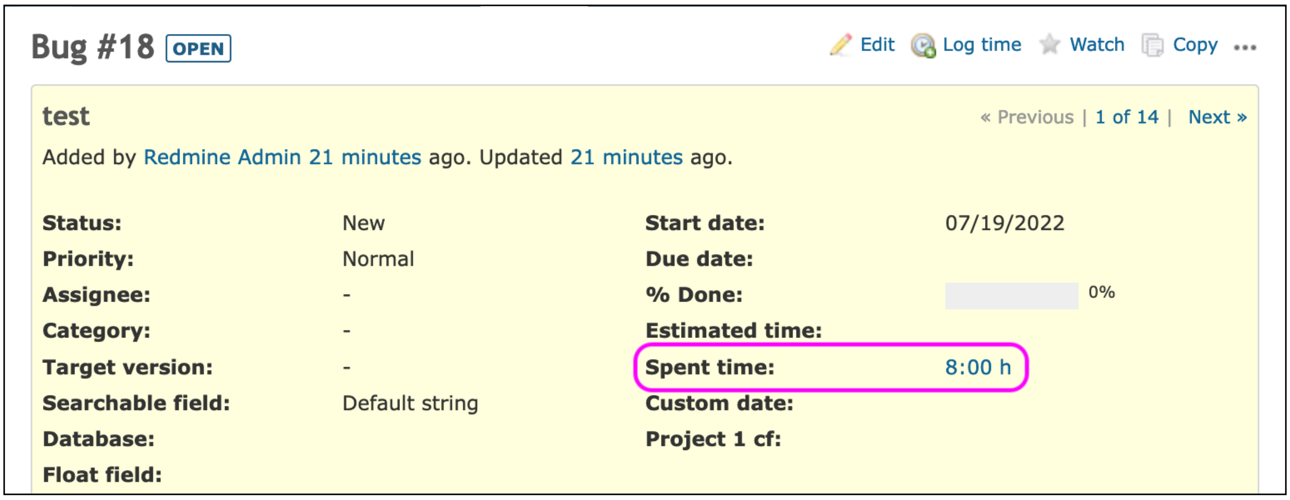
- Status changed from New to Confirmed

Holger Just wrote:

> You can reproduce the issue by creating a time entry with decimal logged hours slightly below 8 hours, e.g. 7.991666666666667 hours. Here, the fractional minutes will be rounded up to 60 minutes because 0.991666666666667 * 60 = 59.50000000000002.

I have confirmed the issue. The spent time value 7.991666666666667 is displayed as "7:60 h" on the issue page and displayed as "8:00 h" after applying the patch.

But I am not sure if the value should be displayed as **"8:00 h"** there because it is displayed as **"7:59"** on the "Spent time" tab. The reason why you see different values on each page is that the Spent time page uses a value 7.99 rounded by TimeEntry#hours (source:tags/5.0.2/app/models/time_entry.rb#L196), while Issue page uses a value 7.991666666666667 obtained from the database (source:tags/5.0.2/app/models/issue.rb#L1140) without using TimeEntry#hours.

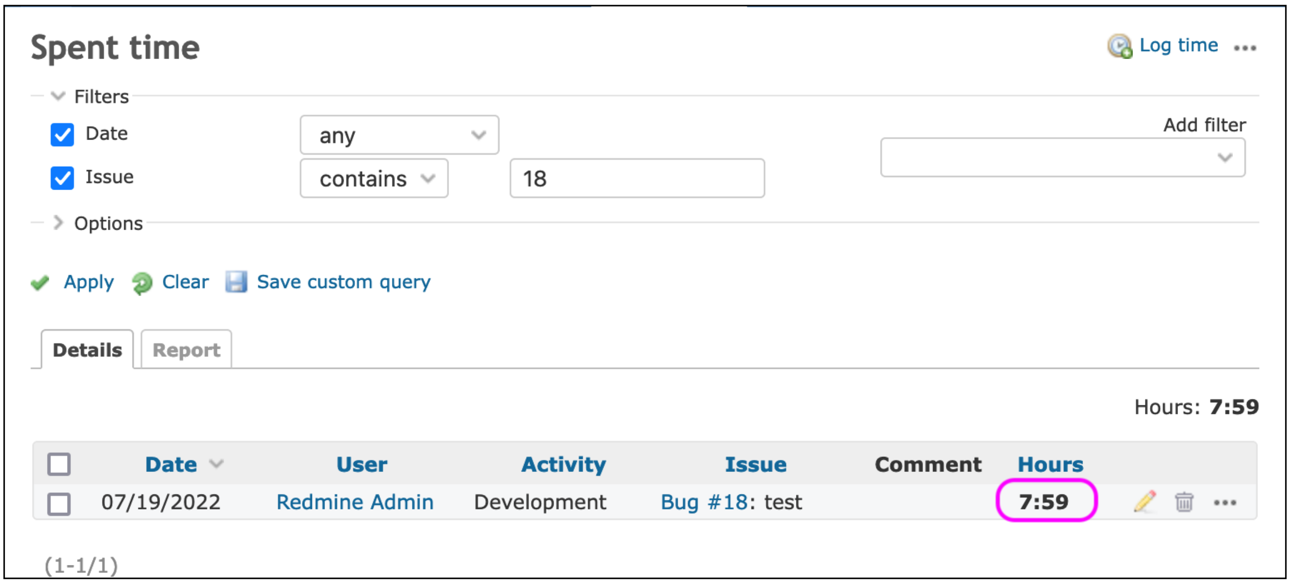


Bug #18 OPEN

test

Added by Redmine Admin 21 minutes ago. Updated 21 minutes ago.

Status:	New	**Start date:**	07/19/2022
Priority:	Normal	**Due date:**	
Assignee:	-	**% Done:**	0%
Category:	-	**Estimated time:**	
Target version:	-	**Spent time:**	8:00 h
Searchable field:	Default string	**Custom date:**	
Database:		**Project 1 cf:**	
Float field:			



Spent time

Filters

☑ Date any

☑ Issue contains 18

Add filter

Options

Hours: **7:59**

□	Date	User	Activity	Issue	Comment	Hours
□	07/19/2022	Redmine Admin	Development	Bug #18: test		7:59

(1-1/1)

#12 - 2024-06-30 10:35 - Go MAEDA

- Related to Defect #40914: Fix precision issues in TimeEntry#hours calculation by returning Rational instead of Float added

#13 - 2024-07-01 03:25 - Go MAEDA

This issue will be fixed if the patch proposed in #40914 is merged.

#14 - 2024-07-22 06:43 - Go MAEDA

- File 36897.patch added

Go MAEDA wrote in #note-13:

> This issue will be fixed if the patch proposed in #40914 is merged.

I would like to propose a new patch to address this issue. This patch can resolve the reported problem without applying the patch in #40914.

This patch converts the given value hours to the nearest minute using (hours * 60).round. Since the converted value is an integer, the minute value obtained from minutes.divmod(60) will always be an integer within the range of 0 to 59. This ensures that edge cases where the minute value would be 60, such as 7:60, are eliminated.

Additionally, whether the timespan_format setting is minutes or decimal, the same rounded minute value is used. This reduces instances where the displayed values appear inconsistent depending on the setting (for example, displaying '8:00' when the setting is minutes but '7.99' when the setting is decimal).

#15 - 2024-07-23 05:53 - Go MAEDA

- Subject changed from Wrong formatting of date in Time Entries to The minutes part of a time entry is displayed as 60 instead of being carried over

#16 - 2024-08-01 07:50 - Go MAEDA

- Target version changed from Candidate for next minor release to 6.0.0

Setting the target version to 6.0.0.

#17 - 2024-08-08 06:25 - Go MAEDA

- Status changed from Confirmed to Closed

- Assignee set to Go MAEDA

- Resolution set to Fixed

Committed the patch in r22946.

#18 - 2025-01-29 02:22 - Go MAEDA

- Related to Defect #42172: `format_hours` method produces incorrect output for negative time values when `Setting.timespan_format` is "minutes" added

Files

Format_hours.diff	450 Bytes	2022-04-05	Denis Yurashku
Снимок экрана от 2022-04-05 16-59-25_0.png	71.6 KB	2022-04-05	Denis Yurashku
Format_hours_edit.diff	443 Bytes	2022-04-28	Denis Yurashku
issue-page.png	98.8 KB	2022-07-19	Go MAEDA
spent-time-page.png	88.4 KB	2022-07-19	Go MAEDA
36897.patch	1.64 KB	2024-07-22	Go MAEDA